Санкт-Петербургский Государственный Университет
Математико-механический факультет

Кафедра информатики

Лыщик Андрей Игоревич

Анализ эмоциональной окраски рецензий к фильмам

Бакалаврская работа

Научный руководитель:
к. ф.-м. н., доцент Бугайченко Д. Ю.

Рецензент:
к. ф.-м. н., доцент Николенко С. И.

Санкт-Петербург
2016

SAINT-PETERSBURG STATE UNIVERSITY

Chair of informatics

Andrey Lyschik

Sentiment analysis for movie reviews

Graduation Thesis

Scientific supervisor:
assistant professor Dmirty Bugaychenko

Reviewer:
assistant professor Sergey Nikolenko

Saint-Petersburg
2016

Оглавление

Введение

В последнее время бурное развитие получила область анализа эмоциональности (sentiment analysis) — семейство методов обработки естественного языка, посвященное идентифицированию и определению эмоциональной окраски текста.

Различные приложения анализа эмоциональности весьма обширны. В связи с повсеместным распространением интернета, социальных сетей, различных аггрегаторов отзывов и рецензий, увеличением вычислительных мощностей, появилась возможность анализа большого количества текстовой информации. Благодаря этому коммерческие компании или исследователи могут эффективно производить анализ отношения к различным продуктам на рынке, автоматическим образом узнавать мнения большого количества людей о происходящих событиях (PR, политические компании).

Эмоциональная окраска текста может определяться различным образом. Популярны градации "положительная"-"отрицательная", возможно добавление нейтральной окраски. Также возможно задавать значение эмоциональности на вещественной шкале. Определять эмоциональность можно как у текста в целом, так и по отношению к определенной теме.

В данной работе рассматривается определение эмоциональной окраски рецензий к кинофильмам. С помощью построения модели, автоматическим образом определяющей тональность данной рецензий можно эффективным образом определять настроение аудитории по отношению к определенной кинокартине, анализируя комментарии, собственно рецензии или посты в социальных сетях.

Существует несколько различных методов и подходов для построения алгоритмов определения эмоциональной окраски. Одним из самых популярных является подход с использованием машинного обучения с учителем. В этой работе рассматриваются методы именно из этой области.

1. Постановка задачи

Целями данной работы являются:

- рассмотреть возможные подходы и алгоритмы к построению моделей классификации рецензий по трем классам эмоциональности: положительный, нейтральный и отрицательный,
- собрать набор данных кинорецензий для обучения и тестирования моделей,
- построить и протестировать различные модели классификации.

2. Обзор литературы

Подходы к решению задачи анализа тональности можно разделить на несколько групп [15]:

- применение машинного обучения;
- словарный;
- статистический;
- семантический.

Также существуют гибридные подходы, применяющие комбинации упомянутых выше.

Рассмотрим теперь основные детали анализа тональности с применением методов машинного обучения, и затем подходы, примененные в некоторых существующих работах.

В данном подходе задача анализа тональности сводится к задаче классификации текстовых документов (рецензий, комментариев, твитов и т.д.). Предполагается, что есть множество документов $D = \{X_1, \ldots, X_n\}$, называемое тренировочным, на основе него с помощью обучения с учителем (если для данных уже известна тональность), или без учителя строится модель, и затем на основе этой модели осуществляются предсказания для новых данных.

Тональность документов в данном случае может описываться как принадлежность документа к одному из классов C. Например, такими классами могут быть $C = \{"Positive","Negative"\}$. Возможна большая гранулярность, где к бинарной классификации добавляется класс нейтральных документов и т.д.

В данном подходе, весь процесс грубо можно разделить на две стадии:

- Извлечение признаков (feature extraction) из документа. Здесь текст рецензии преобразовывается в некий вектор, или набор признаков.

- Применение одного из алгоритмов классификации к полученным наборам признаков для построения модели.

Стадию извлечение признаков в свою очередь можно разделить на несколько этапов [13]:

- Предварительная обработка. На данной стадии выполняются преобразования, не изменяющие синтаксическую структуру документа, такие как частеречная разметка (например, с помощью скрытых марковских моделей или алгоритмов, основанных на правилах), стеммизация или леммификация (процесс приведения слова к "нормальной" форме), удаление стоп-слов (слов, имеющую слишком большую частоту в языке).

- Собственно извлечение признаков.

- Выбор из признаков. На данном этапе могут быть применены такие статистические методы, как χ^2 тест или коэффициент взаимной информации.

В [10] (и почти всех рассмотренных работах) каждый документ d (рецензия на фильм) преобразовывался в вектор $(n_1(d), \ldots, n_F(d))$, где $n_i(d)$ — количество вхождений (или просто наличие) признака f_i в документ d, F — количество всех признаков. В качестве признаков рассматривались:

- Отдельные слова (униграммы); как частота вхождения, так и бинарное наличие слова. Важно отметить, что в данной статье авторы использовали технику отметки слов с отрицанием, указанную в [3], заключающуюся в том, что к униграммам, перед которыми находится слово "not" добавляется пометка "NOT_".

- Только прилагательные.

- Биграммы, и униграммы с биграммами. Для биграмм не проводилось отметки слов с отрицанием.

- Добавление информации о части речи к униграммам.

Стеммизации и удаления стоп-слов авторы данной статьи не проводили, пунктуация рассматривалась как отдельные элемент.

Интересно отметить, что для всех признаков и при всех алгоритмах классификации бинарное вхождение признакак в рецензию показало лучшую точность. В целом, лучшей точности авторам удалось добиться используя униграммы. Добавление информации о частях речи понизило или не изменило точность, с помощью добавления биграмм в набор признаков также не удалось добиться повышения точности.

В [10] были применены следующие алгоритмы классификации:

- Наивный байесовский классификатор.
- Классификатор, использующий метод опорных векторов.
- Классификатор, использующий метод максимума энтропии.

В данной статье рассматривалась задача бинарной классификации документа, имеющего положительную или отрицательную тональность. Авторам удалось добиться лучшей точности в 82.9 % (значение, полученное 3-разовой кросс-валидацией) с помощью классификации методом опорных векторов, с небольшим отставанием двух других алгоритмов.

В [9] были использованы такие методы как

- Лапласовское сглаживание для наивного байесовского классификатора.
- Выборка среди n-грамм (признаков) с помощью коэффициента взаимной информации.
- Отметка униграмм с отрицанием.

В данной работе авторам удалось добиться точности в 88.80 % при бинарной классификации на наборе кинорецензий IMDB.

В [1] авторы использовали несколько типов признаков:

- n-граммы. Были использованы n-граммы встречавшиеся в более чем 10 документах, причем использовались 1-, 2-, и 3-граммы.

- Информация о распределении длин слов: количества слов, встречающихся в документе длины от 1 до 20.

- Информация на уровне конкретных слов: количество слов в документе, средняя длина слова, количество коротких слов (короче 4 символов), и т.д.

- Информация на уровне символов: количество символов в документе, среднее количество символов на предложение, процент символов, формирующих слова и символов, являющихся числами и другими не буквенными символами.

- Информация о величине вокабуляра: количество уникальных слов в документе, количество слов, использованных один раз (hapax legomena), количество слов, использованных дважды.

- Структурные признаки: количество строк, предложений и параграфов, наличие разделений между параграфами, использовалась ли табуляция.

После первоначального выделения признаков среди всех признаков производилась выборка с помощью EWGA (entropy weighted genetic algorithm). В лучшем результате использовалось 1752 признака. Авторам удалось добиться точности на двух классах тональности равной 91.7. В качестве тренировочного и тестового набора данных был использован набор рецензий с IMBD.

В [16] использовали модификацию классификатора методом опорных векторов NBSVM (SVM with Naive Bayes features). Используя вектора вхождений униграмм и биграмм авторам удалось достичь точности в 91.22 % на 10-разовой кросс-валидации на большом наборе данных (50000 примеров) рецензий с IMDB, первоначально представленном в [7]. Авторы не удаляли стоп-слова и расценивали пунктуацию как отдельные символы.

Другой подход к векторизации текста был представлен в [6], в данной статье описывается `Paragraph2Vec` или `Doc2vec`, способ нахождения вектора для параграфа или предложения, основанный на векторах слов, первоначально представленных в [4]. Авторам удалось добиться правильности классификации в 92.58 % на упомянутом ранее наборе рецензий с IMDB.

В [11, 5, 18] рассматриваются применения методов глубокого обучения к классификации текстов по тональности.

