Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

МЕТОДЫ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПРОЦЕССА КРАУДСОРСИНГА С СОХРАНЕНИЕМ ЕГО ВОСПРОИЗВОДИМОСТИ

Автор: Катунина Евгения Артёмовна _____

Направление подготовки: 01.04.02 Прикладная математика и информатика

Квалификация: Магистр

Руководитель: Фильченков А.А., к.ф.-м.н. _____

К защите допустить

Руководитель ОП Парфенов В.Г., проф., д.т.н. _____

«____» _____ 20___ г.

Санкт-Петербург, 2019 г.

Студент Катунина Е. А. Группа М4238 Факультет ИТиП

Направленность (профиль), специализация
Технологии разработки программного обеспечения

ВКР принята «____» _____ 20___ г.

Оригинальность ВКР _____%

ВКР выполнена с оценкой _____

Дата защиты «____» _____ 20___ г.

Секретарь ГЭК Павлова О.Н. _____

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

УТВЕРЖДАЮ
Руководитель ОП
проф., д.т.н. Парфенов В.Г. _____
«____» _____ 20___ г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Студент Катунина Е. А. **Группа** М4238 **Факультет** ИТиП

Руководитель Фильченков А.А., к.ф.-м.н., доц. кафедры КТ

1 Наименование темы: Методы повышения эффективности процесса краудсорсинга с сохранением его воспроизводимости

Направление подготовки (специальность): 01.04.02 Прикладная математика и информатика

Направленность (профиль): Технологии разработки программного обеспечения

Квалификация: Магистр

2 Срок сдачи студентом законченной работы: «6» мая 2019 г.

3 Техническое задание и исходные данные к работе

Требуется усовершенствовать агрегацию ответов в краудсорсинге, добавив поведенческие признаки исполнителей (например, время ответа на вопрос, пропуск вопроса, возвращение к вопросу, количество смен ответа) в модель машинного обучения, и предсказать достоверность полученных от пользователей ответов. Модифицировать алгоритм Дэвида-Скина с помощью полученных значений.

4 Содержание выпускной работы (перечень подлежащих разработке вопросов)

а) Обзор предметной области;
б) теоретические исследования;
в) экспериментальная проверка методов решения и их сравнение.

5 Перечень графического материала (с указанием обязательного материала)

Графические материалы и чертежи работой не предусмотрены

6 Исходные материалы и пособия

а) Hung, Nguyen Quoc Viet, et al. ”An evaluation of aggregation techniques in crowdsourcing.”International Conference on Web Information Systems Engineering. Springer, Berlin, Heidelberg, 2013;
б) Allahbakhsh, Mohammad, et al. ”Quality control in crowdsourcing systems: Issues and directions.”IEEE Internet Computing 17.2 (2013): 76-81;
в) Aker, Ahmet, et al. ”Assessing Crowdsourcing Quality through Objective Tasks.”LREC. 2012.

7 Дата выдачи задания «01» сентября 2017 г.

Руководитель ВКР _____

Задание принял к исполнению _____

«01» сентября 2017 г.

Министерство науки и высшего образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

АННОТАЦИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Студент: Катунина Евгения Артёмовна

Наименование темы ВКР: Методы повышения эффективности процесса краудсорсинга с сохранением его воспроизводимости

Наименование организации, где выполнена ВКР: Университет ИТМО

ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования: Разработка метода повышения эффективности краудсорсинга.

2 Задачи, решаемые в ВКР:

а) обзор существующих решений;

б) реализация своего решения, улучшающего имеющиеся результаты;

в) проведение эксперимента для сравнения лучших существующих решений с полученным в ходе выполнения работы.

3 Число источников, использованных при составлении обзора: 21

4 Полное число источников, использованных в работе: 23

5 В том числе источников по годам:

Отечественных			**Иностранных**		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
0	0	0	8	11	4

6 Использование информационных ресурсов Internet: да, число ресурсов: 23

7 Использование современных пакетов компьютерных программ и технологий:

Пакеты компьютерных программ и технологий	**Раздел работы**
Библиотека для построения графиков XChart	Глава 3
Библиотека для парсинга Univocity Parsers	Глава 3

8 Краткая характеристика полученных результатов

Обучив модель машинного обучения случайный лес на данных с собранной статистикой о поведении пользователей, удалось улучшить качество широкоиспользующегося в краудсорсинговых платформах алгоритма Дэвида-Скина на 2%, а также метод голосования большинства на 1.7% с помощью добавления информации о поведении пользователей. Результаты работы могут быть применены в краудсорсинговых платформах.

9 Гранты, полученные при выполнении работы

Отсутствуют.

10 Наличие публикаций и выступлений на конференциях по теме работы

Публикации:

1. Катунина Е. А. Методы повышения качества краудсорсинга // Сборник тезисов докладов конгресса молодых ученых. Электронное издание. 2019.

Конференции:

1. VIII Конгресс молодых учёных, 2019 г., тема доклада: Методы повышения качества краудсорсинга.

Студент Катунина Е. А. _____

Руководитель Фильченков А.А. _____

«____» _____ 20___ г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Крадсорсинг – это метод решения задач с помощью коллективного разума путём разбиения одной большой сложной задачи на много маленьких простых подзадач, выдачи этих подзадач Интернет-пользователям для получения их решений и последующей агрегации полученных от Интернет-пользователей ответов. В настоящее время краудсорсинг активно используется IT-компаниями для сбора обучающих наборов данных для моделей машинного обучения, примерами решаемых с помощью краудсорсинга задач являются:

- распознавание изображений;
- распознавание речи;
- ранжирование страниц поисковой выдачи;
- фильтрация спама, порнографии;
- обнаружение дубликатов и т.д.

Краудсорсинг за пятнадцать лет своего существования использовался не только для развития искусственного интеллекта, примеры других задач, которые пытались решать с помощью краудсорсинга:

- поиски пропавших людей по фотографиям со спутников (краудсорсинг не помог в решении этого типа задач);
- помощь дизайнерам (люди выбирают, какой подход к оформлению чего-либо им больше нравится);
- различные опросы;
- NASA размещает фотографии с космического телескопа и анализирует снимки, на которых исполнители находят интересные объекты;
- полевые задания, выполняемые на местности, например, обновление информации в мобильных картах о часах работы заведений и т.д.

Для размещения заданий и сбора ответов используются краудсорсинговые платформы, например Яндекс.Толока, Amazon Mechanical Turk, Crowdflower, Microworkers. На данных платформах можно зарегистрироваться как «заказчик» и размещать задания, либо как «исполнитель» для выполнения заданий.

Существует множество факторов, влияющих на качество ответов, получаемых с помощью краудсорсинга, среди них можно выделить такие факторы:

а) плохая формулировка задач заказчиком;

б) неудачно подобранный заказчиком размер поощрения за выполнение задач;
в) недобросовестность исполнителей;
г) человеческий фактор (применимо к исполнителям).

В плохую формулировку задач заказчиком входит:

а) отсутствие примеров задач и их решения;
б) расплывчатые формулировки задач;
в) размер задачи (крупные задачи необходимо разбивать на подзадачи) и т.д.

В исследовании [19] сформулирована хорошая рекомендация к составлению заданий: честно ответить на задание не должно быть сложнее, чем выбрать ответ случайно, вопросы должны быть простыми и чёткими.

Согласно исследованию [13], более значительные денежные поощрения по сравнению со стандартно использующимися заказчиками на данной краудсорсинговой платформе позволяют быстрее получить ответы от исполнителей, но качество может быть хуже, чем при умеренном размере денежных поощрений, в то же время слишком низкий размер поощрений также может отражаться на качестве полученных ответов.

Среди известных проблем краудсорсинга можно выделить проблему спама, например, в 2011 году доля спамеров среди исполнителей на платформе Amazon Mechanical Turk составляла целых 39%. Краудсорсинговые платформы позволяют бороться со спамерами с помощью добавления контрольных вопросов, определения минимальной доли правильных ответов на них и недопуска исполнителей, у которых не получилось преодолеть минимальный порог доли правильных ответов, однако у такого подхода к борьбе со спамерами есть очевидные проблемы: вопросы могут быть слишком сложными, и добросовестные участники также могут не справиться с ними, не очевидно, какую долю правильных ответов нужно устанавливать в качестве минимальной. Также у краудсорсинговых платформ есть возможность автоматического отклонения ответов, которые даны исполнителями слишком быстро, либо заказчик может сам отклонить ответ на основе предоставленных ему данных о выполнении задания исполнителем.

Наконец, последний фактор, человеческий, применим к исполнителям, не являющимся спамерами, но желающим больше заработать и выполняющим задания, несмотря на усталость или сомнения в верности своих ответов.

У краудсорсинговых платформ существуют механизмы, позволяющие заказчику наказывать некомпетентных исполнителей: понижать их рейтинг в системе, отклонять их задания, вследствие чего они не получат за них никаких вознаграждений; можно изначально разрешать выполнять задания только исполнителям с определённым минимальным рейтингом или определённым минимальным уровнем экспертизы в данной области. У такого подхода имеются проблемы: заказчики тоже не всегда добросовестные, могут получить ответы от пользователей и отклонить их, чтобы не производить выплаты, также на ручной контроль качества заказчиком требуется его личное время, что не всегда приемлемо для заказчиков.

Ещё одним способом контроля качества ответов является конвейерный подход, разделённый на следующие стадии:

а) первая группа исполнителей отвечает на вопросы, сформулированные заказчиком;
б) вторая группа получает эти вопросы и ответы на них и оценивает, верные ответы или нет;
в) третья группа исполнителей проверяет, верно ли вторая группа оценила ответы первой группы.

Если на втором или третьем этапе обнаружилась ошибка, задания выдаются заново другим исполнителям. Весь процесс продолжается, пока не будет достигнут определённый уровень согласованности ответов.

Идея данной выпускной квалификационной работы состоит в том, чтобы улучшить качество агрегации ответов следующим образом:

а) анализировать поведение исполнителей программным образом, без непосредственного участия заказчика или других исполнителей, например, участник может долго отвечать на задание или наоборот слишком быстро, может много раз менять ответ, а может дать его сразу, может возвращаться к заданию и т.д.;
б) на основе поведенческих признаков построить рейтинг пользователей со значениями уровня доверия к их ответам;

в) модифицировать хорошо зарекомендовавший себя и использующийся во многих краудсорсинговых платформах алгоритм Дэвида-Скина, добавив туда информацию о поведении пользователей.

В первой главе данной работы будет проведён обзор существующих методов контроля качества в краудсорсинге.

