Санкт-Петербургский государственный университет

Кафедра технологии программирования

Орлов Антон Сергеевич

Выпускная квалификационная работа магистра

Автоматизация процесса сбора и анализа данных об активности учащихся онлайн-курсов

Направление 020402

Фундаментальная информатика и информационные технологии

Научный руководитель,

канд. техн. наук,

Блеканов И. С.

Санкт-Петербург

2020

Содержание

Введение

Онлайн-курсы являются современным образовательным форматом, который отличается от очного обучения большим количеством данных, которые генерируются по результатам взаимодействия обучающихся с курсом. Такие данные подлежат анализу для выявления скрытых закономерностей. Обнаружение таких зависимостей позволит ответить на такие вопросы как:

1) Какая модель поведения слушателя во время работы с образовательными материалами приводит к получению наивысших результатов при сдаче контрольных заданий? Существуют ли различные модели поведения? Как определить, какая модель подойдёт слушателю, который начал проходить онлайн-курс?
2) В каком виде образовательные материалы воспринимаются лучше? Существует ли модель, которая позволяет предсказать, насколько эффективным будет онлайн-курс, на этапе его производства?
3) Как сделать обучение персонализированным? Хватает ли данных платформ онлайн-обучения для создания адаптивных онлайн-курсов?
4) Что можно понимать под образовательной эффективностью онлайн-курсов? Как оценить эффективность существующих образовательных материалов?

Решение этих вопросов позволит организациям-разработчикам онлайн-курсов повысить коммерческий потенциал своих курсов, а университетам позволит решать, какие курсы пригодны для внедрения в образовательные программы и могут использоваться наравне с очными дисциплинами, а какие лучше оставить для дополнительного изучения в качестве факультативов. Это подтверждается запросом ведущих ВУЗов страны на анализ данных своих курсов.

В Санкт-Петербургском Государственном университете (СПбГУ) разработкой, публикацией и поддержкой онлайн-курсов занимается Центр развития электронных образовательных ресурсов (ЦРЭОР). На данный момент было выпущено большое количество курсов, каждый из которых

прошло от 100 до 25000 уникальных слушателей. Курсы публиковались на различных платформах: “Открытое образование” (131 курс), основанная на edX и “Coursera” (82 курса). По всем курсам существует немалое количество собранных данных: журналы событий, оценочные листы, персональные данные слушателей. Данные, которые были собраны, зависят от платформы. Некоторые курсы опубликованы сразу на двух платформах, другие являются эксклюзивными для одной из двух платформ.

Для повышения качества онлайн-курсов СПбГУ было принято решение использовать накопленные данные для изучения формата онлайн-обучения. Чтобы провести этот анализ, сперва необходимо рассмотреть существующие инструменты, оценить их возможности и при необходимости расширить их дополнительными модулями.

Обзор литературы

Understanding Item Analyses [1]

Статья описывает исследование тестов различных видов. В ней особое внимание уделяется среднему баллу среди всех ответов на каждый вопрос. По среднему баллу выводятся параметры вопросов, описывающие тест с точки зрения сложности в различных её проявлениях. Авторы статьи не говорят об онлайн-курсах, а ориентируются на тесты, решаемые во время очных занятий, однако множество параметров можно вывести и для тестов в исследуемом формате обучения.

Basic Concepts in Item and Test Analysis [2]

В этой работе проделана схожая работа, однако с использованием верхних и нижних групп. Текст статьи посвящён анализу вопросов в тесте, однако делаются предположения об оценке теста в целом.

Помимо статей, в которых рассматривается анализ тестов, есть ещё множество различных исследований онлайн-курсов и образовательных форматов, которые посвящены исследованию других обучающих материалов:

- **How video production affects student engagement: an empirical study of MOOC videos**. [3] Исследование о том, как должны выглядеть видеоматериалы в рамках онлайн-курсов.
- **Optimal Video Length for Student Engagement**. [4] Статья с официального сайта edX, крупной образовательной платформы, в которой изучается оптимальная длина видео.
- **Адаптивность: с чего начать и нужно ли?** [5] Рекомендации для построения адаптивного обучения.

Studying learning in the worldwide classroom research into edX's first MOOC [6]

В статье, авторами которой являются исследователи из университета Massachusetts Institute of Technology, впервые поднимаются вопросы об анализе данных онлайн-курсов в результате разработки и публикации первого Massive Open Online Course (MOOC) в области компьютерных наук и электроинженерии. Авторы ставят вопросы: “что выполнимо экономически?”, “что политически возможно?”, “как применить результаты исследований онлайн-курсов для изменения нашего представления об очном обучении?”.

