

Санкт-Петербургский Государственный Университет
Математико-механический факультет

Программная инженерия
Системное программирование

Молчанов Артём Андреевич

Алгоритм выявления нечестного поведения в онлайн-курсах

Выпускная квалификационная работа

Научный руководитель:
д. ф.-м. н., профессор Терехов А. Н.

Рецензент:
старший аналитик данных ООО «Стэпик» Скрыбин М. А.

Санкт-Петербург
2017

SAINT-PETERSBURG STATE UNIVERSITY

Software Engineering
Department of Software Engineering

Molchanov Artyom

Cheating Detection Algorithm in Online Courses

Graduation Project

Scientific supervisor:
professor Andrey Terekhov

Reviewer:
senior data analyst Stepik LLC Maxim Skryabin

Saint-Petersburg
2017

Оглавление

Введение	4
1. Постановка задачи	7
2. Обзор	8
2.1. Существующие подходы и решения	8
2.2. Описание объемлющего проекта	11
2.3. Используемые технологии	12
2.4. Специфика задачи	13
3. Признаки нечестного поведения	14
3.1. Определение признаков на основе поведенческой психологии	14
3.2. Определение признаков на основе анализа существующих решений	15
3.3. Определение окончательного списка признаков	16
4. Алгоритм	21
4.1. Разработка первичного алгоритма	21
4.2. Внедрение в алгоритм условий различного поведения пользователя.	22
5. Оценка алгоритма	24
5.1. Тестирование	24
5.2. Корректность	27
6. Подсистема генерации отчётов	29
Заключение	30
Список литературы	31

Введение

Ценность информации сегодня трудно переоценить. Данные являются ключевой формой ее представления. На основе их интерпретации и статистического анализа можно делать различные выводы и принимать решения.

К сожалению, зачастую выявить выявлять закономерности больших объёмов данных «вручную» весьма затруднительно или вовсе невозможно, тогда применяют методы машинного обучения. Они позволяют строить алгоритмы, которые помогают находить новые, более сложные, не описанные ранее закономерности.

Также особое место в мире исследований и анализа занимает психология. Современная психология представляет собой многоотраслевую науку. Одним из начальных направлений развития психологии двадцатого века является поведенческая психология (бихевиоризм), которая опирается на изучение причин трудностей человека и всевозможных проблем, которые связаны с тем, что человек в неблагоприятных обстоятельствах принял для себя неправильную или даже неадекватную форму поведения.

Образование является фундаментом эволюции общества. Его развитие является одной из самых важных задач сегодня во всём мире. Одна из его форм – дистанционное обучение, при котором взаимодействие учителя и учащихся между собой происходит на расстоянии. Оно содержит все присущие учебному процессу компоненты (цели, содержание, методы, организационные формы, средства обучения) и реализуется специфичными средствами Интернет-технологий или другими средствами, предусматривающими интерактивность.

Благодаря этому, такая форма обучения становится удобной и доступной, а также занимает всё большее место в модернизации образования в целом. Однако возникают различные проблемы, такие как:

- Отсутствие очного общения между обучающимися и

преподавателем.

- Необходимость жесткой самодисциплины, так как результат обучения напрямую зависит от самостоятельности и сознательности учащегося.
- Отсутствие постоянного контроля над обучающимися, который зачастую является мощным побудительным стимулом.

Из-за сложности в соблюдении правил дистанционного обучения и самоконтроля при прохождении онлайн-курсов у пользователей появляется желание (порой произвольное) выполнить задания не совсем честным способом, например:

- Перебирать ответы, совершая много попыток в короткий промежуток времени.
- Быстро решать задания с одной попытки, найдя готовые ответы в Интернет-сети или имея несколько аккаунтов.

Актуальность выявления нечестного поведения пользователей в онлайн-курсах состоит в следующем:

- Возрастает ценность сертификатов дополнительного образования.
- Усиливается конкуренция со стандартным образованием.
- Компании заинтересованы в прочных знаниях сотрудников, проводя корпоративное обучение.
- Оценка реального уровня знаний важна для платформ онлайн-обучения при выявлении неправильно составленных заданий или же при построении рекомендательных систем.
- Выявление странных заданий способствует повышению качества учебных материалов и программ.
- Нечестное поведение является возможным проявлением потери желания обучаться, а также способствует падению качества знаний.

Учитывая вышеизложенные причины, необходимо контролировать подобные моменты, для повышения реального качества знаний пользователей-учеников. В свою очередь, так как подобное поведение трудно и долго идентифицировать «вручную», необходимо, опираясь на исследования в области поведенческой психологии и существующие решения определить паттерны нечестного поведения, и, на основе анализа имеющихся данных с применением методов машинного обучения разработать алгоритм выявления нечестного поведения в онлайн-курсах. Это и является задачей, которая решается в данной работе.