Во второй главе будут рассмотрены теоретические аспекты проведённого иследования: признаки для методов машинного обучения, метрика, с помощью которой можно оценить результат, способ модификации алгоритма Дэвида-Скина.

В третьей главе будет описан проведённый эксперимент, приведены детальные результаты, и будет выявлен наилучший из методов решения задачи.

ГЛАВА 1. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Агрегация ответов

Этапами краудсорсинга являются создание заданий, размещение их в системе, получение ответов пользователей (этап разметки), агрегация полученных от пользователей ответов. Последний этап является наиболее наукоёмким и будет рассматриваться в данном разделе.

1.1.1. Постановка задачи

Для того чтобы агрегировать ответы пользователей нужно решить задачу поиска консенсуса, которая формально описывается для краудсорсинга следующим образом [4]. $E = \{e_i\}_{i=1}^{I}$ - набор заданий, задание $e_i = (x_i, y_i)$, где x_i – набор значений признаков, y_i – истинная метка. $U = \{u_j\}_{j=1}^{J}$ – множество пользователей. Метки принадлежат множеству классов $C = \{c_k\}_{j=1}^{K}$. В случае бинарной классификации, которая часто встречается среди задач краудсорсинга, класс c_1 $(k = 1)$ соответствует положительным меткам, а класс c_2 $(k = 2)$ – отрицательным. Всем заданиям i сопоставлено множество меток $l_i = \{l_{ij}\}_{j=1}^{J}$, где элемент l_{ij} получен от участника j. $L = \{l_i\}_{i=1}^{I}$ – матрица аннотаций, где $l_{ij} \in \{c_1,0,c_2\}$, 0 означает, что участник не отвечал на данное задание. Участник j описывается матрицей $N^{(j)} = \{n_{ik}^{j}\}$,$1 \leqslant i \leqslant I$,$1 \leqslant k \leqslant K$, здесь каждый элемент равен количеству раз, когда пользователь j отнёс пример из задания i к классу k. Обычно n_{ik}^{j} равно нулю или единице. Также для всех классов известны априорные вероятности.

Задачи состоит в подборе таких меток $\hat{y}_i$, на которых достигается минимальное значение функции ошибки:

$$R = \frac{1}{I}\sum_{i=1}^{I}\mathbf{I}(\hat{y}_i \neq y_i)$$

при заданных L и $y \in (c_1, c_2)$, где **I** – индикаторная функция.

1.2. Методы агрегации ответов

Методы агрегации ответов можно разделить на итеративные и неитеративные [2]. Неитеративные методы используют эвристики для вычисления агрегированного значения для каждого вопроса по отдельности. Итеративные методы представляют серию итераций. На каждой итерации выполняется два шага обновлений:

а) обновление агрегированных значений для каждого вопроса, основанные на оценке исполнителей;
б) корректировка оценки для каждого исполнителя, основанная на ответах, данных им.

Преимуществами неитеративных методов являются простота реализации и быстрое время выполнения, преимущество итеративных методов – лучшее качество агрегации ответов по сравнению с неитеративными методами.

1.2.1. Неитеративные методы агрегации ответов

Простейшим неитеративным методом агрегации ответов является голосование большинства (Majority Decision, MD) [12]: правильный ответ тот, за который проголосовало больше всего пользователей. Однако в данной модели все пользователи вносят равный вклад в итоговый ответ, что является существенной проблемой.

Метод Honeypot (HP) [11] является улучшением голосования большинства за счёт добавления контрольных вопросов. Однако у данного метода тоже есть проблема: хотя контрольные вопросы и позволяют избавиться от спамеров, добросовестные исполнители тоже могут быть неверно идентифицированы как спамеры, если контрольные вопросы слишком сложные.

Метод ELICE (Expert Label Injected Crowd Estimation) [8] является улучшением Honeypot: добавляется минимальная доля правильных ответов исполнителя на контрольные вопросы, необходимая, чтобы приступить к заданию, также сложность контрольных вопросов оценивается по тому, насколько люди справляются с ними.

1.2.2. Итеративные методы агрегации ответов

Итеративные методы основаны на EM-алгоритме [15], позволяющем найти оценку максимального правдоподобия вероятностных моделей. Алгоритмы, основанные на методе EM, характерны тем, что сначала даётся начальная оценка правильных ответов на задания, затем оптимизируются параметры, эти действия повторяются, пока не будет достигнута определённая сходимость или не будет выполнено максимальное число итераций, то есть данные алгоритмы работают схожим образом и отличаются лишь набором параметров. Например, в методе Дэвида-Скина [5] параметром модели является матрица ошибок участников, в GLAD [23] – степени компетентности участников

и сложности заданий, в SLME [21] используются чувствительность и специфичность. Итеративный метод ITER [7] отличается от остальных тем, что для каждого ответа исполнителя представлена его надёжность, уровень исполнителя оценивается как сумма значений надёжности, взвешенная по сложности вопросов, другие методы представляют надёжность ответов исполнителя как одно число.

1.2.3. Результаты сравнения итеративных и неитеративных методов

В работе [2] описывается сравнение итеративных и неитеративных методов агрегации ответов с помощью фреймворка, созданного авторами, симулирующего разные типы исполнителей (эксперты, обычные, малоопытные, однообразные спамеры (дают один и тот же ответ), случайные спамеры, разные типы вопросов (два варианта ответа или больше). При сравнении методов учитывались такие метрики как время вычислений, точность, устойчивость к спамерам, адаптивность к мультиразметке. Выводы, которые можно сделать из результатов сравнения, таковы:

— В целом, EM и SLME достигают наибольшей точности и надёжно работают против спамеров. В частности, они превосходят другие методы, когда число ответов на вопросы велико.

— Если среди исполнителей много спамеров (от 30 %), лучше использовать SLME или EM. Интересно, что производительность неитеративных методов (MD, HP, ELICE) не значительно меньше SLME и EM. Если высокая точность не требуется, они лучше всего подходят для приложений, требующих быстрых вычислений. Наиболее чувствительны к спамерам GLAD и ITER.

— Только MD, HP, EM могут адаптироваться к мультиразметке. Для двоичной разметки, EM – победитель. В случае четырёх меток MD и HP также подходящие варианты, разница между ними и EM минимальна.

— Для приложений, требующих быстрых вычислений, MD и HP – победители. Напротив, строго не рекомендуется использовать итеративные методы. Время вычислений не только намного больше, чем у неитеративных методов, но и приходится заново вычислять все агрегированные ответы с приходом новых ответов исполнителей.

1.3. Классические алгоритмы агрегации ответов

В данном разделе будут рассмотрены алгоритмы, которые сейчас применяются в большинстве краудсорсинговых платформ, и являются наиболее известными в своём роде.

1.3.1. Алгоритм Дэвида-Скина

Наиболее популярным и часто использующимся в краудсорсинговых платформах является алгоритм Дэвида-Скина. Алгоритм Дэвида-Скина решает задачу поиска консенсуса, описанную вначале главы, следующим образом: для каждого пользователя j имеется матрицей ошибок классификации $\pi_{kl}^{(j)}$, которая задаёт вероятность, что пользователь j выберет класс l для задания, истинный класс которого k. На expectation-шаге по матрице ответов пользователей L алгоритм вычисляет вероятности того, что пример i является объектом класса k (для каждого класса) в соответствии со следующим уравнением:

$$P(\hat{y}_i = c_k|L) = \frac{\prod_{j=1}^{J}\prod_{l=1}^{L}(\pi_{kl}^{(j)})^{n_{il}^{j}}P(c_k)}{\sum_{q=1}^{K}\prod_{j=1}^{J}\prod_{l=1}^{L}(\pi_{ql}^{(j)})^{n_{il}^{j}}P(c_q)}$$

, где $P(c_k)$ и $P(c_q)$ – априорные вероятности того, что пример относится к классам c_k и c_q соответственно, вычисленные на maximization-шаге алгоритма. На maximization-шаге уточняются значения матрицы ошибок классификации:

$$\hat{\pi}_{kl}^{(j)} = \frac{\sum_{i=1}^{I} I(\hat{y}_i = c_k)n_{il}^{(j)}}{\sum_{l=1}^{K}\sum_{i=1}^{I} I(\hat{y}_i = c_k)n_{il}^{(j)}},$$

Также на maximization-шаге уточняются априорные вероятности классов:

$$\hat{P}(c_k) = \frac{1}{I}\sum_{i=1}^{I} I(\hat{y}_i = c_k).$$

Истинные классы примеров неизвестны, поэтому вместо $I(\hat{y}_i = c_k)$ используется математическое ожидание этой величины: $P(\hat{y}_i = c_k|L)$.

Начальное приближение для правильных ответов обычно выбирается с помощью простого голоса большинства, для матрицы ошибок изначально считается, что пользователи идеальны и не допускали ошибок при ответе на вопросы.

1.3.2. Алгоритм GLAD

GLAD моделирует уровень компетентности каждого пользователя ($\alpha_j \in (-\infty, +\infty)$) и сложность каждого задания ($\frac{1}{\beta_i} \in [0, +\infty)$). Чтобы оценить уровень компетентности пользователя j для задания i используется следующая логистическая модель:

$$P(l_{ij} = y_i|\alpha_j,\beta_i) = \frac{1}{1 + e^{-\alpha_j\beta_i}}.$$

На expectation-шаге алгоритм вычисляет апостериорные вероятности для положительных и отрицательных классов для всех примеров по значениям параметров (α, β) с последнего шага maximization и матрицы аннотаций L:

$$P(y_i = +|L,\alpha,\beta) = P(y_i = +|l_i,\alpha,\beta_i) \propto P(y_i = +) \prod_{j=1}^{J} p(l_{ij}|y_i = +,\alpha_j,\beta_i).$$

На maximization-шаге алгоритм максимизирует стандартную функцию Q и обновляет значения двух параметров (α, β), используя алгоритм градиентного спуска следующим образом:

$$Q(\alpha,\beta) = E[\ln P(L,y|\alpha,\beta)] = E[\ln \prod_{i=1}^{I}(P(y_i = +) \prod_{j=1}^{J} P(l_{ij}|y_i = +,\alpha_j, \beta_i))],$$

где y содержит предполагаемые метки для всех заданий.

1.3.3. Алгоритм Raykar, Yu (RY)

Алгоритм RY [9] моделирует чувствительность (α_j) и специфичность (β_j) пользователя j. В случае двоичных меток чувствительность определяет смещение (bias) в сторону положительного класса, а специфичность – смещение в сторону отрицательного класса.