В [11] описывается применение рекурсивных нейронных тензорных сетей с применением парсинга текста к классификации предложений и более коротких фраз как по бинарной шкале, так и более точной (5-классовой, от очень негативного до очень позитивного). Используя данный метод авторам удалось добиться точности 85.4 % при бинарной шкале, и 45.7 % при пятиклассовой, на наборе фраз, взятых из рецензий с `rottentomatoes.com`.

В [18] рассматривается применение сверточных нейронных сетей к классификации тональности на том же наборе данных, что и в [11]. Авторы использовали извлечение признаков уровня символов, слов и предложений (character-, word-, sentence-level embeddings), и используя данный метод удалось достичь точности 48.3 на пятиклассовой шкале тональности, и 85.7 % на бинарной.

В [5] рассматривалось применение последовательности шумоподавляющих ауто-энкодеров (stacked denoising auto-encoders) для предсказывания рейтинга рецензии ресторана по 5-балльной шкале, используя в качестве признаков модель bag-of-words. Авторам удалось добиться показателя RMSE в 0.746.

3. Рассмотренные методы

3.1. Наивный байесовский классификатор

Семейство байесовских классификаторов основывается на теореме Байеса, одна из формулировок которой следующая: при наличии переменной класса y, и набора признаков $x_1, \ldots, x_n$, верно равенство:

$$P(y|x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots, x_n|y)}{P(x_1, \ldots, x_n)}$$

Делается предположение, что

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y), \forall i$$

При этом предыдущее равенство сводится к

$$P(y|x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \ldots, x_n)}$$

а так как $P(x_1, \ldots, x_n)$ константа при данном наборе признаков, то можно использовать следующее правило для классификации: ([8])

$$\hat{y} = \arg\max_y P(y)\prod_{i=1}^{n} P(x_i|y)$$

Различные виды байесовских классификаторов различаются в предположениях о распределении $P(x_i|y)$.

В данной работе рассматривалось применение варианта классификатора, делающее предположение о мультиномиальном распределении. Вероятность нахождения признака x_i в документе y в данном варианте представляется как

$$P(x_i|y) = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

где N_{yi} — количество вхождений i-го признака в документы, принадлежащие классу y на тренировочном множестве, n — количество признаков, N_y — количество всех признаков, входящих в документы из тренировочного множества, принадлежащих классу y. α — сглаживаю-

щий параметр классификатора.

В работе использовалась реализация, входящая в пакет sklearn ([14]).

3.2. Классификация методом опорных векторов

Классификатор методом опорных векторов с линейным ядром производит классификацию строя разделяющую гиперплоскость в пространстве признаков, допуская неправильную классификацию некоторых наблюдений.

Сформулировать метод классификации можно следующим образом: наблюдения $x_i \in \mathbf{R}^p, i \in 1, \ldots, n$, (где p — количество признаков, n — количество наблюдений в тренировочном множестве) принадлежат одному из двух классов $y_i \in \{-1, 1\}$. Для классификации необходимо построить функцию вида

$$\hat{y} = w^T x + b$$

где $w \in \mathbf{R}^p, b \in \mathbf{R}$. Соответственно для x_i будет предсказан класс в зависимости от знака: $\operatorname{sgn} w^T x_i + b$.

Поиск параметров модели w и b можно производить с помощью минимизации регуляризованной функции ошибки на тренировочном множестве:

$$E(w, b) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \alpha R(w)$$

.

Для метода опорных векторов функция ошибки выглядит как: ([17])

$$L(y_i, f(x_i)) = \max(0, 1 - y_i f(x_i)) = \max(0, 1 - y_i(w^T x_i + b))$$

В качестве регуляризации параметров можно выбрать:

- L_1 норму параметров: $R(w) = \sum_{i=1}^{p} |w_i|$
- L_2 норму параметров: $R(w) = \frac{1}{2} \sum_{i=1}^{p} w_i^2$

Нахождение параметров эффективно производится с помощью стохастического градиентного спуска [17]. Для каждого примера из тре-

нировочного множества рассматривается градиент функции ошибки, и происходит обновление параметров:

$$w \leftarrow w - \eta \left(\alpha \frac{\partial R(w)}{\partial w} + \frac{\partial L(w^T x_i + b, y_i)}{\partial w} \right),$$

где η — параметр, отвечающий за скорость обучения. Значение b обновляется похожим образом, но без регуляризации.

В работе использовалась реализация, входящая в пакет sklearn ([14]).

3.3. NB SVM

В упомянутой в лит. обзорной части работе [16], авторы предложили модификацию метода опорных векторов, показывающую одну из самых высоких точностей на широко рассматриваемом наборе данных кинорецензий от IMDB.

Рассматривается классификация на двух классах $y \in -1, 1$. Определим

$$p_i = \alpha + N_{1i}$$
$$q_i = \alpha + N_{-1i}$$

где N_{yi} — количество документов класса y, в которые входит признак i. Тогда определим

$$\mathbf{p} = (p_1, \ldots, p_n)$$
$$\mathbf{q} = (q_1, \ldots, q_n)$$
$$\mathbf{r} = \log \left(\frac{\mathbf{p}/||\mathbf{p}||_1}{\mathbf{q}/||\mathbf{q}||_1} \right)$$

В таком случае, авторами было предложено использовать

$$\mathbf{r} \circ \mathbf{x}$$

(где $\mathbf{x}_i$ — индикатор, входил ли i-й признак в документ) как вектор

признаков для классификации методом опорных векторов.

3.4. Деревья с градиентным бустингом

В качестве реализации обучения моделей данного типа использовалась реализация XGBoost, приведенная в [2]. Дальнейшие выкладки и описания приведены в соответствии с этой статьей.

В данном методе классификация (или регрессия) происходит с помощью построения ансамбля деревьев. Предсказание для примера x_i формулируется как

$$\hat{y_i} = \sum_{k=0}^{K} f_k(x_i),$$

где K — количество деревьев в ансамбле, $f_i \in F$, F — множество всех деревьев. Поиск оптимального набора происходит оптимизацией следующей функции:

$$Obj(\theta) = \sum_{i}^{n} L(y_i, \hat{y_i}) + \sum_{k=0}^{K} \Omega(f_k), \tag{1}$$

где $\Omega(f_k)$ — терм регуляризации для дерева f_k.

Поиск деревьев происходит пошагово, на каждом шагу добавляя по дереву и обновляя предсказание по входным признакам x_i:

$$\hat{y}_i^{(0)} = 0$$

$$\dots$$

$$\hat{y}_i^{(t)} = \sum_{k=0}^{t} f_k(x_i) = y_i^{(t-1)} + f_t(x_i)$$

Дерево можно описать как

$$f_t(x) = w_{q(x)}, w \in \mathbf{R}^T, q : \mathbf{R}^p \to \{1, 2, \dots, T\},$$

где w — вектор значений в листьях, q — функция, отображающая

набор признаков в определенный лист дерева, T — количество листьев. Терм регуляризации определить как

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2,$$

где λ — параметр модели.

Для оптимального выбора дерева функция 1 для шага t расписывается по формуле Тейлора для членов до второго порядка, которая после удаления констант выглядит как:

$$Obj^{(t)} = \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)w_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2] + \gamma T,$$

где

$$g_i = \partial_{\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)})$$
$$h_i = \partial^2_{\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)}),$$

и $I_j = \{i|q(x_i) = j\}$ — набор индексов обучающих примеров, попавших в j-й лист дерева. Тогда для фиксированной структуры дерева $q(x)$ оптимальные веса для листьев на шагу t можно найти как

$$w_j^* = -\frac{G_j}{H_j + \lambda}$$
$$Obj^* = \frac{1}{2}\sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T,$$

где $G_j = \sum_{i\in I_j} g_i$ и $H_j = \sum_{i\in I_j} h_i$.

Оптимальная структура дерева находится начиная с одного листа, и жадным образом добавляя разделение листа на ветви с новыми листами. При попытке разделить лист дерева на две ветви рассматривается ценность такого разделения:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma$$

Также при обновлении модели на каждом шагу можно рассматривать параметр η, снижающий вклад в модель от дерева, и, возможно, уменьшающий переобучение:

$$y^{(t)} = y^{(t-1)} + \eta f_t(x)$$

3.5. Классификация, основанная на двух классах

Рассмотрим бинарный классификатор (например, логистическую регрессию), по данным признакам x_i предсказывающий вероятность $P(y_i = c|x_i)$ принадлежность к тому или иному классу. Для рецензий, рассматриваемых в данной работе, принадлежащих к одному из трех классов, можно построить классификатор следующего вида:

- выбирается два класса c_1 и c_2 (например положительные и отрицательные рецензии),
- на этапе обучения не рассматриваются рецензии из невыбранного класса c_3, и обучается модель на двух классах
- в момент предсказания на новых данных вычисляются вероятности принадлежности к зафиксированным классам, и на основе двух параметров P_{c_1} и P_{c_2} делается классификация:
  - если вероятность принадлежности x к c_i больше P_{c_i}, то классифицируем как c_i.
  - в ином случае производится выбор в пользу c_3.