1. Постановка задачи

Целью данной дипломной работы является разработка алгоритма выявления нечестного поведения в онлайн-курсах и реализация подсистемы генерации отчётов по подобному поведению на платформе Stepik.org [32]. Для достижения этой цели в рамках работы были сформулированы следующие задачи:

- На основе принципов поведенческой психологии и анализа существующих решений определить признаки нечестного поведения в онлайн-образовании.
- Разработать алгоритм выявления нечестного поведения в онлайн-курсах.
- Протестировать алгоритм и оценить его корректность на реальных данных различных курсов с различными типами задач.
- Реализовать подсистему генерации отчётов по нечестному поведению пользователей на платформе Stepik.org.

2. Обзор

2.1. Существующие подходы и решения

Понимание предметной области обеспечивает обзор [22], [33], [4]. В них нечестное поведение характеризуется как:

- Частые и неоднократные запросы помощи до тех пор, пока система сама не даст очевидную подсказку или вовсе правильный ответ.
- Частым и неоднократным вводом ответов (например, ввод ответов по порядку - 1, 2, 3, и т.д.; или совершение кликов на все подряд варианты в заданиях множественного выбора).

Также систематически наблюдая за поведением учеников, автор (на основе [13], [14], [15], [17]) разделяет типы поведения:

- Работа в системе.
- Разговор с учителем или другим учеником по поводу задания.
- Разговор не о предмете.
- Любое поведение, не связанное с программным обеспечением для обучения или другим лицом (например, чтение новостей или просмотр веб-страниц).
- Отсутствие любой активности (например, ученик спит).
- Нечестное поведение.

В [22] определяются «вредное» (ухудшающее результат обучения) и «невредное» (не влияющее на результат) нечестное поведение. Важным результатом анализа данной диссертации является определение признаков нечестного поведения, например:

- Длительность попытки.

- Отклонение времени попытки ученика от среднего времени попытки в текущем задании среди всех учеников, выраженное в стандартных отклонениях.
- Время последних 3-х и 5-и попыток, выраженное в виде суммы стандартных отклонений от средних временных промежутков попыток среди всех пользователей.
- Общее число ошибок в заданиях на этом же этапе.
- Доля предыдущих заданий, в которых ошибки были совершены на этом же этапе.
- Сколько из последних 5 действий было связано с текущим заданием.
- Количество ошибочных попыток их последних 5-и.

Важными достоинствами являются:

- Сформированы интересные признаки нечестного поведения.
- Один из источников данных - логи (данные о действиях пользователей) программного обеспечения для обучения (аналог платформы онлайн-обучения).

Основные минусы состоят в том, что:

- «Вред» нечестного поведения оценивается при помощи входного и выходного тестирования (на большинстве платформ онлайн-образования отсутствует либо подобный функционал).
- Определяется доля нечестного поведения пользователя во всём уроке (курсе), а не для конкретного задания.
- Оценка частоты проявления нечестного поведения при выполнении заданий происходит на основе наблюдения за учениками. (в онлайн-платформах такой подход трудно реализуем).

В качестве модели используется Latent Response Models (LRM) [19], для обучения которой применяется комбинация Forward Selection (Ramsey and Schafer 1997) и Iterative Gradient Descent (Boyd and Vandenberghe 2004), а если наборы данных становятся больше, вводится фильтрация с быстрой корреляцией [36].

В результате тестирования модели автор получил следующие результаты:

- При тестировании на 2-х разных годах обучения происходит ухудшение качества при применении модели к данным другого года, однако результат всё же значительно отличается от 0.5 (р-значение < 0.05), что означает относительную успешность применения.
- При тестировании на 4-х различных уроках получается удовлетворительный результат, только при тренировке на 3-х уроках, то есть для улучшения результатов нужно либо менять подход, либо увеличивать размер и вариативность данных.

На основе обзора [34] из признаков нечестного поведения можно добавить к уже обнаруженным:

- Время на подсказку после открытия задания.
- Среднюю сложность задачи.
- Количество перезапусков системы, для сброса результатов.

Также признаки сгруппированы по категориям, что даёт понимание того, как сильно различные типы поведения влияют на результат.

Несколько применяемых алгоритмов машинного обучения WEKA (Waikato Environment for Knowledge Analysis) [11], логистическая регрессия (logistic regression) [18] и решающее дерево (decision-tree) [26] показали схожие результаты, исходя из чего, можно сделать вывод, что выбор модели не так сильно влияет на качество предсказания, как качественные данные и верно определённые и отобранные признаки.

На основе обзора литературы по данной теме, можно выделить общие положительные моменты:

- Исследованы поведенческие факторы, определяющие нечестное поведение.
- Точные специфические модели Архитектура системы определения нечестного решения.