RY использует байесовский подход к оценке априорной вероятности параметров (α_j,β_j) и положительного класса:

$$P(\alpha_j|a_j^+,a_j^-) = Beta(\alpha_j|a_j^+,a_j^-),$$

$$P(\beta_j|b_j^+,b_j^-) = Beta(\beta_j|b_j^+,b_j^-),$$

$$P(p^+|n^+,n^-) = Beta(p^+|n^+,n^-),$$

где a_j и a_j^- – число положительных и отрицательных меток соответственно, выданных пользователем j для положительного класса, рассматриваемого в данный момент; b_j^+ и b_j^- – число положительных и отрицательных меток соответственно, выданных пользователем j для отрицательного класса, рассматриваемого в данный момент; n^+ и n^- – число положительных и отрицательных меток, соответственно, выданных всеми пользователями для всех примеров. $Beta$ – это β-функция вероятностного распределения.

На expectation-шаге по примеру x_i, матрице аннотаций L, двум параметрам (α,β) и априорной вероятности положительного класса p^+ RY вычисляет вероятность принадлежности каждого примера i к положительному классу:

$$\mu_i = P(y_i = +|x_i,L,\alpha,\beta,p^+) \propto \frac{p^+ a_i}{p^+ a_i + (1-p^+)b_i},$$

где

$$a_i = \prod_{j=1}^{J} (\alpha_j)^{\mathbf{I}(L_{ij}=+)} (1-\alpha_j)^{\mathbf{I}(L_{ij}=-)}$$

$$b_i = \prod_{j=1}^{J} (\beta_j)^{\mathbf{I}(L_{ij}=-)} (1-\beta_j)^{\mathbf{I}(L_{ij}=+)}.$$

На maximization-шаге RY обновляет параметры (α_j,β_j) пользователя j и априорную вероятность положительного класса p^+ в соответствии с разметкой, проведённой пользователем j $(\sum_{i=1}^{I} l_{ij})$:

$$\alpha_j = \frac{a_j^+ - 1 + \sum_{i=1}^{I} \mu_i l_{ij}}{a_j^+ + a_j^- - 2 - \sum_{i=1}^{I} \mu_i}$$

$$\beta_j = \frac{b_j^+ - 1 + \sum_{i=1}^{I} (1-\mu_i)(1-l_{ij})}{b_j^+ - b_j^- - 2 + \sum_{i=1}^{I} (1-\mu_i)}$$

$$p^+ = n^+ - 1 + \frac{\sum_{i=1}^{I} \mu_i}{n^+ - n^- - 2 + I}.$$

1.3.4. Алгоритм ZenCrowd

ZenCrowd [6] использует только один двоичный параметр $good, bad$ для моделирования надёжности пользователя. Это сложнее, чем голосование боль-

шинства, но проще чем остальные описанные выше методы. Метод используется для решения задачи связывания сущностей для крупных коллекций вебстраниц.

На expectation-шаге по всей разметке от каждого пользователя j ($\sum_{i=1}^{I} l_{ij}$) ZenCrowd вычисляет надёжность пользователя j:

$$P(u_j = reliable) = \frac{\sum_{i=1}^{I} \mathbf{I}(l_{ij} = y_i)}{\sum_{k=1}^{K} \sum_{i=1}^{I} n_{ik}}.$$

На maximization-шаге ZenCrowd использует надёжности пользователей, чтобы обновить вероятность принадлежности примера i классу c_k:

$$P(y_i = c_k) = \frac{\prod_{j=1}^{J} [P(u_j = reliable)]^{\mathbf{I}(y_i = c_k)}}{\sum_{k=1}^{K} \prod_{k=1}^{w} [P(u_j = reliable)]^{\mathbf{I}(y_i = c_k)}}.$$

1.4. Последние исследования в области агрегации ответов

Ранее были рассмотрены классические методы агрегации ответов, которые используются уже давно. Последние исследования в этой области связаны с тем, что классические алгоритмы всё же не очень хорошо справляются со смещённой оценкой пользователей (например, в заданиях, где нужно определить, содержит ли видео элементы порнографии, люди, у которых есть дети, чаще относят примеры к содержащим порнографию), а также они не учитывают человеческий фактор (усталость и потерю концентрации).

В работе [20] авторы модифицируют алгоритм Дэвида-Скина, смещая метку, которую поставил пользователь, к истинной, и утверждают, что им удалось улучшить результаты. В статье [4] авторы утверждают, что взвешенный голос большинства, в котором в качестве весов используются оценки смещения по частоте положительных и отрицательных меток от пользователей, в целом, имеет лучшую производительность, чем все четыре классических алгоритма, описанные выше и основанные на Expectation-Maximization алгоритме.

В работе [3] авторы добавляют проверочные задания с известным результатом во время разметки заданий пользователем, чтобы определить момент, когда он устал, и прекратить выдачу ему заданий. В [16] описано использование техники прайминга (фиксирования установки), пользователям показывались определённые изображения или включались аудиозаписи, создаю-

щие определённое эмоциональное состояние, данный метод позволил улучшить полученные результаты. В статье [22] упомянут такой способ улучшения качества краудсорсинга как добавление на страницу с заданием, ссылок, описывающих специфичные термины, однако не представлены результаты применения этого метода.

В работе [10] представлены методы отбора лучших вариантов из заданного набора. В работах [17] и [18] предлагаются методы очистки зашумлённых данных с помощью обученных классификаторов, то есть классификаторы также выступают в роли пользователей краудсорсинговых платформ, выполняющих задания. В статье [1] используют map-reduce для больших данных и улучшают качество агрегации ответов. В исследовании [14] авторы предлагают четыре новых модели, основанные на байесовских методах, и утверждают, что их модели дают лучшие результаты, чем широко применяющиеся сейчас.

Выводы по главе 1

В главе 1 были рассмотрены разные типы алгоритмов агрегации ответов в краудсорсинге, было подробно рассмотрено четыре классических алгоритма, которые часто используются в современных краудсорсинговых платформах (особенно алгоритм Дэвида-Скина). Также были рассмотрены современные научные работы, в которых авторы предлагают новые методы и утверждают, что они работают лучше классических, а также предлагают способы улучшения качества с существующими алгоритмами, в том числе онлайн-решения, реализуемые ещё на этапе разметки.

ГЛАВА 2. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ИССЛЕДОВАНИЯ

2.1. Признаки для применения машинного обучения

Были составлены следующие признаки, хорошо описывающие поведение пользователя:

а) время ответа на вопрос;
б) число смен ответа;
в) число выбранных вариантов ответа;
г) время между выбором окончательного ответа и его подтверждением;
д) среднее время ответа для данного исполнителя;
е) среднее время ответа для данного вопроса;
ж) время бездействия;
и) время с начала сессии.

Признак «время ответа на вопрос» в комбинации с признаками «среднее время ответа для данного исполнителя» и «среднее время ответа для данного вопроса» позволяет определить, когда пользователь очень быстро отвечает, скорее всего, не подумав, или наоборот очень сомневается в ответе и отвечает слишком долго. Признак «число смен ответа» описывает сомнения пользователя при выборе ответа, учитывается наведение курсора то на один ответ, то на другой, выбор ответа, смена ответа на другой. Признак «число выбранных вариантов» ответа описывает сколько разных вариантов ответа попало под курсор или было выбрано. Среднее значение времени ответа для данного исполнителя позволяет оценивать насколько типичным для него является ответ на один вопрос за то или иное время. Среднее время ответа для данного вопроса позволяет оценить сложность вопроса. Время бездействия позволяет учитывать ситуацию, когда пользователь отвлёкся, например, ушёл пить кофе. Время с начала сессии позволяет оценить, насколько пользователь устал, а следовательно, упала его концентрация. Признак «время между выбором окончательного ответа и его подтверждением» описывает степень сомнения в ответе.

2.2. Метод машинного обучения

Предполагается использовать в качестве обучающего множества признаковое описание ответов пользователей ($F_1(x_i^{(j)}) \dots F_z(x_i^{(j)})$, где F_z – функция, извлекающая признак под номером z, x_i^j – ответ пользователя под номером j на задание с номером i), которому сопоставлены метки $y_i = 1$, если ответ совпал с истинным, $y_i = 0$ в противном случае, обучить «случайный лес» и получить

для каждого ответа уверенность в том, что он правильный ($score_i^{(j)}$, значения $score_i^j$ вещественные от нуля до единицы), затем на основе этих данных вычислить оценку для каждого пользователя по среднему значению уверенности в его ответах ($user^{(j)} = \frac{\sum_{i=1}^{N} score_i^j}{N}$, где N – число заданий, на которые ответил пользователь, $user^{(j)}$ также принимает вещественные значения от нуля до единицы).

Было решено использовать в качестве метода машинного обучения «случайный лес», так как он традиционно показывает более хорошие результаты для задач регрессии, нежели остальные методы. Количество деревьев в случайном лесе – 200, глубина деревьев не ограничена.

2.2.1. Оценка качества модели машинного обучения

Для оценки качества обучения будут использованы матрица ошибок классификации, точность, полнота и $F1$-мера отдельно для каждого класса и в среднем, бинарная логистическая функция потерь, квадратный корень из среднеквадратического отклонения между рассчитанными для пользователей значениями $user$ и долей правильных ответов пользователей по всем вопросам q, где для пользователя j $q_j = \frac{\sum_{i=1}^{N} I(a_u^i, a_c^i)}{N}$, где a_u^i – ответ пользователя u на задание i, a_c^i – правильный ответ на задание i, I – возвращает 1, если аргументы равны, и 0, в противном случае.

Квадратный корень из среднеквадратической ошибки:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}.$$

Бинарная логистическая функция потерь:

$$logloss = -\frac{1}{l} \cdot \sum_{i=1}^{l}(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)),$$

где $\hat{y}$ – ответ алгоритма на i-м объекте, y – истинная метка класса на i-ом объекте, а l - размер выборки.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

, где F_1 - F_1-мера, $recall = \frac{TP}{TP+FN}$, $precision = \frac{TP}{TP+FP}$, TP $(True\ Positive)$ – число верно классифицированных как положительные объектов,

FP $(False\ Positive)$ – число неверно класифицированных как положительные объектов, FN $(False\ Negative)$ – число неверно класифицированных как отрицательные объектов, TN $(True\ Negative)$ – число верно классифицированных как отрицательные объектов. F_1-мера, $precision$ и $recall$ оцениваются для каждого класса по отдельности, затем эти метрики качества классификации, рассчитанные для каждого класса по отдельности, суммируются и дают оценку качества классификации в целом.

Матрица ошибок классификации представлена в таблице 1.