Случай фиксирования классов положительных и отрицательных рецензий, и вычисления вероятностей принадлежности к этим классам можно обосновать интуитивным предположением о том, что нейтральные рецензии лежат где-то "посередине", не обладая признаками отрицательности или положительности.

4. Рассмотренные признаки

4.1. Bag-of-words

Для применения классификации с помощью всех методов, за исключением использующих `doc2vec`, необходимо из текста рецензии получать набор признаков, имеющих числовое значение. В качестве такого набора признаков чаще всего используют модель bag-of-words, выделяющую из текстов рецензий некоторый набор термов, и затем сопоставляющую каждой рецензии вектор $\mathbf{x}$ размера N, где N может быть количеством уникальных термов во всем наборе, и $\mathbf{x}_i$ может быть:

- 1 или 0 — отмечает вхождение токена в текст,
- k, количество вхождений токена в текст рецензии.

Перед выделением термов из текста над словами можно производите следующие манипуляции:

- разбиение на слова, с преобразованием слов к нормальной форме (лемматизация);
- преобразование, заключающееся в том, что токен t, следующий за «не» или «нет», преобразовавывался в токен вида «не_t».

После первичных преобразований и разбиения на отдельные токены, из текста можно выделять термы, такими термами чаще всего являются n-граммы. 1-граммами являются единичные токены, 2-граммами — 2 токена, идущих подряд, и т.д.

Также можно не рассматривать термы, входящие в менее чем min_df рецензий, или термы, частота вхождения которых больше чем max_df (стоп-слова).

4.2. Doc2Vec

Другой подход к преобразованию текстовой информации (предложения, параграфа, документа) в вектор был представлен в [6].

В данной статье предлагается метод векторизации параграфов, основывающийся на получении векторного представления для слов (word embeddings) ([4]). В данной модели каждому слову сопоставляется вектор w_i, i — позиция слова в наборе, все вектора слов из набора образуют матрицу W. Сами вектора ищутся следующим образом: при данной последовательности (контекстного окна) слов $w_1, w_2, \ldots, w_T$, модель максимизирует средний логарифм вероятностей

$$\frac{1}{T}\sum_{t=k}^{T-k}\log p(w|w_{t-k},\ldots,w_{t+k})$$

Данные вероятности можно предсказывать используя softmax функцию:

$$p(w|w_{t-k},\ldots,w_{t+k}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}},$$

где y_i — вероятности для каждого слова из набора, вычисляемые как

$$y = b + Uh(w_{t-k},\ldots,w_{t+k};W),$$

где b, U — параметры модели, h — конкатенация или усреднение векторов, полученных для $w_{t-k},\ldots,w_{t+k}$. Значения параметров и векторов модели ищутся с помощью стохастического градиентного спуска и алгоритма обратного распространения ошибки.

Векторизация параграфов отличается тем, появляются, собственно, вектора для параграфов, являющиеся колонками в матрице D, и вероятность появления слова по контексту вычисляется, учитывая вектор для параграфа, из которого берется контекст:

$$y = b + Uh(w_{t-k},\ldots,w_{t+k};W;D),$$

Матрица векторов слов W является общей для всех параграфов.

После обучения вектора для новых, еще не встречавшихся параграфов получаются фиксированием параметров модели U, b, W, добавлением колонок в матрицу D (хранящих вектора для новых параграфов), и

оптимизацией значений из D методом градиентного спуска.

Далее вектора, полученные для документов можно классифицировать различными алгоритмами классификации.

В данной работе использовалась реализация описанной модели из [12].

5. Применение классификаторов

5.1. Набор данных

Для обучения и тестирования классификаторов был получен набор данных кинорецензий (доступен по ссылке из [19]) с сайта `kinopoisk.ru` количеством 63532 рецензии с проставленной авторами тональностью, из них

- 42643 (67.1%) с положительной,
- 10750 (16.9%) с отрицательной,
- 10139 (16%) нейтральных.

Набор рецензий имеет следующие характеристики:

- среднее количество слов в рецензии — 1971,
- среднее количество предложений — 27.6,
- количество уникальных слов (без приведения к нормальной форме) — 423201,
- количество уникальных слов (с приведением к нормальной форме) — 170530,
- среднее количество уникальных слов без приведения к нормальной форме на рецензию — 254.4,
- среднее количество уникальных слов, приведенных к нормальной форме на рецензию — 220.4.

5.2. Сравнение классификаторов

Набор данных был разделен на тренировочное (60% от всего набора) и тестовое (40%) множества. На тренировочном множестве с помощью 10-разовой кросс-валидации подбирались лучшие варианты гиперпараметров классификаторов и, собственно, сравнивались классифика-

торы между собой. На тестовом множестве проверялось качество классификации на новых данных.

В качестве меры качества классификации было выбрано взвешенное среднее f1-score по классам рецензий. F1-score для класса классификации, определяется как

$$F_1 = 2 \cdot \frac{\text{точность} \cdot \text{полнота поиска}}{\text{точность} + \text{полнота поиска}}$$

где значения точности (precision) и полноты поиска (recall) вычисляются соответственно для данного класса. Взвешенное среднее берется для учета того, что количество представителей разных классов может быть неравномерным.

Кроме подбора гипер-параметров классификаторов также необходимо было выбрать лучший для данного классификатора способ получения модели bag-of-words (для тех классификаторов, которые ее используют). В них возможны следующие различия:

- как учитывать вхождение терма в рецензию, бинарное вхождение (1 или 0), или количество вхождений,
- min_df — минимальное количество рецензий, в которые входит терм, термы с количеством рецензий, в которые они входят, ниже, не рассматриваются,
- max_df — максимально возможная частота вхождения терма в рецензии, термы с частотой вхождения выше не рассматриваются,
- рассматриваемые n-граммы. Рассматривались 1-, 2- и 3-граммы,
- преобразования над словами: без преобразований, лемматизация и приписывания «не» или «нет» к следующему слову.

5.2.1. Байесовский классификатор

У байесовского классификатора возможен подбор (кроме параметров векторизации рецензий) параметра $\alpha \in (0, 1)$, отвечающего за сглаживание (Лидстоуна или Лапласа).

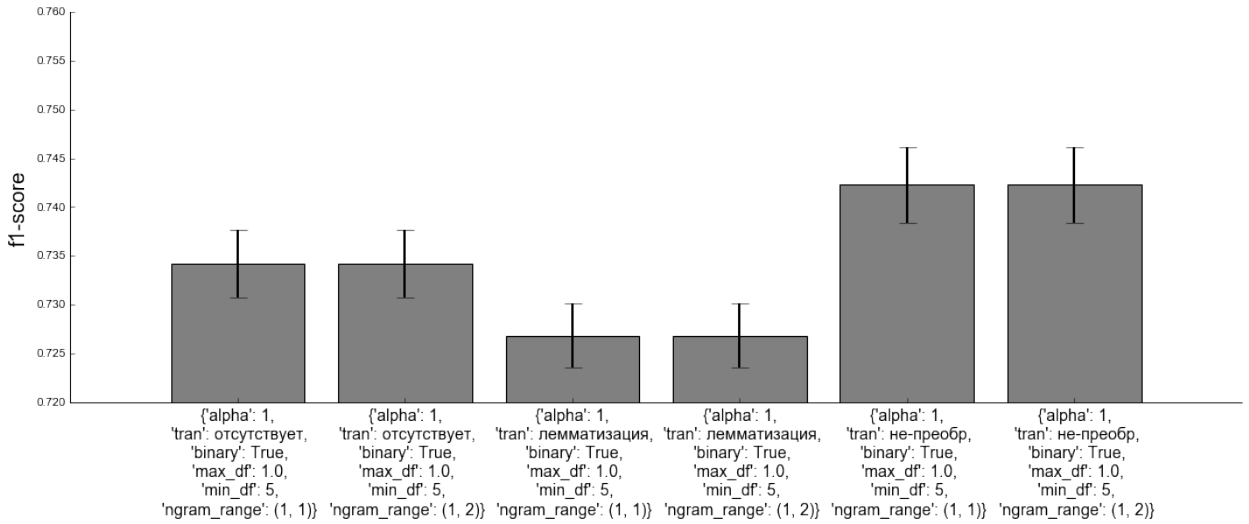

Рис. 1: Результаты по байесовскому классификатору на тренировочных данных

Для всех комбинаций параметров классификатора и векторизатора производился подсчет среднего значения f1-score на 10-разовой классификации, и выявлялись лучшие значения.