Однако присутствует ряд недостатков, которые затрудняют прямое применение и исследование подходов, описанных в обзоре работ:

- Определяется общая доля нечестного поведения на курсе (уроке).
- Анализируются однотипно составленные задания (в рамках 2-х школ по одинаковым предметам и схожим методикам).
- Обучение классификатора производится только на предварительно размеченных данных.

2.2. Описание объемлющего проекта

Stepik.org («Стэпик») – это образовательная платформа и конструктор бесплатных открытых онлайн-курсов и уроков.

Stepik позволяет любому зарегистрированному пользователю создавать интерактивные обучающие уроки и онлайн-курсы, используя тексты, видео и разнообразные задачи с автоматической проверкой и моментальной обратной связью.

Структуру онлайн-курсов можно описать следующим образом: курсы состоят из модулей, модули из уроков, а уроки из шагов (заданий) разного типа. Это позволяет структурировать получаемые знания и наилучшим образом следить за успеваемостью.

Проект сотрудничает как с образовательными учреждениями, так и с индивидуальными преподавателями и авторами.

На настоящий момент на платформе доступны курсы по различным предметам: программированию, математике, экономике,

психологии, биоинформатике, биологии, журналистике и культуре, на русском и английском языках.

2.3. Используемые технологии

Для разработки алгоритма был выбран язык программирования Python версии 3.4 по следующим причинам:

- Возможность быстрой прототипизации.
- Качественная документация в должном объёме.
- Наличие библиотек с большим числом примеров использования.
- На данном языке реализована серверная часть платформы.

Использовались такие библиотеки как:

- Pandas [24] версии 0.15 для обработки и подготовки данных, вычисления признаков и составления итогового отчёта.
- Scikit-learn [29] версии 1.7.0 для построения и тестирования моделей машинного обучения.
- NumPy [37] версии 1.12.1 для проведения эффективных операций над векторами и матрицами.
- SciPy [30] версии 0.19.0 для отбора признаков, проверки статистических гипотез и оценки полученных результатов.

А также в качестве framework-а («фреймворка») для реализации подсистемы генерации отчётов выбран Django [9] версии 1.8, так как он используется для реализации серверной части платформы Stepik.org.

Разработка велась в PyCharm [25] – интегрированной среде разработки для языка программирования Python, преимущество которой заключаются в следующем:

- Позволяет наглядно работать с системой контроля версий Git.

- Обладает удобным интерфейсом и автодополнением кода, что повышает скорость и качество разработки.
- Предоставляет возможность работы с базами данных.
- Реализована интеграция с Django framework («фреймворк»).
- Предлагает хорошие возможности для отладки приложения.

2.4. Специфика задачи

Задача алгоритмического определения нечестного поведения является специфической и трудно реализуемой:

- Нет размеченной выборки, то есть неизвестно конкретно, является ли поведение пользователя нечестным при выполнении задания
- К различным заданиям и различным курсам плохо применим алгоритм, разработанный на основе данных лишь нескольких курсов и типов заданий
- Некоторые поведенческие признаки невозможно определить на платформе реализации подсистемы, так как отсутствуют соответствующая функциональность либо данные.
- В онлайн-курсах на платформе Stepik.org в большинстве случаев не проводится входное и выходное тестирование.
- Затруднительно применять стандартные метрики качества для моделей машинного обучения.

3. Признаки нечестного поведения

Хорошие данные - важнейшая часть решения, поскольку из них получаются признаки для построения алгоритма и применения моделей машинного обучения. Исходя из проблематики, необходимо определить общие паттерны поведения.

3.1. Определение признаков на основе поведенческой психологии

После обзора предметной области, можно выделить следующие паттерны поведения пользователя, ведущего себя нечестно:

- Пользователь совершил хотя бы одну ошибку и попыток уже много.
- Пользователь совершил хотя бы одну ошибку и попытки совершаются через короткие промежутки времени.
- Пользователь осуществил много попыток решить задание и тип задания - множественный выбор.

Исходя из анализа полученных паттернов созданы следующие признаки:

- Признаки, основанные на времени:
 - Суммарное время, затраченное на решение задания.
 - Среднее время среди попыток решить задание.
- Признаки, основанные на количестве попыток:
 - Сколько попыток совершил пользователь в задании.
 - Совершена ли хотя бы одна попытка в задании.
 - Средняя доля ошибочных попыток в заданиях пользователя.
- Признаки, основанные на количестве подсказок:

- Сколько раз пользователь уже запрашивал подсказку.
- Среднее время между началом задания и запросом на подсказку на всех предыдущих заданиях.