Таблица 1 – Матрица ошибок классификации

–	$y = 1$	$y = 0$
$\hat{y} = 1$	TP	FP
$\hat{y} = 0$	FN	TN

Для оценки важности признаков будет использоваться устройство случайного леса, чем выше узел с признаком в дереве, тем большую часть выборки он делит пополам, и, соответственно, вносит больший вклад в качество бинарной классификации.

2.3. Рассматриваемые способы улучшения качества агрегации ответов

В данном разделе будут рассмотрены модификации с добавлением информации о поведении пользователей для базового алгоритма, голосования большинства, и для одного из наиболее часто применяющихся при агрегации ответов в краудсорсинговых платформ алгоритма, алгоритма Дэвида-Скина. Будет рассмотрен способ скрещивания этих методов.

2.3.1. Модификация алгоритма Дэвида-Скина

Оригинальный алгоритм Дэвида-Скина вычисляет вероятности того, что пример i является объектом класса k (для каждого класса) в соответствии со следующим уравнением:

$$P(\hat{y}_i = c_k|L) = \frac{\prod_{j=1}^{J}\prod_{l=1}^{L}(\pi_{kl}^{(j)})^{n_{il}^j}P(c_k)}{\sum_{q=1}^{K}\prod_{j=1}^{J}\prod_{l=1}^{L}(\pi_{ql}^{(j)})^{n_{il}^j}P(c_q)}.$$

Добавим в это уравнение матрицу достоверности ответов B, учитывая, что B не зависит от L:

$$P(\hat{y}_i = c_k|L, B) = \frac{\prod_{j=1}^{J}\prod_{l=1}^{L}(\pi_{kl}^{(j)} \cdot b_{ikl}^{(j)})^{n_{il}^j} P(c_k)}{\sum_{q=1}^{K}\prod_{j=1}^{J}\prod_{l=1}^{L}(\pi_{ql}^{(j)} \cdot b_{iql}^{(j)})^{n_{il}^j} P(c_q)},$$

где $b_{ikl}^{(j)} = score_i^j$ при $k = l$, иначе $b_{ikl}^{(j)} = 1 - score_i^{(j)}$.

В качестве второго варианта модификации рассмотрим следующее уравнение:

$$P(\hat{y}_i = c_k|L, U) = \frac{\prod_{j=1}^{J}\prod_{l=1}^{L}(\pi_{kl}^{(j)} \cdot u_{kl}^{j})^{n_{il}^j} P(c_k)}{\sum_{q=1}^{K}\prod_{j=1}^{J}\prod_{l=1}^{L}(\pi_{ql}^{(j)} \cdot u_{ql}^{j})^{n_{il}^j} P(c_q)},$$

где U – матрица доверия к пользователям, $u_{kl}^{(j)} = user^j$ при $k = l$, иначе $u_{kl}^{(j)} = 1 - user^{(j)}$.

2.3.2. Взвешенный голос большинства

В обычном голосовании большинства веса ответов равны, и никак не учитывается их достоверность. Можно получить агрегированные ответы по тому же принципу, что и в обычном голосовании большинства, но использовать в качестве весов предсказания нашей модели машинного обучения для каждого ответа.

$$y_i = argmax_{c\in[1,\ldots,J]}(\sum_{j=1}^{K} w_j \mathbf{1}(y_i^j = c)),$$

где y_i – агрегированный ответ под номером i, c – метка класса, J – количество возможных меток, K – количество голосов, w_j – вес голоса j.

2.3.3. Взвешенный голос большинства с добавлением матрицы ошибок Дэвида-Скина

Можно попробовать улучшить результаты, добавив к исследуемым поведенческим признакам частоту ошибок из алгоритма Дэвида-Скина, которая для каждого пользователя оценивается как сумма недиагональных элементов матрицы ошибок классификации:

$$ER_j = \sum_{l=1}^{L}\sum_{k=1}^{L}(\pi_{kl}^{(j)}) - \sum_{k=1}^{L}(\pi_{kk}^{(j)}),$$

где ER_j – частота ошибок пользователя j.

2.4. Метрики качества агрегации ответов

Как правило, выходными данными алгоритма агрегации является помимо самих агрегированных ответов, уверенность в том, что они верные (от 0 до 1). Для оценки качества методов агрегации можно использовать бинарную логистическую функцию потерь:

$$Q = -\frac{1}{N} \cdot \sum_{i=1}^{N} (-y_i \log(p_i) + (1 - y_i) \log(1 - p_i)),$$

где Q – качество агрегации, N – число заданий, p_i – вероятность того, что агрегированный ответ на задание i верный, $y_i = 0$, если агрегированный ответ на задание i неверный и $y_i = 1$ в противном случае. Чем ближе к нулю полученное значение, тем лучше.

Когда задачи простые, и в их решениях у отдельных пользователей не может быть разногласий, все агрегированные ответы могут получиться верными. В этом случае можно посчитать качество метода агрегации следующим образом:

$$Q = \frac{\sum_{i=1}^{n} p(a_i)}{n},$$

где Q – качество агрегации, n – число заданий, a_i – агрегированный ответ для задания под номером i, p – вероятность того, что ответ верный. В данном случае, наоборот, чем ближе полученное значение к единице, тем лучше качество агрегации.

Выводы по главе 2

Во второй главе были представлены поведенческие признаки, которые будут использоваться для построения модели оценки достоверности ответов пользователей с помощью метода машинного обучения «случайный лес», хорошо зарекомендовавшего себя для решения задач регрессии и классификации, способы проверки качества обученной модели классификации и оценки важности признаков. Были рассмотрены способы модификации алгоритма Дэвида-Скина, широко использующегося в краудсорсинговых платформах и традиционно демонстрирующего хорошее качество агрегации ответов, также были рассмотрены способы улучшить простое голосование большинства взвешиванием с поведенческими признаками и добавлением такого признака как

частота ошибок пользователей, полученного из модели Дэвида-Скина. Были рассмотрены метрики, которые позволят оценить качество агрегации ответов в случае, если все агрегированные ответы верны, и в случае, когда есть неверные ответы.

ГЛАВА 3. ОПИСАНИЕ ЭКСПЕРИМЕНТА И РЕЗУЛЬТАТОВ

3.1. Описание набора данных

Для проведения эксперимента был выбран набор данных «Russe'2018: Human-Annotated Sense-Disambiguated Word Contexts for Russian». Этот набор данных содержит идентифицированные людьми значения слов для 2562 контекстов из 20 слов, используемых в задаче RUSSE'2018 по устранению неоднозначности в смыслах слов для русского языка. Помимо ответов пользователей в наборе данных присутствует статистика о поведении пользователей, отражены действия в UI и время для каждого из них. Исследование проводилось с помощью краудсорсинговой платформы Яндекс.Толока.

Пример задания: дан отрывок из текста

> «..., и провалов, к счастью, не было. Правда, в некоторых случаях наши рекламные акции могут закончиться судебным разбирательством. Но иногда бренду, для того чтобы себя эффективно...»,

нужно оценить значение слова «акции» в данном контексте, варианты ответов:

а) «ценная бумага, выпускаемая акционерным обществом»;

б) «действие, выступление кого-либо, предпринимаемое для достижения какой-либо цели».

На рисунках 1 и 2 представлена часть набора данных. Поля данных (слева направо):

— «INPUT:id» – номер задания;

— «INPUT:left» – часть текста до слова, значение которого нужно оценить;

— «INPUT:word» – слово, значение которого нужно оценить, в том падеже и числе, в котором оно употребляется в отрывке из текста;

INPUT:id	INPUT:left	INPUT:word	INPUT:lemma	INPUT:right	INPUT:senses	OUTPUT:activity	OUTPUT:sense_id	GOLDEN:activity	GOLDEN:sense_id	HINT:text	ASSIGNMENT:link	ASSIGNMENT:assignment_id	ASSIGNME	ASSIGNME	ASSIGNMENT:
146	», и провал	акции	акция	могут зако	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	0253445b-e40b-4de1-84eb-d(	3b737352(	APPROVEI	2017-12-01T1
961	прошлый р	домино	домино	: они играл	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	0253445b-e40b-4de1-84eb-d(	3b737352(	APPROVEI	2017-12-01T1
2558	шаха черн	шахов	шах	, делают ма	[{"sense":1,"de	["2017-12-01T16:	2		2		https://toloka.yar	0253445b-e40b-4de1-84eb-d(	3b737352(	APPROVEI	2017-12-01T1
268	Старкове с	БАЙКИ	байка	О БРАТЕ «Я	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	0253445b-e40b-4de1-84eb-d(	3b737352(	APPROVEI	2017-12-01T1
411	ягодки; ли	гвоздики	гвоздика	, корицы на	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	0253445b-e40b-4de1-84eb-d(	3b737352(	APPROVEI	2017-12-01T1
1556	будет внес	крон	крона	. В деревне	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	0253445b-e40b-4de1-84eb-d(	3b737352(	APPROVEI	2017-12-01T1
2301	* Лодка. С	таза	таз	. Обхватите	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	0253445b-e40b-4de1-84eb-d(	3b737352(	APPROVEI	2017-12-01T1
2165	? Может, в	стопке	стопка	. -- Придет,	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	0253445b-e40b-4de1-84eb-d(	3b737352(	APPROVEI	2017-12-01T1
504	рекоменде	Гипербола	гипербола	в эпическо	[{"sense":1,"de	["2017-12-01T16:	1				https://toloka.yar	0253445b-e40b-4de1-84eb-d(	3b737352(	APPROVEI	2017-12-01T1
528	поэта, обра	гиперболам	гипербола	и мистифи	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	0253445b-e40b-4de1-84eb-d(	3b737352(	APPROVEI	2017-12-01T1
423	слив вынут	гвоздикой	гвоздика	, корицей и	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f	APPROVEI	2017-12-01T1
2266	вкусные ма	тазы	таз	, полные с	[{"sense":1,"de	["2017-12-01T16:	1				https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f	APPROVEI	2017-12-01T1
2039	расспраши	слогом	слог	рассказыва	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f	APPROVEI	2017-12-01T1
2162	дожидаясь	Стопка	стопка	за стопкой,	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f	APPROVEI	2017-12-01T1
1317	правый по	карьером	карьер	пошел вдо	[{"sense":1,"de	["2017-12-01T16:	1				https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f	APPROVEI	2017-12-01T1
2155	. Так и про	стопке	стопка	выпили, хо	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f	APPROVEI	2017-12-01T1
758	чёрный с б	гусеница	гусеница	чёрная, с т	[{"sense":1,"de	["2017-12-01T16:	1				https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f	APPROVEI	2017-12-01T1
1675	ребенку??:	крупом	круп	-ему тогда	[{"sense":1,"de	["2017-12-01T16:	2				https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f	APPROVEI	2017-12-01T1
1751	размягчён	мандарины	мандарин	, муку, соль	[{"sense":1,"de	["2017-12-01T16:	1				https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f	APPROVEI	2017-12-01T1
588	числа наст	градом	град	. Буря сия	[{"sense":1,"de	["2017-12-01T16:	1		1		https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f	APPROVEI	2017-12-01T1