На рисунке 1 представлены результаты кросс-валидации среди лучших комбинаций параметров (даны средние значения по кросс-валидации и 95 процентный доверительный интервал f1-score). Все показанные классификаторы используют значение α равное 1, и бинарный (1 или 0) показатель вхождения терма в документ. Классификаторы, применяющие лемматизацию показали худший результат. Лучший результат показал классификатор, использующие «не»-преобразование над словами и использующее только 1-граммы, и не берущий в рассмотрение термы, встречающиеся в менее чем 5 рецензиях. Среднее значение f1-score 0.742 и доверительный интервал $(0.738, 0.746)$, что делает превосходство в результате по f1-score над остальными вариантами байесовского классификатора статистически значимым. Среди двух классификаторов, использовавших «не»-преобразование, использование биграмм не дает никакого преимущества.

Рассмотрим далее поведение данного классификатора на тестовых данных.

В таблице 1 представлена матрица смешения классификатора на тестовых данных. Различные показатели классификатора на тестовых

	Отр. (классиф.)	Нейтр. (классиф.)	Полож. (классиф.)
Отр. (реал.)	2921	752	622
Нейтр. (реал.)	1090	1328	1612
Полож. (реал.)	795	1645	14648

Таблица 1: Матрица смешения байесовского классификатора

	Точность	Полнота поиска	f1-score	Представителей
Отрицательные	0.61	0.68	0.64	4295
Нейтральные	0.36	0.33	0.34	4030
Положительные	0.87	0.86	0.86	17088
Среднее / Всего	0.74	0.74	0.74	25413

Таблица 2: Байесовский классификатор на тестовых данных

данных можно увидеть в таблице 2. По ним можно видеть, что классификация на положительных данных происходит хорошим образом, ситуация с отрицательными несколько хуже, и на классе нейтральных рецензий достигаются худшие показатели.

Для проверки стабильности классификатора были проделаны следующие действия. В первом прогоне кросс-валидации были выбраны по 20 признаков, для каждого класса, с максимальным значением вероятности $P(x_i|y)$. Затем в каждом последующем прогоне значения данной вероятности для каждого класса сортировались по убыванию, и рассматривалась позиция выбранных признаков в таком списке. В таблице 3 представлены средние значение такой позиции. Видно, что для всех классов наибольшие вероятности имеют униграммы, не дающие конкретной информации о принадлежности рецензии к определенному классу, а просто слова, наиболее часто встречающиеся в текстах рецензий.

5.2.2. SVM

Для классификатора методом SVM можно выбрать следующие гиперпараметры:

- α — выражающийся через параметр C как $C = \frac{\text{количество примеров}}{\alpha}$,

Признак	Средняя позиция
,	1.0
.	2.0
и	3.0
в	4.0
что	5.0
на	6.0
но	7.35
из	8.32
с	8.67
это	9.67
а	12.35
как	12.0
фильм	11.60
я	14.0
так	16.39
все	15.32
же	17.57
то	19.25
по	19.10
к	18.67

(a) Отрицательные

Признак	Средняя позиция
,	1.0
.	2.0
и	3.0
в	4.0
что	5.0
на	6.0
но	7.35
с	8.67
это	9.67
из	8.32
фильм	11.60
как	12.0
а	12.35
я	14.0
все	15.32
так	16.39
к	18.67
же	17.57
по	19.10
то	19.25

(b) Нейтральные

Признак	Средняя позиция
,	1.0
.	2.0
и	3.0
в	4.0
что	5.0
на	6.0
из	8.32
но	7.35
с	8.67
это	9.67
фильм	11.60
как	12.0
а	12.35
я	14.0
все	15.32
его	16.0
он	17.0
так	16.39
к	18.67
то	19.25

(c) Положительные

Таблица 3: Топ-20 признаков по классам (Байесовский классификатор)

- L_1 или L_2 норма члена регуляризации, L_1 дает более "разреженные" модели
- n_iter — количество итераций (используемый метод нахождения функции классификации SVM находит ее методом градиентного спуска, соответственно n_iter — количество итераций поиска).

Среди возможных значений этих параметров и параметров векторизатора производился поиск лучших значений по f1-score с помощью кросс-валидации. На рисунке 2 представлены результаты (f1-score) лучших комбинаций параметров 10-разовой кросс-валидации классификаторов типа SVM на тренировочном множестве. Интересно заметить, что

- все классификаторы из лучших используют L_1 норму,
- все классификаторы используют количество вхождений, а не бинарные показатели вхождений,
- все классификаторы используют отсечение термов, входящих в слишком малое количество рецензий ($min\_df > 0$).

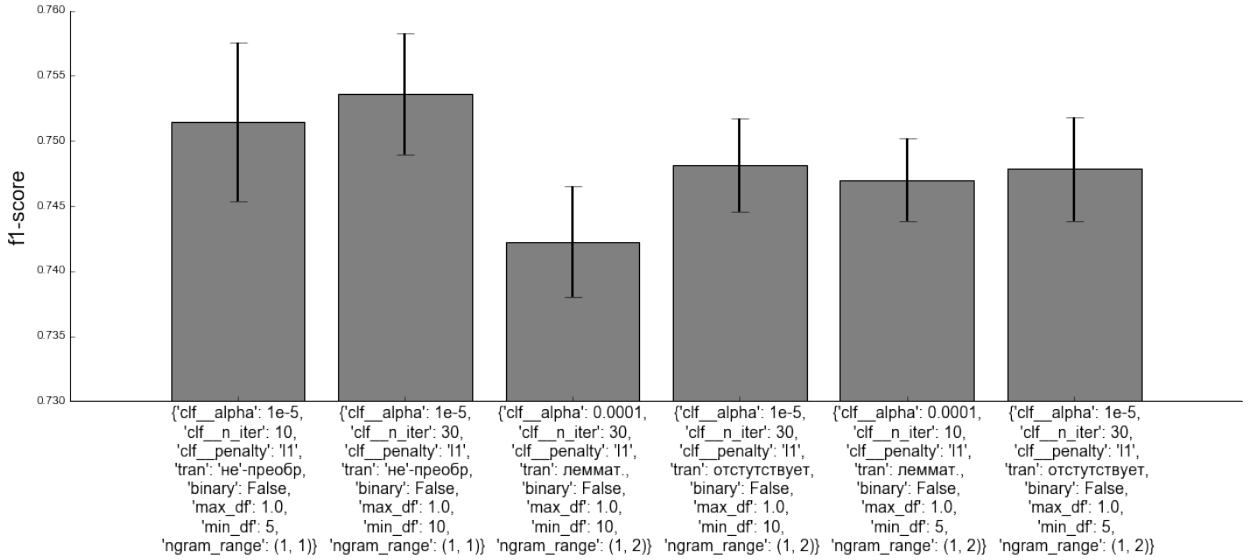

Рис. 2: Результаты по SVM классификатору на тренировочных данных

	Точность	Полнота поиска	f1-score	Представителей
Отрицательные	0.65	0.67	0.66	4295
Нейтральные	0.37	0.35	0.36	4030
Положительные	0.88	0.88	0.88	17088
Среднее / Всего	0.76	0.76	0.76	25413

Таблица 4: Результаты SVM на тестовых данных

- 2 классификатора с лучшим показателем среднего f1-score по кросс-валидации использовали «не»-преобразование и только униграммы.

Лучшее значение среднего f1-score по кросс-валидации равно 0.754 (2 значение на диаграмме), с доверительным интервалом (0.749, 0.758). Нельзя сказать, что преимущество данного классификатора над остальными (за исключением классификатора 3) статистически значимо, так как интервалы остальных класификаторов пересекаются с интервалом этого классификатора.

В таблице 4 представлены различные метрики классификатора 2 на тестовых данных, а в таблице 5. Можно сказать, результаты аналогичны результатам байесовского классификатора, низкие результаты на нейтральных рецензиях и хорошие показатели на положительных.