3.2. Определение признаков на основе анализа существующих решений

Проанализировав существующие решения и зная уже определённые паттерны нечестного поведения, можно усложнить имеющиеся, а также добавить новые, такие как:

- Пользователь совершил хотя бы одну ошибку и количество попыток в текущем задании $>$ (больше) 5.
- Пользователь совершил хотя бы одну ошибку и среднее время среди попыток решить задание пользователем $<$ (меньше) среднего по времени попыток в текущем задании среди всех пользователей.
- Количество попыток в текущем задании $>$ 3 и медианы средних промежутков времени попыток в текущем задании среди всех пользователей.
- Пользователь не совершает ошибок уже 5 заданий подряд и попытки совершаются через короткие ($<$ медиана - медианное абсолютное отклонение) промежутки времени.
- Пользователь часто (количество подсказок $>$ среднего по всем пользователям + стандартное отклонение) делает запросы на подсказку, даже в заданиях с высоким ($>$ среднее + стандартное отклонение) процентом удачных попыток.

Также можно уточнить пространство признаков и добавить новые, более сложные, в том числе учитывающие предыдущие задания, например:

- Признаки, основанные на времени:
 - Среднее время среди попыток решить задание $<$ медианы средних временных промежутков на попытку решения задания среди всех пользователей.
 - Суммарное время, затраченное на решение последних 3-х и 5-и заданий $<$ медианы суммарных временных промежутков на решение последних 3-х и 5-и заданий среди всех пользователей - медианное абсолютное отклонение.
- Признаки, основанные на количестве попыток:
 - Количество попыток пользователя решить задание $>$ среднего количества попыток решить задание среди всех пользователей.
 - Количество попыток в последних 3-х и 5-и заданиях $>$ среднего по суммарному количеству попыток решить последние 3-х и 5-и заданий среди всех пользователей.
- Признаки, основанные на количестве подсказок:
 - Среднее время между началом задания и запросом на подсказку на всех предыдущих заданиях $<$ среднего по средним временным промежуткам между началами заданий и запросами на подсказку на всех предыдущих заданиях среди всех пользователей - стандартное отклонение.
 - Количество запросов пользователя на подсказку в последних 3-х и 5-и заданиях $>$ среднего по суммарному количеству запросов на подсказки последние 3-х и 5-и заданий среди всех пользователей.

3.3. Определение окончательного списка признаков

Для создания признаков нечестного поведения, изучена структура курсов на платформе Stepik.org, а также проанализированы имеющиеся

данные. О действиях пользователей известна следующая информация:

- Время начала попытки.
- Время отправки решения.
- Удачная или нет была попытка.
- Уникальный идентификатор пользователя.
- Уникальный идентификатор шага (задания).

Для получения базовых признаков данные были сгруппированы по паре (идентификатор пользователя, идентификатор задания). Для каждой группы рассчитаны основные признаки, такие как:

- Количество совершённых пользователем попыток в задании.
- Решено ли задание пользователем.
- Среднее время попытки в задании.

На основе полученных базовых признаков было создано 46 различных признаков. Такое количество признаков достаточно велико как и вероятность того, что среди них есть неинформативные. Присутствие неинформативных признаков в имеющихся данных может привести к снижению точности большинства моделей, особенно линейных, таких как линейная [16] и логистическая регрессия. Поэтому возникла необходимость сократить размерность признакового пространства, при этом произвести отбор признаков – процесс выбора признаков, имеющих наиболее сильные взаимосвязи с целевой переменной.

Отбор признаков и определение наиболее важных обеспечивает следующие преимущества:

- Уменьшение переобучения (уменьшение избыточных данных, также уменьшает возможности для модели принимать решения на основе «шумовых» данных).

- Повышение точности (при уменьшении противоречивых данных повышается точность).
- Сокращение времени обучения (уменьшение количества данных способствует более быстрому обучению модели).

В данной работе отбор признаков производился в три параллельно выполняющихся этапа:

- После предварительной нормализации данных применяется принцип главных компонент (PCA (principal component analysis) [23]).
- После передискретизации (подробнее в следующем разделе), к размеченной¹ выборке применяется дисперсионный анализ (ANOVA) [1].
- Также после передискретизации, на размеченной выборке обучается модель машинного обучения с применением метода рекурсивного исключения признаков (recursive feature elimination, RFE) [27].

В итоге, для каждой уникальной пары (идентификатор шага, идентификатор пользователя) был определён ряд наиболее сильно взаимосвязанных с целевой переменной признаков:

1. В текущем задании:

1.1 По количеству попыток:

1.1.1 Есть ли хоть одна ошибка в задании.

1.1.2 Количество попыток пользователя решить задание $>$ (медианы количества попыток решить задание среди всех пользователей + медианное абсолютное отклонение).

¹Чтобы выборка была размечена, необходимо наличие целевой переменной, однако первоначально метки классов отсутствуют. Эта проблема была частично решена с помощью имитационных и размеченных «вручную» данных (подробнее в следующем разделе).