Рисунок 1 – Набор данных, часть 1

INPUT:left	INPUT:word	INPUT:lemma	INPU	ASSIGNMENT:link	ASSIGNMENT:assignment_id	ASSIGNMENT:worker_id	ASSIGNMENT:status	ASSIGNMENT:started
», и провал	акции	акция	мог	https://toloka.yar	0253445b-e40b-4de1-84eb-d0	3b73735298ddfad1d9cdf45a	APPROVED	2017-12-01T16:02:32.509
прошлый р	домино	домино	: они	https://toloka.yar	0253445b-e40b-4de1-84eb-d0	3b73735298ddfad1d9cdf45a	APPROVED	2017-12-01T16:02:32.509
шаха черны	шахов	шах	, дел	https://toloka.yar	0253445b-e40b-4de1-84eb-d0	3b73735298ddfad1d9cdf45a	APPROVED	2017-12-01T16:02:32.509
Старкове с	БАЙКИ	байка	О БР	https://toloka.yar	0253445b-e40b-4de1-84eb-d0	3b73735298ddfad1d9cdf45a	APPROVED	2017-12-01T16:02:32.509
ягодки; ли	гвоздики	гвоздика	, кор	https://toloka.yar	0253445b-e40b-4de1-84eb-d0	3b73735298ddfad1d9cdf45a	APPROVED	2017-12-01T16:02:32.509
будет внес	крон	крона	. В д	https://toloka.yar	0253445b-e40b-4de1-84eb-d0	3b73735298ddfad1d9cdf45a	APPROVED	2017-12-01T16:02:32.509
* Лодка. С	таза	таз	. Об	https://toloka.yar	0253445b-e40b-4de1-84eb-d0	3b73735298ddfad1d9cdf45a	APPROVED	2017-12-01T16:02:32.509
? Может, в	стопке	стопка	. -- Г	https://toloka.yar	0253445b-e40b-4de1-84eb-d0	3b73735298ddfad1d9cdf45a	APPROVED	2017-12-01T16:02:32.509
рекоменде	Гипербола	гипербола	в эп	https://toloka.yar	0253445b-e40b-4de1-84eb-d0	3b73735298ddfad1d9cdf45a	APPROVED	2017-12-01T16:02:32.509
поэта, обра	гиперболам	гипербола	и ми	https://toloka.yar	0253445b-e40b-4de1-84eb-d0	3b73735298ddfad1d9cdf45a	APPROVED	2017-12-01T16:02:32.509
слив вынут	гвоздикой	гвоздика	, кор	https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f7ec6bfb150495b6	APPROVED	2017-12-01T16:03:36.613
вкусные мя	тазы	таз	, пол	https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f7ec6bfb150495b6	APPROVED	2017-12-01T16:03:36.613
расспраши	слогом	слог	расс	https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f7ec6bfb150495b6	APPROVED	2017-12-01T16:03:36.613
дожидаясь	Стопка	стопка	за ст	https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f7ec6bfb150495b6	APPROVED	2017-12-01T16:03:36.613
правый по	карьером	карьер	пош	https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f7ec6bfb150495b6	APPROVED	2017-12-01T16:03:36.613
. Так и про	стопке	стопка	выпи	https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f7ec6bfb150495b6	APPROVED	2017-12-01T16:03:36.613
чёрный с б	гусеница	гусеница	чёрн	https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f7ec6bfb150495b6	APPROVED	2017-12-01T16:03:36.613
ребенку???	крупом	круп	-ему	https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f7ec6bfb150495b6	APPROVED	2017-12-01T16:03:36.613
размягчён	мандарины	мандарин	, мул	https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f7ec6bfb150495b6	APPROVED	2017-12-01T16:03:36.613
числа наст	градом	град	. Бур	https://toloka.yar	35a116e3-8d8c-4fae-9c38-58	f9b02596f7ec6bfb150495b6	APPROVED	2017-12-01T16:03:36.613

Рисунок 2 – Набор данных, часть 2

— «INPUT:lemma» – слово, значение которого нужно оценить в именительном падеже единственном числе;
— «INPUT:right» – часть текста после слова, значение которого нужно оценить;
— «INPUT:senses» – значения слова, из которых нужно выбрать;
— «OUTPUT:activity» – взаимодействие пользователя с UI веб-страницы при ответе на вопрос, данные представлены в формате JSON;
— «OUTPUT:sense_id» – ответ, выбранный пользователем;
— «GOLDEN:activity» – автоматически сгенерировалось Яндекс.Толокой, поле не имеет смысла и всегда пустое;
— «GOLDEN:sense_id» – если вопрос контрольный, правильный ответ, иначе поле пустое;
— «HINT:text» – подсказок к заданиям не было, поэтому данное поле всегда пустое;
— «ASSIGNMENT:link» – ссылка на страницу с заданиями;
— «ASSIGNMENT:assignment:id» – идентификатор страницы с заданиями;
— «ASSIGNMENT:worker_id» – идентификатор пользователя;
— «ASSIGNMENT:status» – статус задания, принято заказчиком или отклонено;
— «ASSIGNMENT:started» – время и дата, когда отрендерилась страница с заданиями.

Все задания были приняты заказчиком (имели статус «APPROVED»). Заданий было 2562, пользователей 116, в среднем на пользователя приходилось

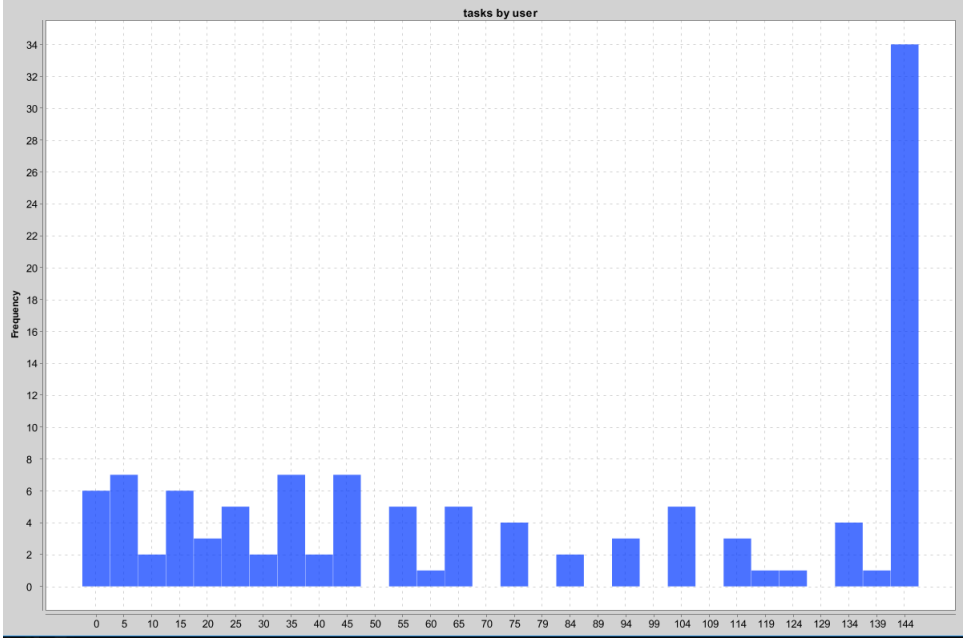

Рисунок 3 – Гистограмма числа заданий, выполненных пользователями

по 83 задания. Гистограмма с распределением по количеству заданий на пользователя представлена на рисунке 3.

3.2. Извлечение признаков

Пример данных об активности пользователя при ответе на вопрос представлен на рисунке 14. Изначально JSON разбивался на список строк, содержащихся внутри квадратных скобок, то есть на отдельные действия в UI. Первое событие (страница с заданиями отрендерилась) никак не учитывалось. Время ответа на вопрос рассчитывалось как разница между временем для последнего действия и второго. Время после подтверждения ответа на вопрос рассчитывалось как разница между временем для последнего действия и действия «click». Число смен ответа оценивалось по количеству действий «mouseenter». Число вариантов ответа рассчитывалось по количеству разных значений у «answer». Время бездействия оценивалось как сумма периодов времени, когда никаких действий в UI не происходило дольше 20 секунд.

Ответы, для которых по техническим причинам не было собрано достаточно данных об активности (была собрана не полностью, либо было только первое действие, отрисовка страницы) не учитывались и по ним не извлекались признаки.

На гистограммах 5-13 представлены данные о распределении значений неотнормированных признаков (для подачи на вход алгоритму машинного

```
["2017-12-01T15:54:47.187Z"\,"render"\,{}],
["2017-12-01T15:55:01.741Z"\,"mouseenter"\,{"target":"task"}],
["2017-12-01T15:55:02.042Z"\,"mouseleave"\,{"target":"task"}],
["2017-12-01T15:55:05.658Z"\,"mouseenter"\,{"target":"task"}],
["2017-12-01T15:55:05.725Z"\,"mouseenter"\,{"answer":"2"\,"target":"answer"}],
["2017-12-01T15:55:05.891Z"\,"mouseleave"\,{"answer":"2"\,"target":"answer"}],
["2017-12-01T15:55:05.891Z"\,"mouseenter"\,{"answer":"1"\,"target":"answer"}],
["2017-12-01T15:55:06.520Z"\,"click"\,{"target":"answer"}],
["2017-12-01T15:55:06.751Z"\,"mouseleave"\,{"answer":"1"\,"target":"answer"}],
["2017-12-01T15:55:06.752Z"\,"mouseleave"\,{"target":"task"}]
```


Рисунок 4 – Пример данных об активности пользователя при ответе на вопрос

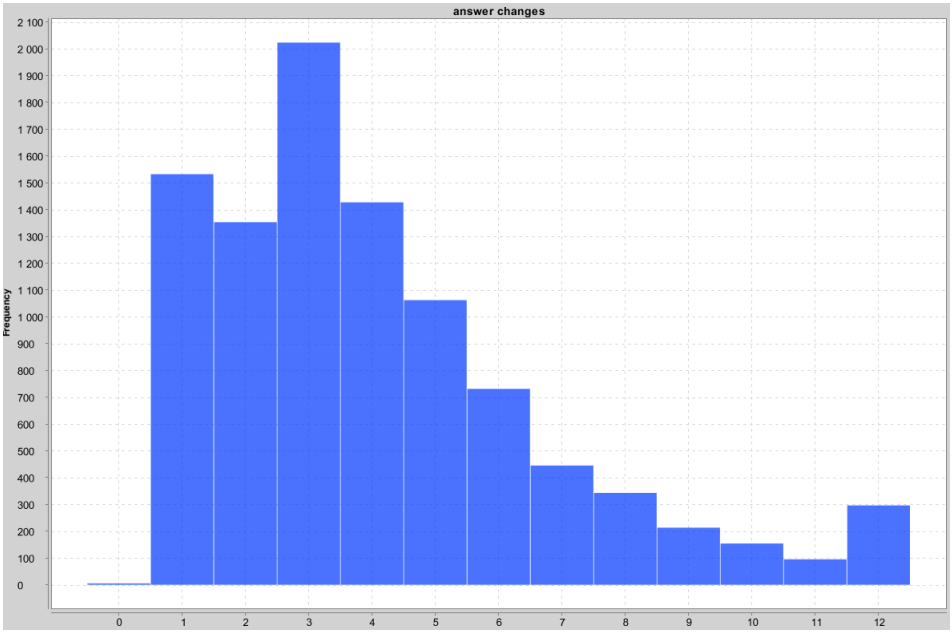

Рисунок 5 – Число смен ответа. Для выбросов, портящих нормировку признака, было принято значение 13.