В качестве анализируемых данных у исследователей был следующий набор:

1) IP адреса всех слушателей курса;
2) Данные о 230 миллионах кликов, которые совершили слушатели на платформе онлайн-обучения;
3) Оценки за домашние задания;
4) Оценки за экзамены и лабораторные работы;
5) Публикации студентов и обучающего персонала в разделе обсуждения курса;
6) Результаты опроса 6002 учащихся по окончании ими прохождения онлайн-курса.

По этим данным были построены следующие распределения:

1) Количество уникальных слушателей, которые посетили курс на каждой неделе;
2) Количество обращений к обучающим материалам курса и контрольных заданий на каждой неделе;
3) Время, которое слушатели проводили на страницах с различными видами материалов, на каждой неделе;
4) Какими материалами слушатели пользовались во время выполнения домашних заданий, лабораторных работ и сдачи экзамена;
5) Географическое распределение собранных IP адресов;
6) Распределение возрастов учащихся;

7) Причины, по которым учащиеся записались на курс.

Анализ этих распределений не позволил исследователям найти корреляцию между возрастом, полом учащегося и его достижениями на курсе, но при этом на прохождение курса влияли предварительные знания в области математики. Также удалось найти незначительную зависимость между полученной учёной степенью и достижениями на курсе.

Вопросы, которые были поставлены в результате исследования, следующие:

1) Можно ли сказать, что успеваемость студентов, сдавших тест с первой попытки, “равна” успеваемости тех, кто решил его только со второго или третьего раза?
2) По каким причинам учащиеся перестают учиться на онлайн-курсах?
3) Как форумы обсуждений помогают слушателям решать задания?

Рассмотренное исследование можно расширить ещё и по следующим направлениям:

1) Анализ не одного, а нескольких курсов. Под этим подразумеваются различные направления изучения: сравнивать курсы разных форматов, разных научных областей и т. п.;
2) Сравнение результатов прохождения курсов разных слушателей, например студентов университета и вольных слушателей;
3) Выделение зависимостей между активностью во время работы с образовательными материалами и результатами сдачи контрольных заданий, проверяющих освоение этих материалов.

Mining theory-based patterns from Big data: Identifying self-regulated learning strategies in Massive Open Online Courses [7]

В этой работе исследовалось то, в каком порядке слушатели проходят онлайн-курсы. В статье делается вывод, что учащихся можно поделить на три категории: те, кто смотрит материалы в той последовательности, в которой они представлены разработчиками курса; те, кто сначала смотрит на контрольные задания и потом возвращается к обучающим материалам, и остальные

пользователи без определенной стратегии. В статье делается вывод, что учащихся можно поделить на три категории: те, кто смотрит материалы в той последовательности, в которой они представлены разработчиками курса; те, кто сначала смотрит на контрольные задания и потом возвращается к обучающим материалам, и остальные пользователи без определенной стратегии. Помимо этого, в статье исследуются материалы, которые используют учащиеся, так, например, выделяется целый класс слушателей, которые только смотрят видео. Также рассматривается, к каким материалам обращаются слушатели после просмотра видео. При этом исследуется не простая модель перехода из лекции в контрольное задание, но детальный процесс слушателей: исследуются переходы между следующими состояниями:

1) Начал смотреть видео
2) Закончил смотреть видео
3) Переход в контрольное задание
4) Попытка сдачи контрольного задания
5) Успешная попытка сдачи контрольного задания

Статья имеет огромное значение для проводимого в этой статье исследования. Данные, которые предстоит собирать, будут активно использоваться для построения персонализированных траекторий на платформе адаптивных курсов СПбГУ. Адаптивность онлайн-курсов на сегодняшний день достигается при помощи выстраивания траекторий для слушателей. Поэтому помимо добавления вспомогательных материалов, допустим вариант их ротации для лучшего усваивания слушателями. По этой причине необходимо рассмотреть вопросы порядка прохождения материалов онлайн-курсов в проводимом исследовании.

Постановка задачи

Для возможности глубокого исследования формата онлайн-курсов необходимо иметь удобные способы взаимодействия с данными, которые генерируются в ходе прохождения онлайн-курсов слушателями. Однако образовательные платформы предоставляют данные, которые не способны ответить на множество вопросов без обработки. Среди таких данных можно выделить:

1) Журналы событий. В них приводится детальное описание каждого действия пользователей. Однако эти данные не представлены в человекочитаемом виде и их необходимо обрабатывать.
2) Оценочные листы. С их помощью персонал онлайн-курсов способен получать информацию об оценках каждого слушателя за каждый урок. Однако оценки слушателей не позволяют ответить на множество вопросов. Сложно понять, о чём говорит высокий средний балл. Он может говорить как о низкой сложности курса, так и о высокой подготовленности обучающихся, или же о том, что в курсе присутствуют ошибки и их можно использовать для получения высоких оценок.