В таблице 6 представлены данные по признакам, построенные по

	Отр. (классиф.)	Нейтр. (классиф.)	Полож. (классиф.)
Отр. (реал.)	2891	879	525
Нейтр. (реал.)	1015	1407	1608
Полож. (реал.)	572	1535	14981

Таблица 5: Матрица смешения SVM на тестовых данных

Признак	Средняя позиция
скучно	3.9
разочарование	1.5
отсутствует	6.9
скучный	4.7
бред	4.7
никакой	10.8
понимаю	43.2
не_верю	20.8
балл	90.2
логики	33.6
мягко	41.6
оказалась	89.3
итак	18.3
разочаровал	42.5
простите	26.8
увы	33.0
единственный	10.1
диалоги	35.8
ничем	20.2
ужасно	12.4

(a) Отрицательные

Признак	Средняя позиция
неплохой	7.8
нейтральная	5.5
не_плохой	10.8
не_более	6.8
претензий	70.8
показалась	210.0
сюда	126.9
рассказать	15.7
вот-вот	462.3
старались	95.7
половину	164.1
противоречивые	86.6
нейтральной	10.6
плохим	127.2
ровно	116.4
более-менее	116.2
живых	105.8
жили	801.1
чрезмерно	313.8
не_впечатлила	708.5

(b) Нейтральные

Признак	Средняя позиция
браво	1.3
сумел	34.6
марк	5.5
отличный	7.9
гармонично	32.6
район	75.9
великолепно	53.2
идеально	18.1
кристофер	84.5
смог	37.6
приятно	10.6
впервые	28.5
потрясающая	46.9
рекомендую	39.7
гениально	14.9
потрясающе	12.3
глаз	318.6
удалось	68.5
смотрите	277.6
камеры	115.6

(c) Положительные

Таблица 6: Топ-20 признаков по классам (SVM)

методике, схожей с той, что была описана в разделе 5.2.1. Так как SVM в случае трех классов работает по схеме "один против всех", то для каждого класса строился классификатор, определяющий принадлежит ли тренировочный пример данному классу, или нет. В первом прогоне для каждого класса выбирались признаки с наибольшим присвоенным весом в соответствующем классификаторе построенной модели, и затем вычислялась средняя позиция в таких же отсортированных списках по убыванию в остальных прогонах кросс-валидации. Видно, что модели удалось определить как довольно стабильные признаки, имеющие высокий вес в каждом прогоне, так и признаки, повышающие качество классификации только на первом прогоне. Нужно отметить высокий присвоенный вес термам, полученным в ходе "не"-преобразования. Также можно заметить зависимость модели от данного набора данных — как например, довольно высокую среднюю позицию униграммы "кристофер" для положительного класса, появившуюся из-за обилия положительных рецензий на фильмы Кристофера Нолана.

5.2.3. NB SVM

Для данного классификатора гиперпараметры аналогичны обычному SVM. Лучшие результаты по кросс-валидации показаны на рисунке 3.

Основные выводы:

- как и в статье [16], лучшего результата удалось добиться используя L_2 норму,
- различия с результатами статьи в том, что лучший f1-score на данном наборе данных достигается с использованием количества вхождений, а не бинарного присутствия или отсутствия терма
- лучшие результаты по среднему значению f1-score, как и на других классификаторах, достигаются с использованием «не»-преобразования,
- в отличие от предыдущих классификаторов лучшие результаты у моделей, использующих биграммы и триграммы.

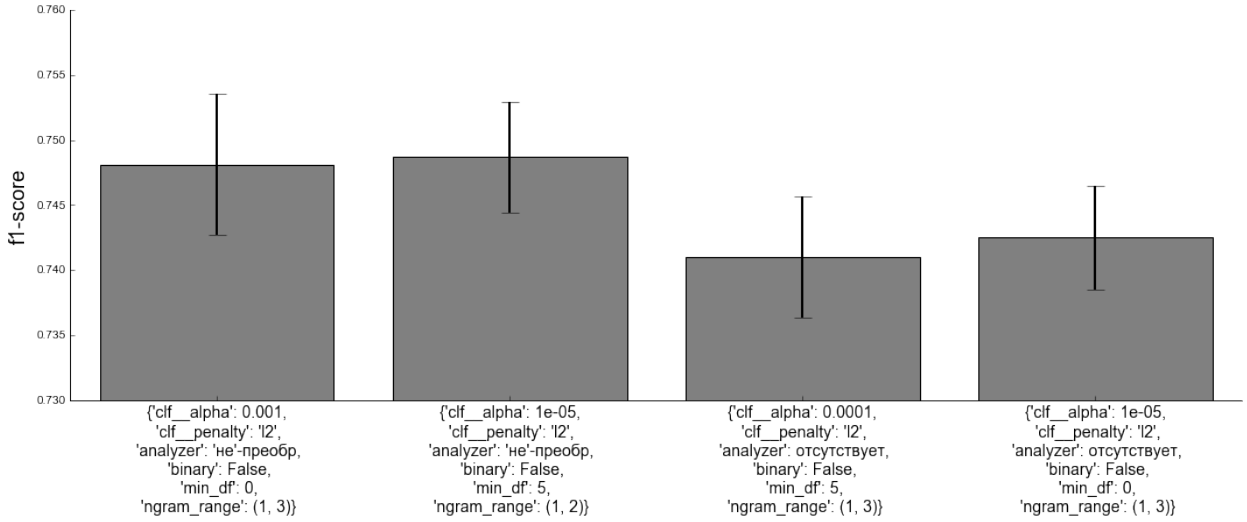

Рис. 3: Результаты по NB SVM классификатору на тренировочных данных

	Точность	Полнота поиска	f1-score	Представителей
Отрицательные	0.67	0.60	0.63	4295
Нейтральные	0.36	0.29	0.32	4030
Положительные	0.84	0.91	0.87	17088
Среднее / Всего	0.74	0.76	0.75	25413

Таблица 7: Результаты NB SVM на тестовых данных

Лучший результат среднего f1-score получился у классификатора 2 на диаграмме 3: 0.749 с доверительным интервалом (0.744, 0.753).

В таблицах 7 и 8 представлены метрики и матрица смешения на тестовых данных. Видно, что за счет уменьшения показателей на отрицательных и нейтральных рецензиях поднялась полнота поиска на положительных рецензиях.

В таблице 9 представлена информация о стабильности весов, присвоенных признакам. Видно, что в данном случае в каждом прогоне кросс-валидации большие веса получали примерно одни и те же признаки, что

	Отр. (классиф.)	Нейтр. (классиф.)	Полож. (классиф.)
Отр. (реал.)	2562	862	871
Нейтр. (реал.)	851	1164	2015
Полож. (реал.)	387	1183	15518

Таблица 8: Матрица смешения NB SVM на тестовых данных

Признак	Средняя позиция
увы	1.1
скучно	2.1
разочарование	2.8
разочаровал	4.2
сожалению	6.6
зачем	7.4
большего	10.1
видимо	7.8
не	8.9
неплохой	9.8
не	8.8
не	20.8
ничего	11.0
плохо	14.1
бред	23.4
плюсы	27.9
ощущение	24.0
снимать	18.4
ужасно	20.5
плохой	15.4

(a) Отрицательные

Признак	Средняя позиция
из	1.0
шедевр	7.1
фильм	9.78
рекомендую	14.26
фильмом	5.1
часа	31.7
советую	6.15
заставляет	19.0
первых	16.4
остаться	17.2
отличный	8.78
смотрите	18.1
долго	29.4
ставлю	5.6
не	33.2
дуэт	57.9
скучный	29.5
актёры	21.8
актеры	15.4
хочется	38.7

(b) Нейтральные

Признак	Средняя позиция
понравился	1.0
удалось	7.2
высоте	5.7
приятно	3.2
получился	2.9
несмотря	6.1
немного	8.0
советую	6.15
отличный	8.78
отлично	15.0
потрясающий	12.9
понравилось	5.7
стоит	11.2
слегка	18.1
великолепно	32.3
рекомендую	14.26
целом	16.5
фильм	9.78
заслуживает	22.1
браво	37.8

(c) Положительные

Таблица 9: Топ-20 признаков по классам (NB SVM)

говорит о способности модели справляться с переобучением. В данном случае значение средних позиций по прогонам кросс-валидации ниже, чем у модели на основе SVM. Также можно видеть, что в признаках с наибольшим весом для нейтрального класса присутствуют слова, имеющие как положительную, так и отрицательную окраску. Это объясняет низкое качество классификации (в частности recall) данной модели на нейтральных рецензиях.