1.1.3 Количество попыток пользователя решить задание $>$ среднего количества попыток решить задание среди всех пользователей.

1.1.4 Количество попыток пользователя решить задание $>$ (среднего количества попыток решить задание среди всех пользователей + стандартное отклонение).

1.2 По времени решения задания или попыток решить задание:

1.2.1 Среднее время попытки пользователя на шаге $<$ (медианы по средним временным промежуткам на попытки решить задание среди всех пользователей - медианное абсолютное отклонение).

1.2.2 Время решения задания пользователем $<$ (медианы временных промежутков на решение задания среди всех пользователей - медианное абсолютное отклонение).

1.2.3 Среднее время попытки решить задание пользователем $<$ (среднего времени попытки решить задание среди всех пользователей - стандартное отклонение).

1.2.4 Медиана времени попыток решить задание пользователем $<$ (среднего медиан времени попыток решить задание среди всех пользователей).

1.3 По трудности задания:

i. Процент успеха на шаге $>$ среднего процента успеха при решении всех заданий.

2. В решённых ранее заданиях:

2.1 Нет ошибок в последних 3-х заданиях.

2.2 Нет ошибок в последних 5-и заданиях.

2.3 Условие (1.1.4) выполнялось последние 2 задания.

2.4 Условие (1.1.4) выполнялось последние 3 задания.

2.5 Условие (1.2.4) выполнялось последние 2 задания.

2.6 Условие (1.2.4) выполнялось последние 3 задания.

4. Алгоритм

4.1. Разработка первичного алгоритма

Стандартный подход при решении задач подобного типа - кластеризация без обучения на размеченных данных. Однако в этом случае оценить качество алгоритма, и тем более его настроить весьма затруднительно, так как нужно понимать, по какому принципу образованы кластеры. Результаты построенных моделей кластеризации были оценены (Таблица 1) с помощью параметрического Т-критерия Стьюдента для независимых выборок с различными дисперсиями. Исходя из относительно небольших по размеру выборок, значение уровня значимости было выбрано равным 0.05.

Таблица 1: Оценка результатов различных алгоритмов кластеризации

Модель	T-значение	p-значение
K-means [12]	0.305 ± 0.050	0.312
Mean Shift [20]	0.271 ± 0.050	0.463
Spectral Clustering [31]	0.847 ± 0.050	0.105
DBSCAN [8]	1.572 ± 0.050	0.051
MiniBatch k-means [21]	0.638 ± 0.050	0.167
Affinity Propagation [5]	0.798 ± 0.050	0.271
Agglomerative Clustering [6]	1.120 ± 0.050	0.089

Хоть и DBSCAN показал достаточно хорошие результаты с точки зрения статистического анализа, нельзя сказать, определяет ли эта модель пользователей ведущих себя нечестно достаточно точно и полно.

Исходя из этого было принято решение разметить выборку и добавить синтетические (искусственно созданные на основе имеющихся) данные:

- Добровольцы из числа пользователей платформы Stepik.org специально вели себя максимально нечестно при выполнении заранее оговоренных заданий в нескольких разных курсах, и затем на основе идентификатора при выгрузке их можно было легко определить.
- Также были проанализированы значения признаков у добровольцев, с помощью чего некоторые данные решений других пользователей также были размечены, а также добавлены новые, на основе алгоритма K-means (к ближайших соседей).

4.2. Внедрение в алгоритм условий различного поведения пользователя.

В процессе добавления синтетических данных на основе анализа поведения добровольцев и имеющихся паттернов поведения и признаков было решено разработать условия честного и нечестного поведения, которые будут определять пользователей с самым обычным поведением, а также с высоким процентом уверенности нечестном поведении. Это необходимо, так как только при таких условиях возможна адаптация к различным типам заданий и различным курсам, поскольку обучающая выборка будет генерироваться автоматически при каждом новом запуске.

Условия честного и нечестного поведения состоят из различных комбинаций, а также логического умножения и сложения признаков, рассмотренных в разделе **3.3**.

На размеченной выборке с помощью разработанных условий обучается логистическая регрессия и затем предсказывает **вероятность** принадлежности каждой уникальной пары (<пользователь, задание>) классу нечестного поведения на всех данных.

Архитектуру предложенного алгоритма можно изобразить в виде

схемы (Рис. 1).