обучения признаки нормировались от 0 до 1). В данные о некоторых признаках были внесены изменения, так как исходные данные портили нормировку признаков, были выбросы со слишком большими значениями, значения, принятые для выбросов, отмечены на подписях к рисункам.

3.3. Работа с данными

Вопросы, которые задавались пользователям, были простыми. В наборе данных содержалось 396 заданий, для которых был хотя бы один неверный ответ, и 2166 заданий, для которых были только правильные ответы от пользователей. Поэтому для дальнейшего применения машинного обучения набор данных был разбит по заданиям на два множества:

а) множество заданий, для которых встречались неверные ответы;

б) множество заданий, для которых все ответы были верными.

Для обучения модели и получения предсказаний использовалась десятифолдовая кросс-валидация: каждое из десяти тестовых множеств содержало все ответы на примерно 40 заданий, для которых были неправильные ответы, и

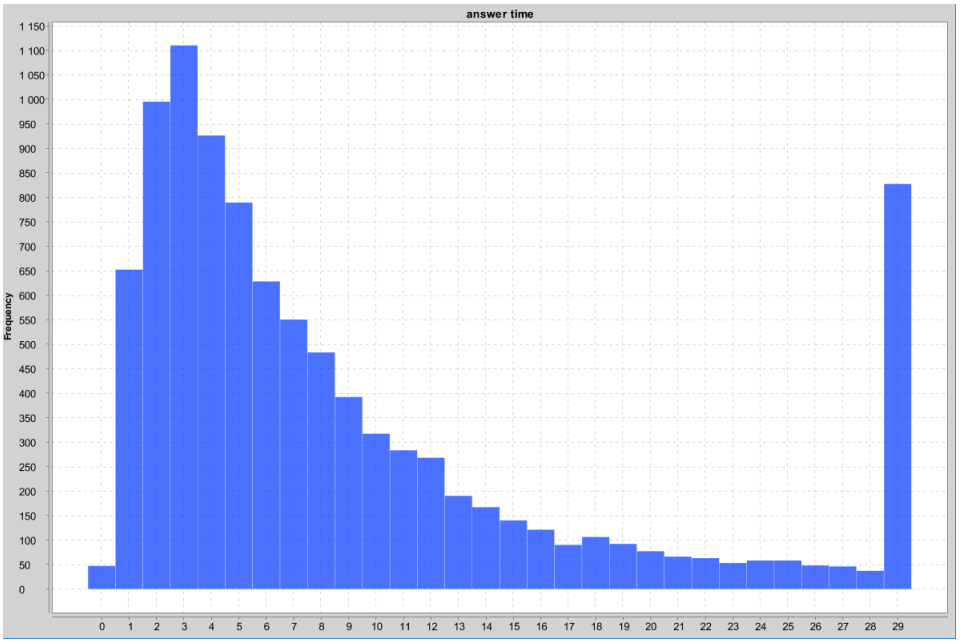

Рисунок 6 – Время ответа на задание (в секундах). Для выбросов, портящих нормировку признака, было принято значение 30.

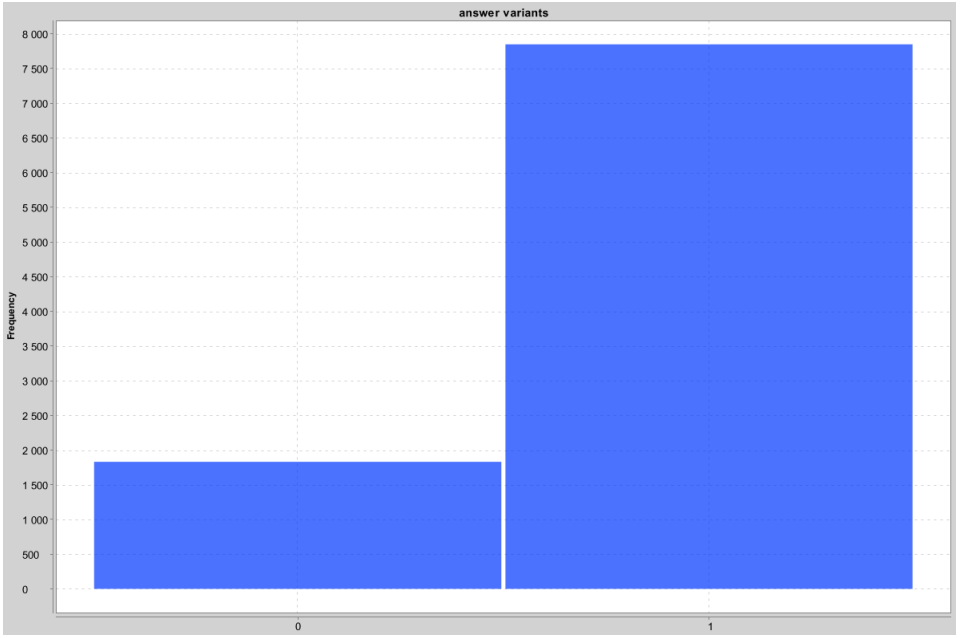

Рисунок 7 – Число разных выбранных вариантов ответа на задание

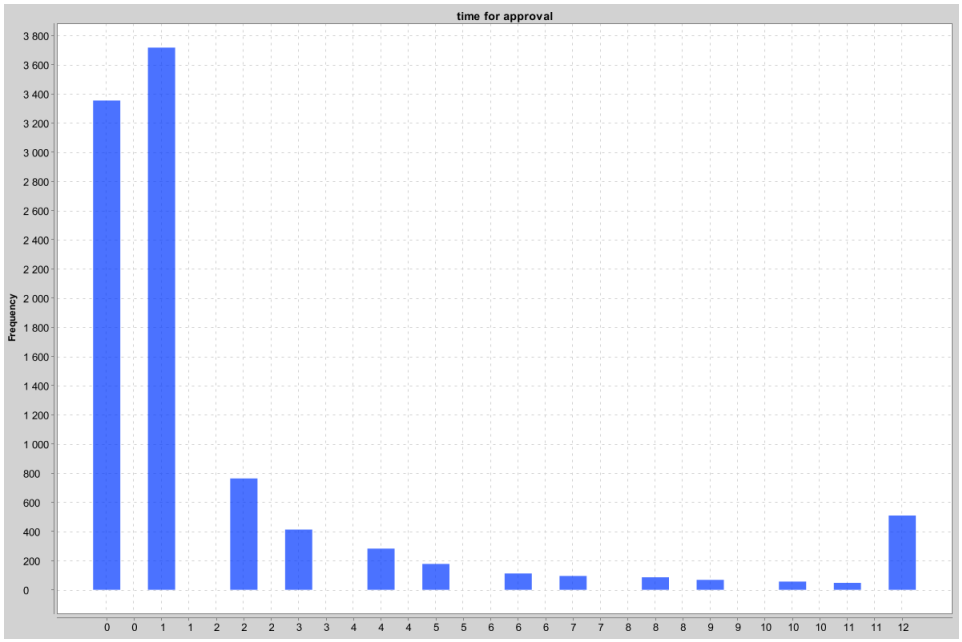

Рисунок 8 – Время между подтверждением ответа и переходом к следующему заданию (в секундах). Для выбросов, портящих нормировку признака, было принято значение 12.

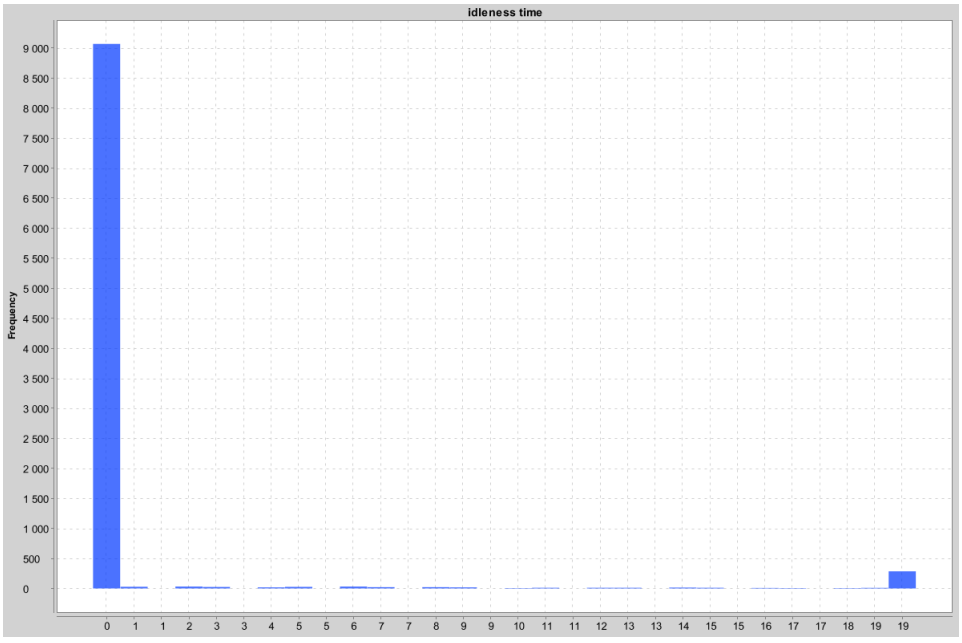

Рисунок 9 – Время бездействия (в секундах). Для выбросов, портящих нормировку признака, было принято значение 20.