5.2.4. Классификация на основе вероятностей принадлежности к двум фиксированным классам

В случае данного классификатора необходимо подбирать P_{c_1} и P_{c_2}, пороговые значения для вероятности принадлежности к одному из фиксированных классов.

Результаты можно видеть на диаграмме 4.

Можно видеть, что:

- среди лучших моделей все с фиксированными положительным и отрицательным классами,
- лучшие результаты показали варианты классификаторов, использующие P_{pos} и P_{neg} равные 0.8 и 0.6 соответственно,

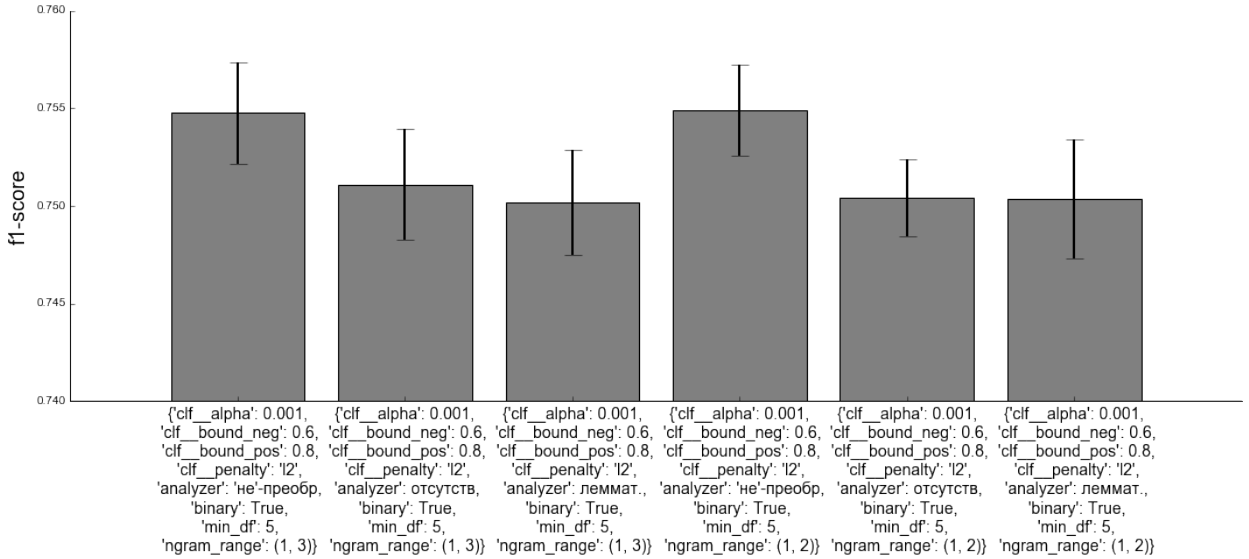

Рис. 4: Результаты полож./отриц. классификаторов на тренировочных данных

	Точность	Полнота поиска	f1-score	Представителей
Отрицательные	0.69	0.63	0.66	4295
Нейтральные	0.31	0.21	0.25	4030
Положительные	0.85	0.93	0.89	17088
Среднее / Всего	0.73	0.77	0.75	25413

Таблица 10: Результаты полож./отриц. классификаторов на тестовых данных

- классификаторы, использующие «не»-преобразование снова лидируют,
- все лучшие модели используют L_2 норму.

Лучший результат по среднему f1-score получила модель 4 на диаграмме 4, среднее 0.755, доверительный интервал (0.752, 0.757).

Как и ожидалось для классификатора этого типа, на тестовых данных можно видеть (таблицы 10, 11), что модель довольно неплохо предсказывает положительные и отрицательные классы.

5.2.5. Doc2Vec + SVM

Для преобразования рецензий в вектора модели doc2vec необходимо было подобрать следующие параметры:

	Отр. (классиф.)	Нейтр. (классиф.)	Полож. (классиф.)
Отр. (реал.)	2708	926	661
Нейтр. (реал.)	969	830	2231
Полож. (реал.)	261	900	15927

Таблица 11: Матрица смешения полож./отриц. классификатора на тестовых данных

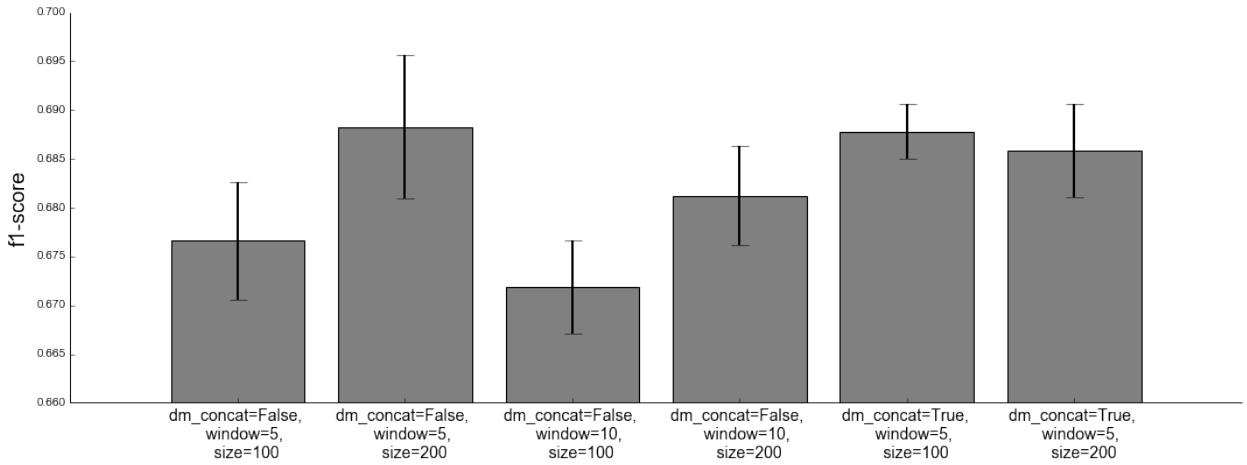

Рис. 5: Результаты doc2vec на тренировочных данных

- использовать конкатенацию векторов из контекстного окна, или усреднение
- размерность векторов (слов) рецензий
- размер контекстного окна

Также в подборе нуждаются гиперпараметры SVM.

На диаграмме 5 отмечены результаты f1-score по кросс-валидации. Лучший результат (0.688, доверительный интервал (0.681, 0.696)) показала модель 2.

	Отр. (классиф.)	Нейтр. (классиф.)	Полож. (классиф.)
Отр. (реал.)	3320	277	698
Нейтр. (реал.)	1857	481	1692
Полож. (реал.)	2153	924	14011

Таблица 12: Матрица смешения классификатора использующего doc2vec вектора на тестовых данных

	Точность	Полнота поиска	f1-score	Представителей
Отрицательные	0.45	0.77	0.57	4295
Нейтральные	0.29	0.12	0.17	4030
Положительные	0.85	0.82	0.84	17088
Среднее / Всего	0.70	0.70	0.69	25413

Таблица 13: Результаты классификатора использующего doc2vec вектора на тестовых данных

В таблицах 12, 13 представлено поведение классификатора на тестовых данных. Видно, что из результатов можно разве что отметить довольно высокую полноту поиска по отрицательным рецензиям, в остальном показатели низкие.

5.2.6. Ансамбли классификаторов

На основе получившихся моделей было решено построить ансамбли. Рассматривались ансамбли двух типов:

- модель голосования, где на основе предсказаний всех ранее рассмотренных моделей с лучшими комбинациями параметров выдается класс, предсказанный больше всего,
- модель, использующая в качестве признаков предсказания и другую информацию, выдаваемую классификаторами (вероятности, функции решения), и обучающаяся с помощью XGBoost.

Результат модели голосования по среднему f1-score на кросс-валидации оказался равен 0.761, с доверительным интервалом в (0.756, 0.767). Результаты данной модели на тестовых данных представлены в таблицах 14 и 15.