Необходимо учесть, что существуют инструменты, которые обрабатывают эти данные. Поэтому перед разработкой собственных методов необходимо оценить существующие инструменты.

Таким образом, цель работы заключается в повышении доступности данных об активности пользователей онлайн-курсов для последующего их анализа.

Для достижения цели необходимо решить следующие задачи:

1) Проанализировать существующие инструменты анализа данных онлайн-курсов;
2) Собрать требования к разрабатываемому ПО;
3) Разработать ПО для постоянного сбора данных с образовательных платформ и их хранения;

4) Разработать методы анализа получаемых данных с возможностью интеграции анализа в другие системы (такие как адаптивная платформа онлайн-курсов) и визуализации этого анализа для решения рабочих вопросов при создании онлайн-курсов.

По результатам работы необходимо представить совокупность программных продуктов (существующих и созданных в результате данной работы) и способов их использования для повышения доступности сведений об учащихся онлайн-курсов и их действиях.

Глава 1. Существующие инструменты для анализа данных

На данный момент существует несколько готовых инструментов, которые позволяют анализировать данные, либо собирать данные для самостоятельного проведения такого анализа. Для этого необходимо рассмотреть эти инструменты, оценить их возможности и сделать вывод о том, как можно использовать существующие инструменты при разработке ПО для анализа данных онлайн-курсов.

В качестве таких инструментов были рассмотрены образовательные платформы, на которых ЦРЭОР реализует свои курсы: Национальная платформа открытого образования (НПОО) [8] и Coursera [9]; платформы для анализа данных онлайн-курсов и платформы для анализа произвольных данных.

1.1 НПОО

“Открытое образование” – платформа для онлайн-обучения, в основе которой лежит ПО с открытым исходным кодом open edX. НПОО на сегодняшний день не имеет аналитического модуля, но при этом сама платформа open edX обладает встроенными решениями для анализа данных (они будут рассмотрены отдельно). При этом open edX позволяет собирать tracking logs [10]. Они содержат всю информацию о взаимодействии студентов с платформой. Примеры типов событий из tracking logs с их кратким описанием:

1. Pause_video. Генерируется при нажатии на кнопку паузы видео слушателями. Такие события содержат информацию о видео, о пользователе и о моменте видео, в который произошла пауза.
2. Problem_check. Генерируется при нажатии учащимся кнопки “Отправить” в контрольных и проверочных тестах. Содержит данные о пользователе, о контрольном вопросе и о том, какую оценку этот слушатель получил за этот вопрос.

Разработчики edX выделяют 22 группы таких событий. [10]

Можно сделать вывод о возможностях по анализу данных при помощи НПОО: присутствуют богатые возможности для сбора данных, но эти данные не анализируются НПОО.

1.2 Coursera

Coursera, в отличии от НПОО, предоставляет уже существующие возможности для анализа данных. Первая часть аналитики строится по всем данным онлайн-курсов организации, а вторая часть относится к конкретным курсам.

Предоставляемый анализ по данным всей организации включает в себя следующие показатели и аналитические срезы:

- Количество обучающихся на предыдущей неделе;
- Среднее количество дней, когда слушатели были активны;
- График, показывающий количество активных пользователей в каждый из дней;
- График, показывающий среднее количество активных пользователей в каждую из недель;
- Данные о курсах организации (будут рассмотрены во второй части, в панель организации выводится лишь часть этих данных);
- Количество записавшихся на курсы и при этом не учившихся ранее на курсах организации;
- Количество записавшихся на курсы и при этом учившихся ранее на курсах организации;
- График среднего количества записавшихся на курсы и при этом не учившихся ранее на курсах организации против среднего количества записавшихся и при этом учившихся ранее на курсах организации;
- Распределение количества курсов организации, на которые записывается один обучающийся;

- Количество “завершений” (сколько раз обучающийся прошел курс) за последний месяц;
- Распределение количества завершений за последний месяц по неделям;
- Количество завершений курсов, приходящихся на одного обучающегося;
- Количество обучающихся, получивших 80% баллов за курс;
- Детальная статистика завершений по каждому курсу;
- Оценки, которые слушатели поставили курсу;
- Отзывы слушателей;
- Распределение слушателей по уровню знаний в определенной области. Этот параметр указывается каждым слушателем;
- Динамика изменения количества слушателей на каждом уровне знаний (beginner, intermediate, advanced);
- Распределение попыток сдачи контрольных заданий по предметным областям онлайн-курсов;
- Просмотр распределения слушателей по профессиональной принадлежности (указывается слушателем онлайн-курса);
- Тренды областей в онлайн-курсах;

Предоставляемый анализ по каждому курсу включает в себя следующие пункты:

- Количество уникальных посетителей;
- Количество записавшихся на курс;
- Количество начавших изучение курса;
- Количество обучающихся, завершивших курс;
- Количество обучающихся, оплативших курс, распределенное по типам оплат (оплата организацией, оплата программой “финансовая помощь”, обычная оплата);
- График записавшихся на курс по дням;
- Количество отметок “мне нравится” и отметок “мне не нравится”;

- Количество отзывов о курсе;
- Сравнение рейтинга курса со средним рейтингов курсов на платформе/курсов страны организации;
- Распределение количества учащихся, дошедших до каждого модуля;
- Распределение количества учащихся, покинувших курс на каждом модуле курса;
- Возрастное распределение учащихся (заполняется учащимися);
- Географическое распределение учащихся;
- Распределение учащихся по статусу (работает полный рабочий день, безработный и т. п.);
- Распределение учащихся по учебной ступени.

Данные, которые используются для расчёта показателей и построения аналитических срезов, можно импортировать с Coursera.

Вывод, который можно сделать о Coursera как средстве анализа данных: количество исследований, которые позволяет проводить Coursera довольно велико, и они позволяют получить крайне детализированную картину о слушателях курсов. При этом можно заметить следующий недостаток: большинство исследований показывают только данные об обучающихся до прохождения курса (сведения, которые учащиеся оставляют о себе) и после прохождения курса (средний балл, количество окончивших курс, количество оплативших). При этом отсутствуют данные, отображающие работу с материалами курса в ходе его прохождения учащимися. В качестве примера можно рассмотреть работу с видео: аналитика Coursera не проводит детального анализа того, как учащиеся смотрят видео и смотрят ли они их вообще. При этом стоит отметить, что видео онлайн-курса являются его самой большой частью. На основании того, как учащиеся смотрят видео, можно далее изучать и зависимость оценок, получаемых учащимися в результате прохождения итогового испытания, от просмотра видео. Данных Coursera не хватит для проведения такого исследования.

1.3 edX Insights

edX Insights [11] – платформа для анализа данных платформы open edX. С её помощью возможно проводить следующие виды анализа:

1. Общие сведения о записавшихся на все курсы организации слушателей
2. Детализированный анализ записавшихся на определенный курс слушателей
3. Географическое распределение слушателей
4. Демографические показатели слушателей
5. Общий обзор количества слушателей, которые взаимодействуют с материалами курса
6. Общий обзор просматриваемости видео на курсе (какие видео слушатели смотрят чаще, какие фрагменты видео пересматриваются слушателями
7. Динамика оценок, получаемых слушателями
8. Активность во время работы с проверочными заданиями.

Такой анализ предусматривает изучение активности слушателей во время работы с образовательными материалами, но при этом отсутствует возможность изучения зависимостей между разными видами активности слушателей. Другим недостатком в рамках поставленной задачи является то, что edX Insights используется именно для анализа данных в формате open edX, в то время как ведущие ВУЗы на сегодняшний день часто используют несколько образовательных платформ, данные которых тоже необходимо анализировать.

1.4 ELK

Самым популярным решением для анализа логов является набор технологий Elasticsearch, Logstash, Kibana (ELK Stack или ELK) [12] компании Elastic. Он рассматривался вместе с дополнительной технологией той же компании Filebeat [13]. Возможности этих технологий следующие:

1) Мониторинг файлов. Filebeat может просматривать файлы и отслеживать изменения в них. В рамках этой задачи он может выполнять роль отслеживания новых событий в файлах логов. При добавлении новых событий он запускает сценарий, который может описать пользователь. В сценарии можно указать, в какие дальнейшие системы нужно отправлять события. Также присутствует возможность отправлять данные во множество различных систем в зависимости от вида полученного события. В разрабатываемой системе это необходимо для отправки логов в разные компоненты. Например, если появляется новое событие типа “play_video”, то оно отправится в систему анализа видео, а если “problem_check” – в систему анализа тестов.
2) Возможности Logstash схожи с возможностями Filebeat, но он не мониторит события, а получает их из других систем.
3) Elasticsearch – известная NoSQL база данных с ориентиром на полнотекстовый поиск. Позволяет хранить обработанные события и применять множество различных фильтров при извлечении этих событий для последующего анализа.
4) Kibana предоставляет возможности для построения анализа, построения графиков, карт, диаграмм и т.п.

Исходя из описания, эти инструменты позволяют решить поставленные задачи, однако в действительности их функционала недостаточно. Logstash оказывается ненужным из-за Filebeat, так как платформа онлайн-обучения не имеет заранее подготовленной возможности писать логи в Logstash, то удобнее поставить Filebeat на их мониторинг. Kibana обладает широкими возможностями по визуализации данных, однако реализовать трудные аналитические запросы на её основе становится либо слишком сложно, либо невозможно. Так, не получится исследовать, сколько раз был просмотрен каждый момент видео. Чтобы найти это значение, необходимо использовать алгоритм, который будет описан далее.