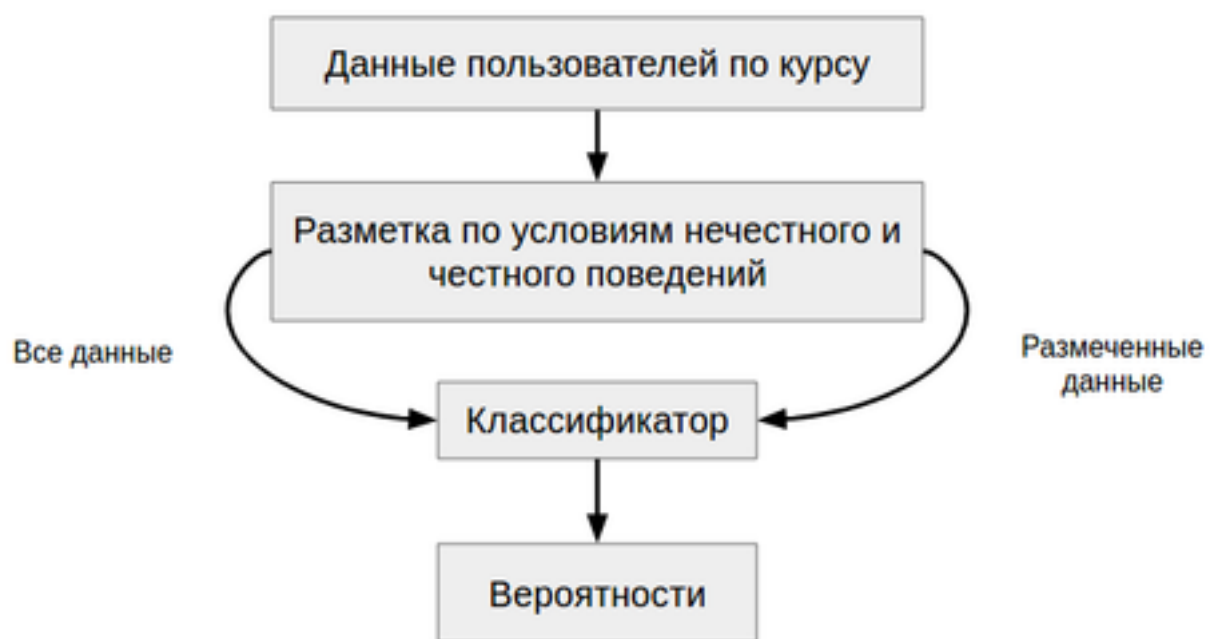


Рис. 1: Схема алгоритма

5. Оценка алгоритма

5.1. Тестирование

Лучший результат кластеризации на размеченных данных при выборках около 2000 пар (<пользователь, задание>) показала (Таблица 2) модель DBSCAN.

Таблица 2: Результаты алгоритма кластеризации DBSCAN на размеченных данных

Доля верных ответов	Точность	Полнота
0.825 ± 0.005	0.310 ± 0.005	0.751 ± 0.005

Для других курсов результаты обученной на одном курсе модели хуже в 0.23 ± 0.02 , обученной на двух курсах хуже в 0.18 ± 0.02 , на 3-х из 4-х хуже в 0.12 ± 0.02 . Исходя из результатов (Таблица 2) можно сделать следующие выводы:

- Большая доля ложно-положительных решений (0.857 ± 0.05).
- Высокая вероятность ошибки первого рода (0.690 ± 0.05).
- Данная модель даже при обучении на нескольких курсах не показывает приемлемых результатов на других курсах, что говорит о её специфичности и неудовлетворительном качестве результатов. Однако, с ростом обучающей выборки за счёт дополнительных курсов качество растёт, тем не менее требуется размеченные выборки из разных курсов и не учитываются возможные особенности заданий, не находящиеся в обучающем множестве.

Тестирование только на имитационных данных проводилось с целью нахождения лучшей модели машинного обучения как по качеству предсказаний, так и по времени работы.

Таблица 3: Сравнение результатов алгоритмов классификации на имитационных данных

Модель	Доля верных ответов	Точность	Полнота	AUC -ROC [2]	Время обучения (сек.)
Наивный байесовский классификатор [10]	0.89	0.86	0.79	0.84	13.25
Логистическая регрессия	0.93	0.92	0.88	0.89	15.67
Лес случайных деревьев [28]	0.89	0.90	0.81	0.85	21.74
XGBoost [35]	0.94	0.91	0.88	0.90	24.38
AdaBoost [3]	0.91	0.87	0.82	0.85	34.65

Модели показали (Таблица 3) схожие результаты по всем метрикам. Однако, лучшими оказались логистическая регрессия и XGBoost. Первая модель проигрывает в доле верных ответов, однако, что более важно, выигрывает в точности, хотя всё же не намного. В общем, результаты этих двух моделей можно считать равными, но быстрее работает первая модель, что является серьёзным преимуществом при работе с большим количеством данных на веб-платформе.

Первичный алгоритм плохо применим к данным других курсов: резко падает точность (в 1.42 раза) и возрастает (на 0.3), как и в случае с кластеризацией, вероятность ошибки первого рода, что является весомым недостатком при решении текущей задачи. То есть можно утверждать, что на реальных данных, без имитационных, результаты сопоставимы с результатами тестирования полностью на

имитационных данных, но только в пределах одного курса.