Второй тип ансамбля, на основе XGBoost использовал следующие признаки:

- предсказания всех классификаторов, закодированные с помощью one-hot-encoding,

	Точность	Полнота поиска	f1-score	Представителей
Отрицательные	0.63	0.73	0.67	4295
Нейтральные	0.42	0.26	0.32	4030
Положительные	0.87	0.91	0.89	17088
Среднее / Всего	0.76	0.78	0.76	25413

Таблица 14: Результаты модели голосования

	Отр. (классиф.)	Нейтр. (классиф.)	Полож. (классиф.)
Отр. (реал.)	3115	629	551
Нейтр. (реал.)	1177	1060	1793
Полож. (реал.)	684	829	15575

Таблица 15: Матрица смешения модели голосования на тестовых данных

- вероятности принадлежности к определенному классу от байесовского классификатора,
- вероятности принадлежности к одному из двух фиксированных классов от модели из раздела 3.5,
- значение функций решения $w^T x + b$ от SVM и NB SVM.

Также для модели необходимо было подобрать гиперпараметры η и γ, максимальную глубину деревьев и количество раундов бустинга, и выбрать какие два класса фиксировать и считать вероятности в модели из главы 3.5.

По результатам кросс-валидации лучший f1-score показала модель с $\eta = 0.5$, $\gamma = 5$, максимальной глубиной дерева равной 10 и количеством раундов бустинга 200. Лучше всего оказалось использовать предсказания вероятностей модели, фиксирующей положительный и нейтральный классы и выдающей вероятности по принадлежности к этим классам. Средний f1-score этого ансамбля равен 0.758, с доверительным интервалом (0.754, 0.762).

На рисунке 6 представлена важность признаков у модели, построенной с помощью XGBoost. Интересно заметить, что предсказания всех моделей имеют маленькое значение, а самое большое значение имеет ве-

роятность принадлежности к классу нейтральных от классификатора, описанного в разделе 3.5, и обучающегося только на положительных и нейтральных рецензиях.

5.2.7. Общее сравнение

Итоговые результаты сравнения моделей разного типа между собой можно видеть на диаграммах 7, 8, 9, на которых представленны, соответственно, показатели взвешшенного среднего f1-score, правильности классификации, и макро-усредненного f1-score, что представляет собой невзвешенное среднее f1-score по всем классам.

Лучший средний результат по среднему взвешенному f1-score показал ансамбль из классификаторов, работающий по схеме голосования. Нельзя сказать, что это статистически значимый результат, так как доверительные интервалы пересекаются с доверительными интервалами ансамбля XGBoost и SVM.

Лучший показатель правильности классификации достигла модель, построенная с помощью NB SVM и bag-of-words, причем в данном случае имеется статистическая значимость, по скольку доверительные интервалы не пересекаются с доверительными интервалами остальных моделей, за исключением результата ансамбля по схеме голосования. На показатель правильности классификации, как в случае данного набора рецензий, прямым образом влияет качество классификации на самом большом классе — положительных рецензиях.

В показателе макро-усредненного f1-score, вследствие того, что учитываются показатели f1-score по всем классам без взвешивания, больший вклад имеет качество классификации на малых классах, в данном случае нейтральных и отрицательных рецензиях. Можно сказать, что лучше всего с этой задачей справляются ансамбли и SVM.

В целом можно сказать, что лучше всего на данном наборе данных показывают себя модели, построенные на основе NB SVM и SVM. SVM показывает более сбалансированные показатели по всем классам, а лучшее качество классификации на положительных и отрицательных у NB SVM. Применение ансамблей не дало статически значимого улучшения

результатов по сравнению с этими моделями.

Заключение

В ходе данной работы были достигнуты следующие результаты:

- получен набор кинорецензий для обучения и тестирования алгоритмов классификации,
- построены и сравнены модели классификации рецензий по трем классам эмоциональности.