1.5 Apache Kafka

Kafka [14] используется для построения систем, обрабатывающих данные в реальном времени. Она горизонтально масштабируема, толерантна к ошибкам, обладает высокой скоростью и используется во множестве компаний.

Разрабатываемая система должна обрабатывать все логи, которые генерируются в результате прохождения онлайн-курсов пользователями. Если число пользователей будет расти, то будет расти и число логов, а значит и нагрузка на системы анализа. Если компоненты, отвечающие за анализ, не будут успевать анализировать событие до появления нового, то начнёт копиться очередь событий для обработки и через какое-то время в системе будут анализироваться совсем не новые события. Такая ситуация теоретически оправдана: количество слушателей онлайн-курсов со временем растёт, а значит и растёт количество событий, которые нужно анализировать.

Для решения этой проблемы необходимо использовать распределённые системы, обслуживать которые удобно при помощи Kafka. Эта система реализует шаблон “подписчик – издатель”. Если какая-то компонента перестаёт успевать обрабатывать события, то можно добавить в систему такую же компоненту на другом сервере, для ускорения процесса. Две системы будут обращаться к Kafka и получать из неё новые события. Таким образом можно избежать потенциальных проблем с производительностью.

1.6 Выводы по главе 1

В этой главе были рассмотрены возможностей нескольких инструментов, которые позволяют анализировать данные онлайн-курсов. Исходя из этого обзора можно сделать выводы о каждом инструменте.

1) НПОО не предоставляет аналитики данных об онлайн-курсах и их участниках, но осуществляет сбор подробных данных;

2) Coursera предоставляет множество аналитических срезов и показателей, но в них не учитывается детализированная активность слушателей во время изучения материалов и решения контрольных заданий. Такие срезы не позволят изучать активность слушателей. Существует возможность получения данных, используемых для создания этих аналитических срезов;
3) edX Insight является платформой для анализа данных. В ней используются данные, извлекаемые из платформы open edX, поэтому аналитические срезы сильно детализированы, хотя их намного меньше, чем в Coursera;
4) Существующие инструменты ELK Stack и Filebeat позволяют обеспечить постоянный сбор, хранение и визуализацию данных, но возможностей для обработки данных недостаточно для построения таких же аналитических срезов, как на Coursera или edX Insight;
5) Kafka может использоваться для передачи данных из одних инструментов в другие.

Таким образом, НПОО предоставляет богатые возможности для создания системы на основе данных, которые она генерирует и хранит. Coursera при этом отличается богатым анализом данных об учащихся на онлайн-курсах, но анализ можно расширить, исследуя активность слушателей во время просмотра видео и влияние этой активности. Однако данных, которые предоставляет Coursera, для такого анализа не хватает. edX Insights предоставляет некоторый анализ активности во время просмотра видео, но при этом отсутствует возможность изучения влияния одних видов активностей на другие.

Существующие инструменты предоставляют множество ограничений. По этой причине необходимо разрабатывать собственный инструмент для анализа данных онлайн-курсов.

Для использования существующих инструментов вместе с разрабатываемым ПО необходимо поставить следующие требования:

1) Возможность использования нескольких инструментов в качестве источников данных;
2) Возможность обработки данных различных форматов.

Для передачи данных из одних инструментов в другие возможно использовать Kafka. Для сбора, хранения и визуализации данных можно использовать технологии компании Elastic. При этом их возможностей недостаточно для построения любых аналитических срезов, поэтому необходимо учесть возможность добавления сервисов обработки данных для их визуализации.

Глава 2. Проектирование системы анализа данных

2.1 Обзор существующих компонент

Перед проектированием системы необходимо рассмотреть существующие системы, используемые в процессе создания и реализации онлайн-курсов, и особенности архитектуры этих систем.

Платформа анализа будет использоваться как для составления отчётов о реализации онлайн-курсов, так и для передачи данных в другие системы. Получать данные система будет из нескольких источников:

1) Разрабатываемая на данный момент платформа адаптивных онлайн-курсов СПбГУ
2) Промежуточные срезы журналов событий НПОО
3) Потенциально будут использоваться данные из Coursera.

Таким образом необходимо предусмотреть не только возможность загрузки данных сотрудниками ЦРЭОР в систему (то есть ручную загрузку данных пользователями системы), но и то, что данные сами должны автоматически попадать в систему, когда они генерируются в платформе СПбГУ (постоянно или с некоторым интервалом времени для снижения нагрузки на сервисы).

Для платформы СПбГУ на данный момент выделены несколько серверов. Предполагается, что запуск платформы произойдёт с использованием четырех серверов:

1) Сервер MongoDB для данных о курсах
2) Сервер MySQL для хранения данных о пользователях
3) Сервер платформы онлайн-курсов СПбГУ
4) Сервер для хранения видео онлайн-курсов.