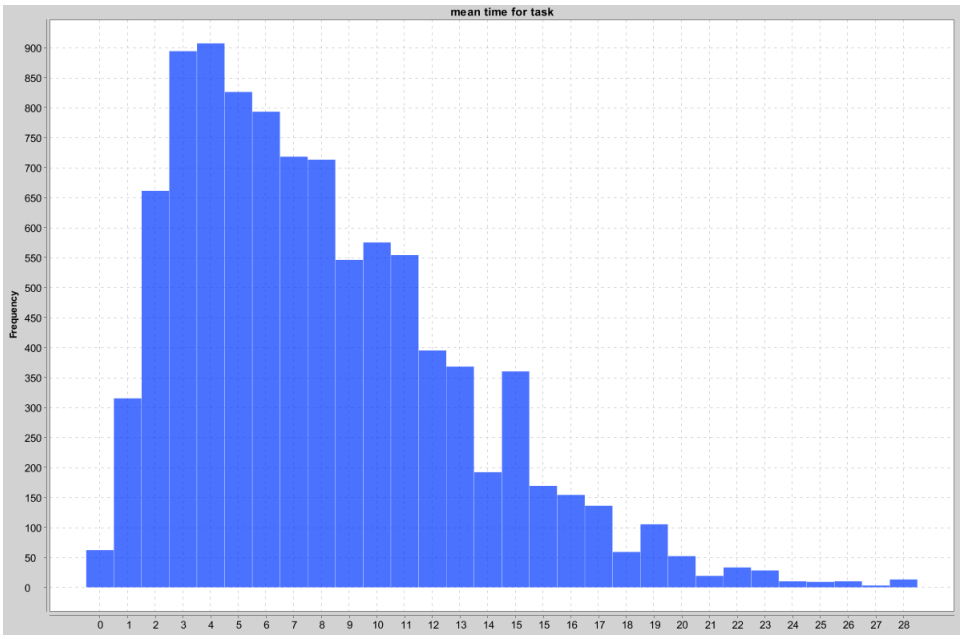

Рисунок 10 – Среднее время ответа для задания (в секундах)

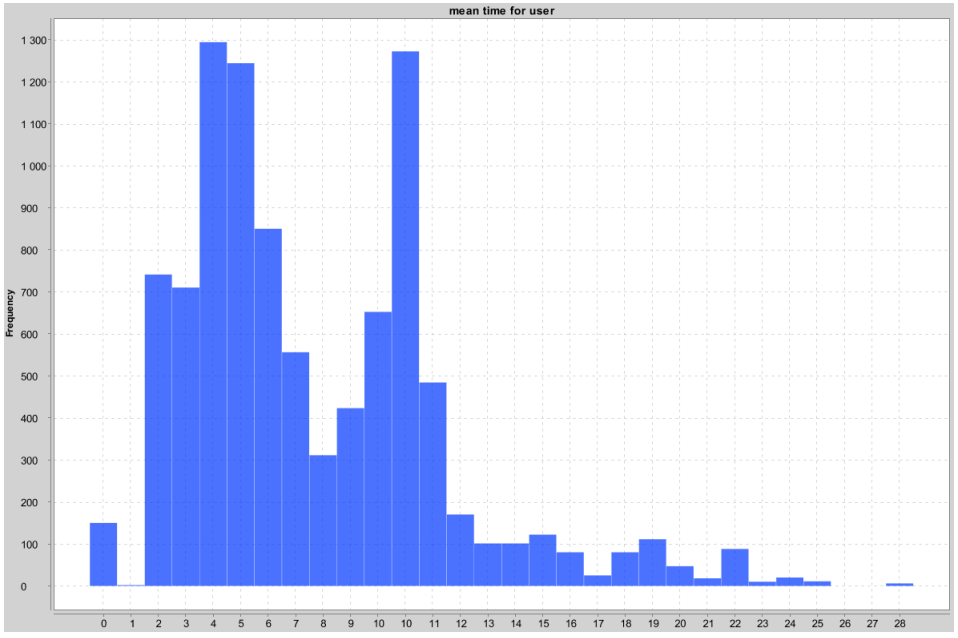

Рисунок 11 – Среднее время ответа пользователя (в секундах)

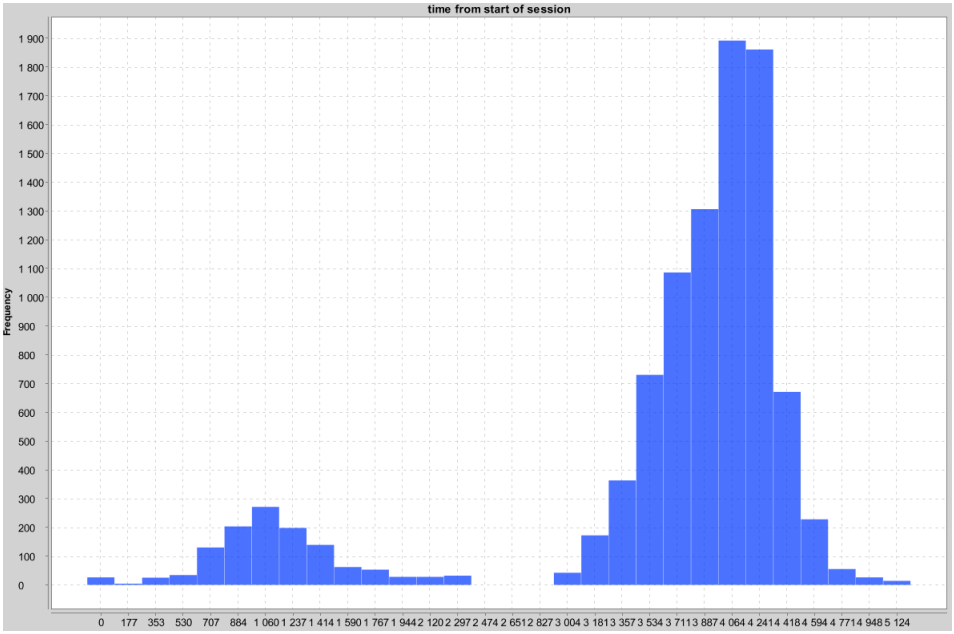

Рисунок 12 – Время с начала сессии (в секундах)

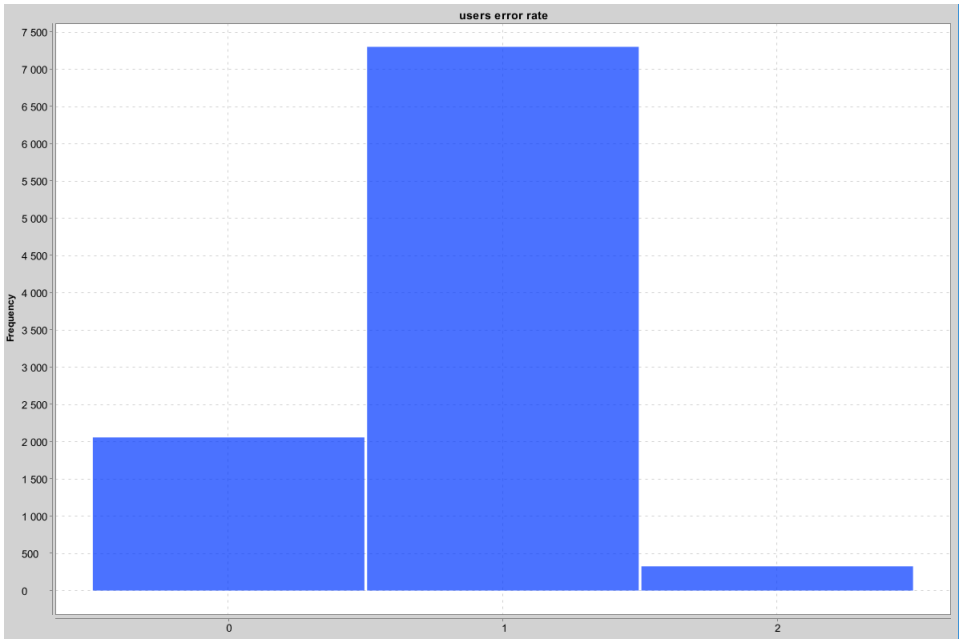

Рисунок 13 – Частота ошибок пользователей. Первому столбцу со значением около 0 соответствуют компетентные исполнители, второму – менее компетентные, допускающие больше ошибок, третьему – спамеры.

все ответы на примерно 217 заданий, для которых все ответы были верны. Обучающее множество было несбалансированным из-за преобладания верных ответов пользователей (в наборе данных было всего 535 ложных ответов против 9144 верных), модели было выгоднее в большинстве случаев присваивать объектам метку 1. Для решения проблемы была применена повышающая дискретизация (upsampling): в обучающее множество было равновероятно добавлено столько же объектов редкого класса, сколько было объектов класса , встречающегося часто.

Итоговая статистика по всем ответам пользователей строилась как объединение результатов для десяти тестовых множеств, полученных при кросс-валидации.

3.4. Оценка качества обученной модели

Матрица ошибок классификации представлена в таблице 2.

Таблица 2 – Матрица ошибок классификации

–	$y = 1$	$y = 0$
$\hat{y} = 1$	8954 (TP)	149 (FP)
$\hat{y} = 0$	95 (FN)	374 (TN)

По значениям из матрицы ошибок классификации были посчитаны точность (precision), полнота (recall) и F_1-мера для каждого класса и были получены их средние значения, результаты представлены в таблице 3. Значения для класса 0 (неверный ответ) ощутимо хуже, чем для класса 1, что связано с несбалансированностью набора данных.

Таблица 3 – Точность, полнота, F1-мера

Класс	Точность	Полнота	F1-мера
1 (верный ответ)	0.98	0.96	0.97
0 (неверный ответ)	0.8	0.71	0.75
Среднее значение	0.89	0.84	0.86

Значение бинарной логистической функции потерь составило 0.07, что является хорошим показателем.

Квадратный корень из среднеквадратического отклонения между усреднённой оценкой пользователей по всем их ответам и доли правильных ответов пользователей составил 0.07. Данное значение улучшается с ростом числа ответов пользователя, например, если взять пользователей, у которых было больше 100 ответов, и посчитать квадратный корень из среднеквадратического отклонения только для них получается 0.001. Данные значения говорят о том, что хотя классификация отдельных неверных ответов пользователей не очень хороша, судя по значению F_1 меры, средняя оценка пользователей по всем ответам даёт очень хорошие результаты.