Таблица 4: Сравнение результатов работы логистической регрессии на данных произвольного курса в зависимости от способа подготовки данных для обучения

Способ подготовки данных для обучения	Доля верных ответов	Точность	Полнота
Имитационные данные одного курса	0.801	0.648	0.721
Размеченные с помощью условий честного и нечестного поведения данные	0.853	0.829	0.736

Качество моделирующей способности логистической регрессии было проверено (Таблица 4) на данных произвольного курса в зависимости от вида подготовки данных для обучения.

Таблица 5: Сравнение результатов алгоритма на данных различных курсов

Курс	Доля верных ответов	Точность	Полнота
1	0.803 ± 0.005	0.810 ± 0.005	0.736 ± 0.005
2	0.881 ± 0.005	0.802 ± 0.005	0.782 ± 0.005
3	0.936 ± 0.005	0.875 ± 0.005	0.766 ± 0.005
4	0.792 ± 0.005	0.798 ± 0.005	0.701 ± 0.005
Среднее	0.853 ± 0.010	0.829 ± 0.010	0.736 ± 0.010

Улучшенный алгоритм был протестирован (Таблица 5) с помощью метода кросс-валидации [7] (при разбиении на 5 частей) на выборках 4-х различных курсов с размеченными выборками. Для того, чтобы

проверить качество работы алгоритма, использовались размеченные выборки для каждого курса.

5.2. Корректность

Основными базовыми признаками в поведении пользователей, на основе которых были созданы описанные в предыдущих разделах сложные признаки, а также определены условия честного и нечестного поведения - количество попыток решить задание и среднего времени между попытками решить задание. Для анализа зависимости нечестного поведения от данных базовых признаков после предварительной нормализации был применён (Таблица 6) параметрический Т-Критерий Стьюдента для независимых выборок с одинаковой дисперсией (с применением критерия рассеяния Фишера для проверки равенства дисперсий двух выборок). Исходя из высокой вероятности ошибки первого рода и достаточно больших по размеру выборок, значение уровня значимости выбрано равным 0.001.

Таблица 6: Результаты применения параметрического Т-критерия Стьюдента для независимых выборок с одинаковой дисперсией для оценки зависимости нечестного поведения от базовых признаков

Признак	T-значение	p-значение
Количество попыток решить задание	7.8 ± 1.5	1.4e-09
Среднее время попыток решить задание	-8.6 ± 1.7	5.3e-08

Анализируя результаты, можно сделать следующие выводы:

- Средние значения количества попыток в двух классах сильно различны, причём количество попыток в нечестных решениях больше, чем в честных.

- Если предположить, что количество попыток в задании и нечестное поведение никак не связаны, то вероятность получить такие или еще более выраженные различия крайне мала (значительно меньше 0.001).
- Средние значения средних временных интервалов попыток в задании в двух классах сильно различны, причём среднее время попыток в задании в нечестных решениях меньше, чем в честных.
- Если предположить, что среднее время попыток в задании и нечестное поведение никак не связаны, то вероятность получить такие или еще более выраженные различия крайне мала (значительно меньше 0.001).

6. Подсистема генерации отчётов

Любой преподаватель на платформе может выгрузить отчёт по нечестному поведению в любом из своих курсов. Процесс генерации отчётов в реализованной подсистеме выглядит следующим образом:

1. Преподаватель выбирает курс и заходит в раздел «Отчёты».
2. Выбирает отчёт о нечестном поведении.
3. Через 1-3 минуты он получает отчёт в двух форматах (CSV и XLSX).

В отчёте содержится наиболее важная информация для понимания уровня нечестного поведения в целом по курсу или в конкретных заданиях, а также определению наиболее нечестных пользователей. Он содержит в себе такие пункты, как:

- Идентификатор пользователя.
- Идентификатор урока.
- Идентификатор задания.
- Порядковый номер задания в уроке.
- Вероятность того, что пользователь выполнил задание нечестно.
- Ссылку на задание.

Заключение

В рамках работы были выполнены следующие задачи.

- На основе принципов поведенческой психологии, анализа существующих решений и имеющихся данных определено 15 признаков нечестного поведения в онлайн-курсах на платформе Stepik.org.
- Разработан алгоритм выявления нечестного поведения в онлайн-курсах.
- Алгоритм протестирован на реальных данных различных курсов с различными типами задач и его корректность оценена с помощью методов статистического анализа.
- Реализована подсистема генерации отчётов по нечестному поведению пользователей на платформе Stepik.org.