Данные журналов событий генерируются на сервере, где размещена платформа онлайн-курсов. Дополнительные сведения можно получить из серверов баз данных. При разработке адаптивной платформы онлайн-курсов будет использоваться база данных с курсами.

Таким образом, поступать данные в систему будут с серверов 1, 2 и 3, а отправляться после их анализа они должны на сервер базы данных 1.

2.2 Архитектура системы анализа данных

Обзор существующих компонент позволяет определить место создаваемой компоненты относительно существующих систем. Для этого рассмотрим следующую схему (рис. 1).

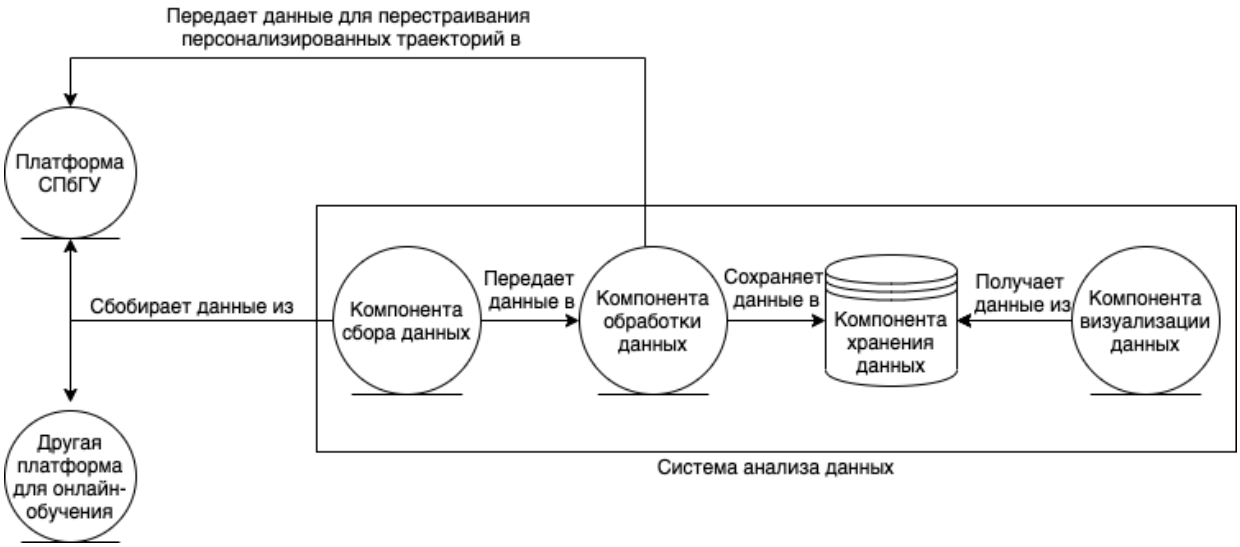


Рис. 1 Процесс анализа курсов

Если платформа СПбГУ будет частью системы анализа данных, то это не нарушит существующие процессы, но позволит использовать её для повторного использования компонент платформы. Таким образом необходимо решить, должна ли система анализа данных быть отдельным продуктом или ее можно разрабатывать на базе платформы СПбГУ.

Для решения этого вопроса необходимо отметить, что Open edX постоянно развивается. Каждый год выходит большое обновление платформы. На эти обновления ориентируются разработчики плагинов, которые решают множество задач электронного обучения. Если строить систему на основе существующей архитектуры, это неизбежно внесет изменения в кодовую базу Open edX. Нарушения в кодовой базе сделают процесс обновления сложнее.

Кроме того, встраивание системы в существующую платформу ограничит возможности масштабирования платформы. Если нагрузка на какую-то часть системы анализа данных возрастет и это приведет к снижению производительности, а возможности увеличить мощность используемого сервера не будет, придется использовать дополнительный сервер. В таком случае, вместо развертывания одного сервиса, придется развертывать дополнительную инсталяцию целой платформы.

При этом нельзя не учитывать то, что хоть и open edX является приоритетным источником данных для анализа, не исключено использование в будущем и других платформ для разработки адаптивных онлайн-курсов. Встраивание в Open edX для сторонних организаций, не использующих Open edX, вызовет дополнительные издержки с их стороны.

Исходя из этого, разработка платформы анализа данных видится исключительно в качестве отдельного продукта.

2.3 Процесс анализа данных

В предыдущем параграфе были рассмотрены процессы, которые вводят данные в систему и забирают данные, но необходимо рассмотреть и внутреннее устройство системы.

Процесс анализа данных состоит из трёх основных этапов:

1) Сбор данных
2) Предобработка данных
3) Анализ данных.