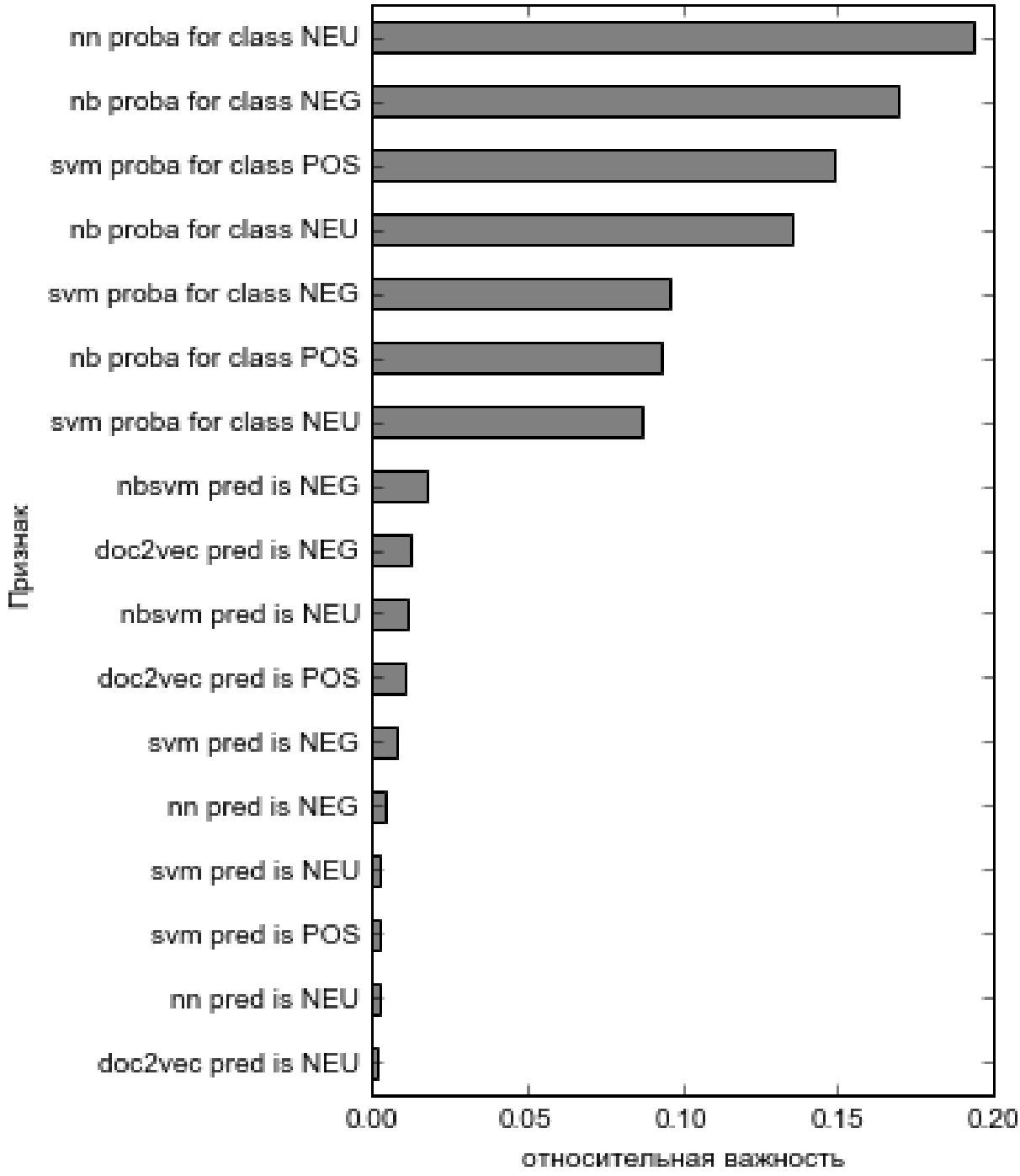

Рис. 6: Важность признаков XGBoost

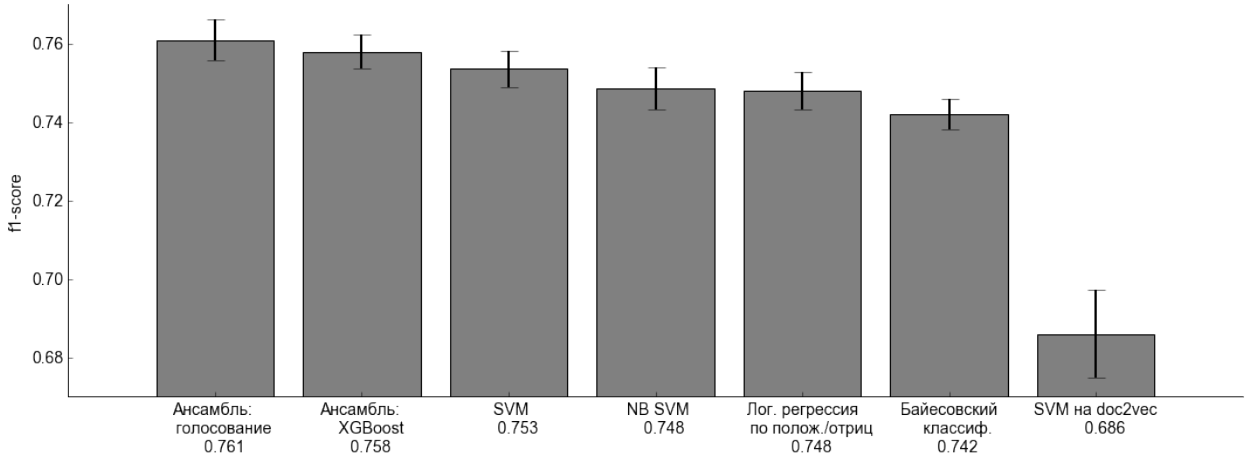

Рис. 7: Показатели f1-score по кросс-валидации

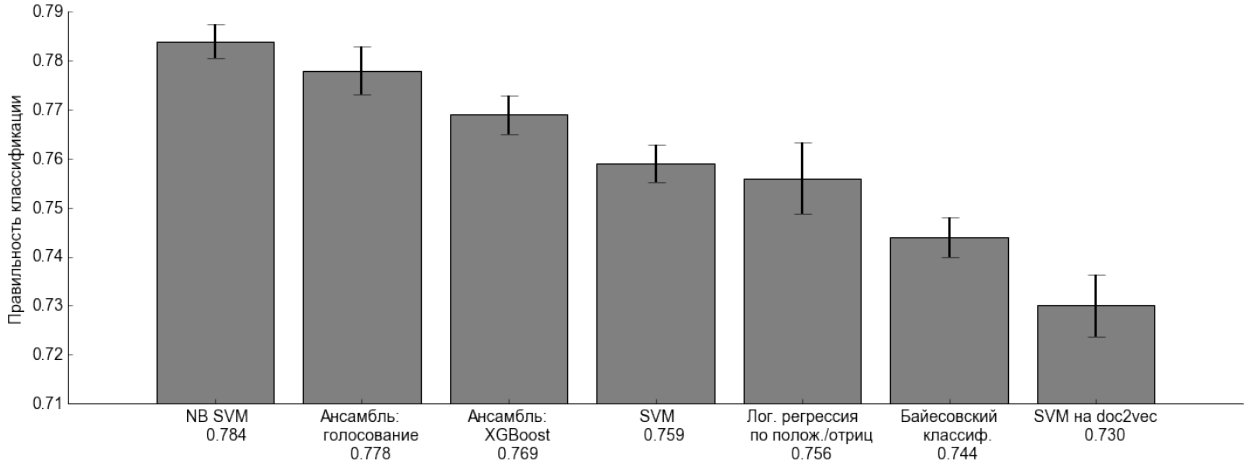

Рис. 8: Правильность классификации по кросс-валидации

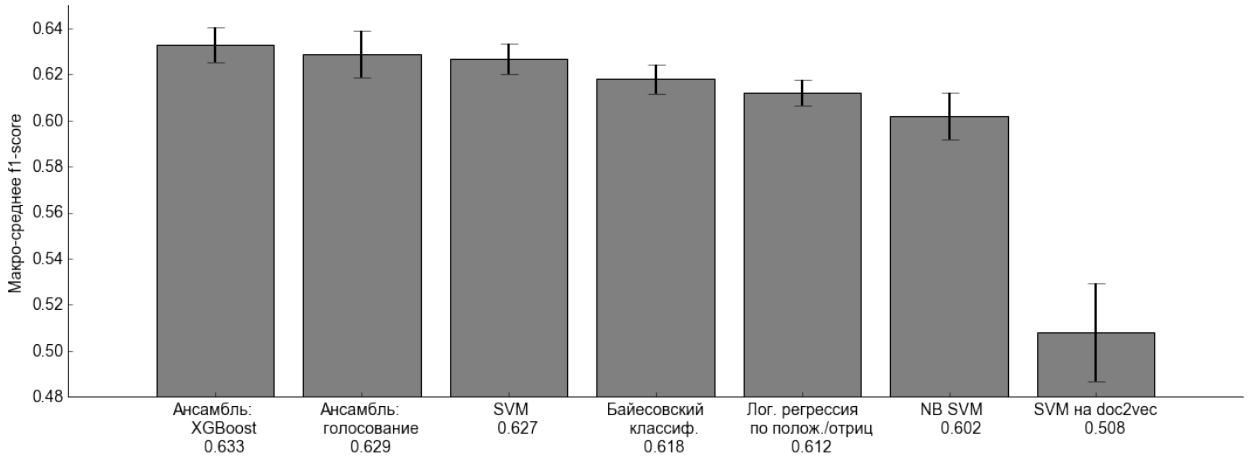

Рис. 9: Макро-усредненный f1-score по кросс-валидации

Список литературы

[1] Abbasi Ahmed, Chen Hsinchun, Salem Arab. Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums // ACM Transactions on Information Systems (TOIS). — 2008. — Vol. 26, no. 3. — P. 12.

[2] Chen Tianqi, Guestrin Carlos. XGBoost: A Scalable Tree Boosting System // arXiv preprint arXiv:1603.02754. — 2016.

[3] Das Sanjiv, Chen Mike. Yahoo! for Amazon: Extracting market sentiment from stock message boards // Proceedings of the Asia Pacific finance association annual conference (APFA) / Bangkok, Thailand. — Vol. 35. — 2001. — P. 43.

[4] Efficient estimation of word representations in vector space / Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean // arXiv preprint arXiv:1301.3781. — 2013.

[5] Glorot Xavier, Bordes Antoine, Bengio Yoshua. Deep sparse rectifier neural networks // International Conference on Artificial Intelligence and Statistics. — 2011. — P. 315–323.

[6] Le Quoc V, Mikolov Tomas. Distributed representations of sentences and documents // arXiv preprint arXiv:1405.4053. — 2014.

[7] Learning word vectors for sentiment analysis / Andrew L Maas, Raymond E Daly, Peter T Pham et al. // Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 / Association for Computational Linguistics. — 2011. — P. 142–150.

[8] Manning Christopher D., Raghavan Prabhakar, Schütze Hinrich. Introduction to Information Retrieval. — Cambridge University Press, 2008.

[9] Narayanan Vivek, Arora Ishan, Bhatia Arjun. Fast and accurate sentiment classification using an enhanced Naive Bayes model // Intelligent Data Engineering and Automated Learning–IDEAL 2013. — Springer, 2013. — P. 194–201.

[10] Pang Bo, Lee Lillian, Vaithyanathan Shivakumar. Thumbs up?: sentiment classification using machine learning techniques // Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 / Association for Computational Linguistics. — 2002. — P. 79–86.

[11] Recursive deep models for semantic compositionality over a sentiment treebank / Richard Socher, Alex Perelygin, Jean Y Wu et al. // Proceedings of the conference on empirical methods in natural language processing (EMNLP) / Citeseer. — Vol. 1631. — 2013. — P. 1642.

[12] Řehůřek Radim, Sojka Petr. Software Framework for Topic Modelling with Large Corpora // Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. — Valletta, Malta : ELRA, 2010. — . — P. 45–50. — http://is.muni.cz/publication/884893/en.

[13] A Review of Feature Extraction in Sentiment Analysis / M Zubair Asghar, Aurangzeb Khan, Shakeel Ahmad, Fazal Masud Kundi. — 2014.

[14] Scikit-learn: Machine Learning in Python / F. Pedregosa, G. Varoquaux, A. Gramfort et al. // Journal of Machine Learning Research. — 2011. — Vol. 12. — P. 2825–2830.

[15] Tsytsarau Mikalai, Palpanas Themis. Survey on mining subjective data on the web // Data Mining and Knowledge Discovery. — 2011, volume=.

[16] Wang Sida, Manning Christopher D. Baselines and bigrams: Simple, good sentiment and topic classification // Proceedings of the 50th

Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2 / Association for Computational Linguistics. — 2012. — P. 90–94.

[17] Zhang Tong. Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms // ICML 2004: PROCEEDINGS OF THE TWENTY-FIRST INTERNATIONAL CONFERENCE ON MACHINE LEARNING. OMNIPRESS. — 2004. — P. 919–926.

[18] dos Santos Cícero Nogueira, Gatti Maira. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. // COLING. — 2014. — P. 69–78.

[19] Набор рецензий с проставленной тональностью. — `https://www.dropbox.com/s/9lbm8cgfvksyfo6/data.csv?dl=0`. — 2016.