Список литературы

- [1] ANOVA. — 2017. — May. — URL: <https://goo.gl/cSR9mh>.
- [2] AUC-ROC. — 2017. — May. — URL: <https://goo.gl/Xf40q0>.
- [3] AdaBoost. — 2017. — May. — URL: <https://goo.gl/SAL7gq>.
- [4] Adapting to when students game an intelligent tutoring system / Ryan SJ d Baker, Albert T Corbett, Kenneth R Koedinger et al. // International Conference on Intelligent Tutoring Systems / Springer. — 2006. — P. 392–401.
- [5] Affinity Propagation. — 2017. — May. — URL: <https://goo.gl/0Rrk1o>.
- [6] Agglomerative Clustering. — 2017. — May. — URL: <https://goo.gl/z23Stv>.
- [7] Cross-validation. — 2017. — May. — URL: <https://goo.gl/0VrrwI>.
- [8] DBSCAN. — 2017. — May. — URL: <https://goo.gl/3pD041>.
- [9] Django. — 2017. — May. — URL: <https://www.djangoproject.com/>.
- [10] Gaussian Naive Bayes. — 2017. — May. — URL: <https://goo.gl/7FyV5D>.
- [11] Holmes Geoffrey, Donkin Andrew, Witten Ian H. Weka: A machine learning workbench // Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on / IEEE. — 1994. — P. 357–361.
- [12] K-means. — 2017. — May. — URL: <https://goo.gl/gzo7Re>.
- [13] Karweit Nancy, Slavin Robert E. Time-on-task: Issues of timing, sampling, and definition. // Journal of educational psychology. — 1982. — Vol. 74, no. 6. — P. 844.

- [14] Lahaderne Henriette M. Attitudinal and intellectual correlates of attention: a study of four sixth-grade classrooms. // Journal of educational psychology. — 1968. — Vol. 59, no. 5. — P. 320.
- [15] Lee Steven W, Kelly Karen E, Nyre Joseph E. Preliminary report on the relation of students' on-task behavior with completion of school work // Psychological Reports. — 1999. — Vol. 84, no. 1. — P. 267–272.
- [16] Linear regression. — 2017. — May. — URL: <https://goo.gl/WKjmxxy>.
- [17] Lloyd John W, Loper Ann B. Measurement and evaluation of task-related learning behaviors: Attention to task and metacognition. // School Psychology Review. — 1986.
- [18] Logistic Regression. — 2017. — May. — URL: <https://goo.gl/5jWqqG>.
- [19] Maris Eric. Psychometric latent response models // Psychometrika. — 1995. — Vol. 60, no. 4. — P. 523–547.
- [20] Mean Shift. — 2017. — May. — URL: <https://goo.gl/20MA3v>.
- [21] MiniBatch k-means. — 2017. — May. — URL: <https://goo.gl/vuT8Y7>.
- [22] Off-task behavior in the cognitive tutor classroom: when students game the system / Ryan Shaun Baker, Albert T Corbett, Kenneth R Koedinger, Angela Z Wagner // Proceedings of the SIGCHI conference on Human factors in computing systems / ACM. — 2004. — P. 383–390.
- [23] PCA. — 2017. — May. — URL: <https://goo.gl/jRGbqz>.
- [24] Pandas. — 2017. — May. — URL: <https://goo.gl/9oit1n>.
- [25] Pycharm. — 2017. — May. — URL: <https://www.jetbrains.com/pycharm/>.
- [26] Quinlan J. Ross. Induction of decision trees // Machine learning. — 1986. — Vol. 1, no. 1. — P. 81–106.

- [27] RFE. — 2017. — May. — URL: <https://goo.gl/6ulB1H>.
- [28] Random forest classifier. — 2017. — May. — URL: <https://goo.gl/oS3B0L>.
- [29] Scikit-learn. — 2017. — May. — URL: <http://scikit-learn.org/stable/>.
- [30] Scipy. — 2017. — May. — URL: <https://www.scipy.org/>.
- [31] Spectral Clustering. — 2017. — May. — URL: <https://goo.gl/wJMjqy>.
- [32] Stepik.org. — 2017. — May. — URL: <https://stepik.org/>.
- [33] Toward tutoring help seeking / Vincent Aleven, Bruce McLaren, Ido Roll, Kenneth Koedinger // Intelligent Tutoring Systems / Springer. — 2004. — P. 227–239.
- [34] Walonoski Jason, Heffernan Neil. Detection and analysis of off-task gaming behavior in intelligent tutoring systems // Intelligent Tutoring Systems / Springer. — 2006. — P. 382–391.
- [35] XGBoost. — 2017. — May. — URL: <https://xgboost.readthedocs.io>.
- [36] Yu Lei, Liu Huan. Feature selection for high-dimensional data: A fast correlation-based filter solution // ICML. — Vol. 3. — 2003. — P. 856–863.
- [37] numpy. — 2017. — May. — URL: <http://www.numpy.org/>.