Процесс предобработки данных должен быть включен в процесс анализа данных для решения проблем, указанных в главе 1, из-за которых некоторые данные невозможно проанализировать с использованием существующих систем.

Каждый слой должен иметь возможность к расширяемости:

1) При сборе данных могут использоваться различные источники, и эти источники могут добавляться в будущем;

2) Предобработчики данных могут добавляться, поскольку предполагается, что в системе будут постоянно реализовываться новые анализаторы;
3) Количество методов анализа данных постоянно увеличивается, потому что появляются новые алгоритмы и образуются новые потребности; они будут использоваться с разными целями (визуализация анализа, использование в системе подбора персонализированных траекторий прохождения онлайн-курсов и этот список может расширяться в будущем).

Таким образом можно заключить, что внутреннее устройство системы анализа данных должно включать в себя три слоя, каждый из которых выполняет подпроцесс анализа данных. При этом необходимо обеспечить расширяемость.

Потребность в функциональной масштабируемости и постоянное расширение количества слушателей онлайн-курсов приводит к тому, что система должна масштабироваться для сохранения производительности при повышении нагрузки.

2.4 Проект системы

Резюмируя предыдущие параграфы этой главы, можно сделать следующие выводы: система не будет использовать части Open edX, а будет самостоятельным продуктом, способным интегрироваться с любыми платформами (путем разработки дополнительных модулей сбора данных для этих платформ); платформа должна иметь три слоя, каждый из которых обладает возможностью масштабирования.