3.5. Результаты

3.5.1. Модификация метода Дэвида-Скина

Агрегированные ответы совпали с истинными, так как вопросы были простыми, поэтому для оценки результатов использовалась вторая метрика из представленных в главе 2.

Таблица 4 – Результаты с модификацией алгоритма Дэвида-Скина

–	Качество
Алгоритм Дэвида-Скина с оценкой исполнителей	0.996
Алгоритм Дэвида-Скина с оценкой ответов	0.995
Алгоритм Дэвида-Скина без модификаций	0.976

Оценки качества агрегации на данном наборе данных для алгоритма Дэвида-Скина и его модификаций представлены в таблице 4. Из них следует, что удалось улучшить алгоритм Дэвида-Скина на 2%.

Исследованный набор данных содержал в основном верные ответы, поэтому немодифицированный алгоритм Дэвида-Скина выдал на нём очень высокое качество агрегированных ответов, вероятнее всего, удалось бы больше улучшить качество агрегации на наборе данных с большей долей неправильных ответов пользователей.

3.5.2. Модификации голосования большинства

Результаты для метода голосования большинства и его модификаций представлены в таблице 5. Удалось улучшить результаты по сравнению с простым голосованием большинства на 1.7%.

Таблица 5 – Результаты с модификациями голосования большинства

–	Качество
Голос большинства, взвешенный поведенческими признаками и частотой ошибок из алгоритма Дэвида-Скина	0.956
Голос большинства, взвешенный поведенческими признаками	0.955
Голос большинства	0.939

3.5.3. Итоговые результаты

Итоговые результаты представлены в таблице 6.

Таблица 6 – Итоговые результаты

–	Качество
Алгоритм Дэвида-Скина с оценкой исполнителей	0.996
Алгоритм Дэвида-Скина с оценкой ответов	0.995
Алгоритм Дэвида-Скина без модификаций	0.976
Голос большинства, взвешенный поведенческими признаками и частотой ошибок из алгоритма Дэвида-Скина	0.956
Голос большинства, взвешенный поведенческими признаками	0.955
Голос большинства	0.939

3.6. Важность признаков

На рисунке 14 представлен вклад признаков в ответ. Важность признаков оценена с помощью случайного леса. Лучшими признаками оказались (по убыванию значимости):

а) среднее время ответа пользователя;
б) среднее время ответа на задание;
в) время с начала сессии;
г) время ответа;
д) число смен ответа.

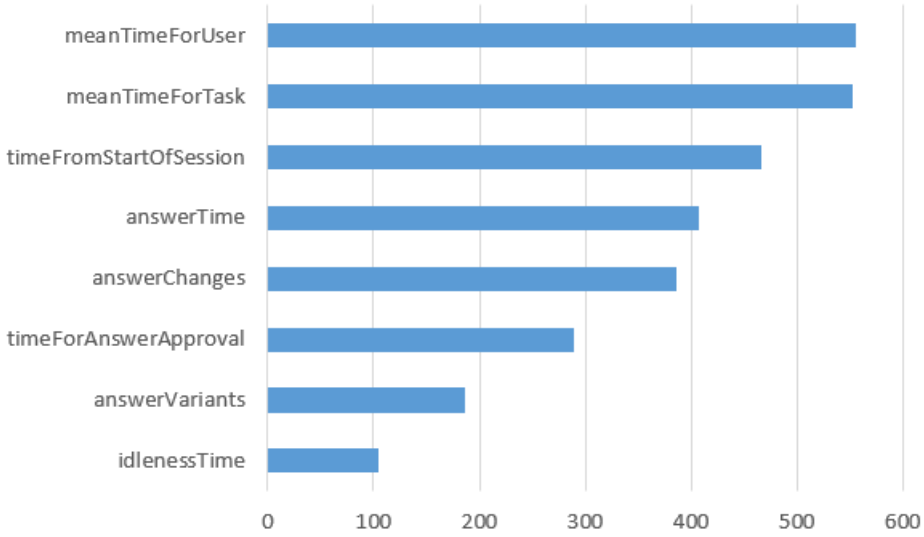

Рисунок 14 – Вклад признаков в ответ. MeanTimeForUser – среднее время ответа на вопрос для пользователя, meanTimeForTask – среднее время ответа на вопрос для задания, answerChanges – число смен ответа, timeForAnswerApproval – время между ответом на вопрос и его подтверждением, answerVariants – число разных вариантов ответа на вопрос, на которые пользователь наводил курсор или выбирал их, idlenessTime – время бездействия пользователя

3.7. Использованные библиотеки и программное обеспечение

Для проведения исследования и получения результатов использовалась среда разработки Intellij IDEA, язык программирования Java, система сборки Gradle. Для парсинга и записи .tsv и .csv-файлов использовалась библиотека univocity-parsers, случайный лес был взят из библиотеки smile regression, графики строились с помощью библиотеки xchart, точность, полнота, F_1-мера и матрица ошибок классификации рассчитывались с помощью библиотеки org.apache.commons.

Выводы по главе 3

В главе 3 были описаны исходные данные для эксперимента и способ их обработки. Были представлены результаты сравнения модификации алгоритма Дэвида-Скина с его же версий без модификации, удалось улучшить качество агрегации ответов. Был составлен список наиболее удачно подобранных признаков, внёсших наибольший вклад в результат. Были описаны модификации метода голосования большинства, дающие лучший результат, чем простой метод голосования большинства. Была приведена оценка качества обученной

модели машинного обучения. Были описаны использованные библиотеки и программное обеспечение.

ЗАКЛЮЧЕНИЕ

В работе предложен способ модификации широко использующегося при агрегации ответов в краудсорсинге и хорошо себя зарекомендовавшего алгоритма Дэвида-Скина. Предложенный метод учитывает поведение пользователей, что позволяет бороться с такими проблемами как падение концентрации пользователя с течением времени, сомнения в ответе.

Полученный метод можно внедрять в краудсорсинговые платформы и получать более качественные результаты.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 A crowdsourcing worker quality evaluation algorithm on MapReduce for big data applications / D. Dang [et al.] // IEEE Transactions on Parallel and Distributed Systems. — 2015. — Vol. 27, no. 7. — P. 1879–1888.

2 An evaluation of aggregation techniques in crowdsourcing / N. Q. V. Hung [et al.] // International Conference on Web Information Systems Engineering. — Springer. 2013. — P. 1–15.

3 *Burmania A.*, *Parthasarathy S.*, *Busso C.* Increasing the reliability of crowdsourcing evaluations using online quality assessment // IEEE Transactions on Affective Computing. — 2015. — Vol. 7, no. 4. — P. 374–388.

4 Consensus algorithms for biased labeling in crowdsourcing / J. Zhang [et al.] // Information Sciences. — 2017. — Vol. 382. — P. 254–273.

5 *Dawid A. P.*, *Skene A. M.* Maximum likelihood estimation of observer error-rates using the EM algorithm // Journal of the Royal Statistical Society: Series C (Applied Statistics). — 1979. — Vol. 28, no. 1. — P. 20–28.

6 *Demartini G.*, *Difallah D. E.*, *Cudré-Mauroux P.* ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking // Proceedings of the 21st international conference on World Wide Web. — ACM. 2012. — P. 469–478.

7 *Karger D. R.*, *Oh S.*, *Shah D.* Iterative learning for reliable crowdsourcing systems // Advances in neural information processing systems. — 2011. — P. 1953–1961.

8 *Khattak F. K.*, *Salleb-Aouissi A.* Quality control of crowd labeling through expert evaluation // Proceedings of the NIPS 2nd Workshop on Computational Social Science and the Wisdom of Crowds. Vol. 2. — 2011. — P. 5.

9 Learning from crowds / V. C. Raykar [et al.] // Journal of Machine Learning Research. — 2010. — Vol. 11, Apr. — P. 1297–1322.

10 *Lee J.*, *Lee D.*, *Hwang S.-w.* CrowdK: Answering top-k queries with crowdsourcing // Information Sciences. — 2017. — Vol. 399. — P. 98–120.

11 *Lee K.*, *Caverlee J.*, *Webb S.* The social honeypot project: protecting online communities from spammers // Proceedings of the 19th international conference on World wide web. — ACM. 2010. — P. 1139–1140.

12 Limits on the majority vote accuracy in classifier fusion / L. I. Kuncheva [et al.] // Pattern Analysis & Applications. — 2003. — Vol. 6, no. 1. — P. 22–31.

13 *Mason W.*, *Watts D. J.* Financial incentives and the performance of crowds // Proceedings of the ACM SIGKDD workshop on human computation. — ACM. 2009. — P. 77–85.

14 Modeling annotator behaviors for crowd labeling / Y. E. Kara [et al.] // Neurocomputing. — 2015. — Vol. 160. — P. 141–156.

15 *Moon T. K.* The expectation-maximization algorithm // IEEE Signal processing magazine. — 1996. — Vol. 13, no. 6. — P. 47–60.

16 *Morris R. R.*, *Dontcheva M.*, *Gerber E. M.* Priming for better performance in microtask crowdsourcing environments // IEEE Internet Computing. — 2012. — Vol. 16, no. 5. — P. 13–19.

17 *Nicholson B.*, *Sheng V. S.*, *Zhang J.* Label noise correction and application in crowdsourcing // Expert Systems with Applications. — 2016. — Vol. 66. — P. 149–162.

18 Noise filtering to improve data and model quality for crowdsourcing / C. Li [et al.] // Knowledge-Based Systems. — 2016. — Vol. 107. — P. 96–103.

19 Quality control in crowdsourcing systems: Issues and directions / M. Allahbakhsh [et al.] // IEEE Internet Computing. — 2013. — Vol. 17, no. 2. — P. 76–81.

20 *Sheng V. S.*, *Provost F.*, *Ipeirotis P. G.* Get another label? improving data quality and data mining using multiple, noisy labelers // Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. — ACM. 2008. — P. 614–622.

21 Supervised learning from multiple experts: whom to trust when everyone lies a bit / V. C. Raykar [et al.] // Proceedings of the 26th Annual international conference on machine learning. — ACM. 2009. — P. 889–896.

22 Use of ontology structure and Bayesian models to aid the crowdsourcing of ICD-11 sanctioning rules / Y. Lou [et al.] // Journal of biomedical informatics. — 2017. — Vol. 68. — P. 20–34.

23 Whose vote should count more: Optimal integration of labels from labelers of unknown expertise / J. Whitehill [et al.] // Advances in neural information processing systems. — 2009. — P. 2035–2043.