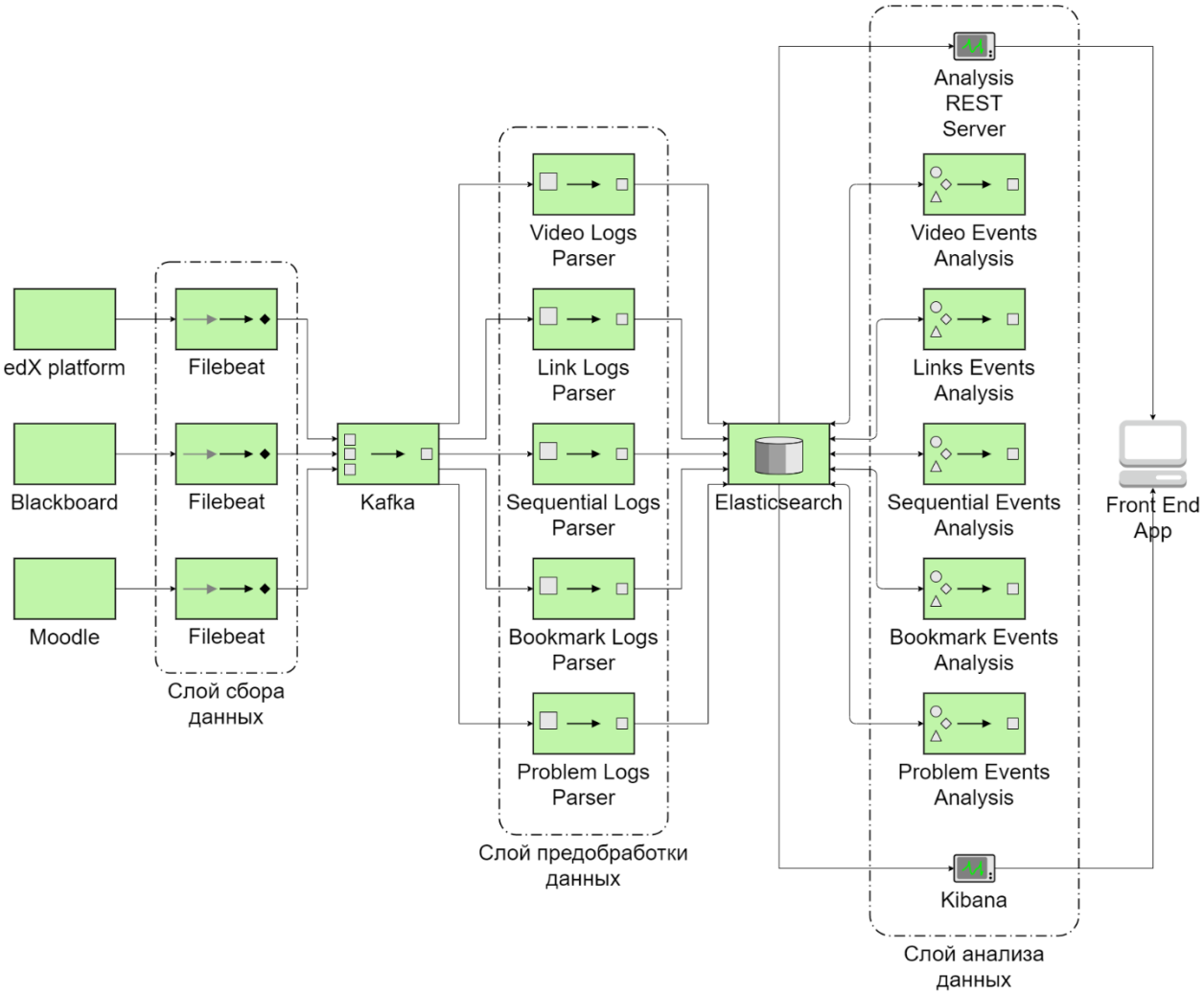


Рис. 2. Проект системы анализа данных

С учётом этих особенностей был составлен проект системы (рис. 2). Проект системы не является точным описанием платформы. При проектировании был допущен определенный набор условностей. Так, в качестве компонент edX platform, Blackboard, Moodle должны быть определенные платформы (Blackboard СПбГУ, Платформа онлайн-обучения СПбГУ (edX), НПОО, Coursera). Слой обработчиков данных тоже может расширяться. Например, если новые данные имеют формат не логов, а выгрузок, то для этого к системе достаточно присоединить дополнительный модуль, который будет превращать данные в тот же формат, в который, например, преобразовываются логи edX, после чего их можно будет анализировать вместе. Такая возможность, безусловно, будет полезна при исследовании онлайн-курсов:

существует ряд одинаковых курсов, размещенных на разных платформах, что даёт возможность проанализировать влияние площадок реализации онлайн-курсов.

Каждый слой является набором отдельных друг от друга сервисов. Слои сбора данных и предобработки данных связаны при помощи Kafka. Слои предобработки данных и анализа данных связываются при помощи Elasticsearch базы данных. Таким образом достигается постоянное хранение необходимых сведений о слушателях курсов, и данные могут переиспользоваться для различных исследований.

2.5 Выводы по главе 2

В этой главе был произведен обзор всех существующих компонент, которые могут быть интегрированы в создаваемой платформой. Для этого были рассмотрены процессы, которые могут потребовать этой интеграции. Таким образом удалось зафиксировать потребность в интеграции с несколькими системами, которые будут отправлять данные в систему.

Для процесса анализа данных были рассмотрены его основные этапы. Каждый этап требует масштабируемости: будет появляться запрос на расширение функционала системы и на увеличение мощностей. По этой причине система должна иметь несколько отдельных слоев.

С учетом указанного анализа был составлен проект системы. Каждый слой представлен набором независимых сервисов, что позволит расширять функционал системы (добавлением новых независимых сервисов) и мощность системы (использованием новых серверов, на которых будут запускаться копии существующих сервисов). Слои связываются друг с другом при помощи Kafka и при помощи Elasticsearch.

Глава 3. Разработка прототипа

Прежде чем приступать к разработке прототипа, необходимо определиться с выбором технологий.

3.1 Используемые технологии

В главе 1 приведен обзор существующих технологий. В этот обзор включен инструмент, называемый ELK Stack – набор технологий Elasticsearch, Logstash, Kibana. Этот инструментарий подходит для анализа логов, но некоторые из потенциальных задач, решаемых при помощи создаваемого в этой работе продукта, эти технологии решить не могут. Таким образом, использование только ELK недостаточно, поэтому рассмотрим дополнительные компоненты, которые были включены в систему.

Filebeat

В первую очередь необходимо иметь инструмент для постоянного перенаправления генерируемых данных на платформе в систему анализа. ELK такой возможности не имеет, однако инструмент Filebeat от этой же компании разработан как раз для этих целей. Для Filebeat устанавливается файл, который он постоянно мониторит и при появлении новых строк, отправляет их в другой сервис.

Kafka

Поскольку Filebeat именно отправляет данные, это может создать затруднения для масштабирования системы. С этой целью перед отправкой новых событий на предобработку используется Kafka.

Разные виды событий отправляются в разные очереди Kafka. Filebeat позволяет на этапе сбора данных определить вид события и в зависимости от этого использовать ту или иную очередь Kafka. При этом эти события можно считать тогда, когда на их обработку будут ресурсы.

Без Kafka работа системы выглядела бы следующим образом: Filebeat отправляет события определенного типа определенному обработчику по его адресу. Если обработчик перестаёт успевать обрабатывать события, то аналитические срезы становятся все менее и менее актуальными. При этом в условиях развивающихся систем, которые повышают свою аудиторию ежедневно (что наблюдается в сегодняшних реалиях систем онлайн-обучения), растёт не просто функция, которая определяет отставание системы от актуальности, а её производная. Для решения этой проблемы придётся подключать новый сервис и менять конфигурацию Filebeat, чтобы он отправлял данные в несколько мест. При этом если сервисы располагаются на разных по мощности серверах, они всё равно будут получать одинаковую нагрузку. Это обяжет создавать балансировщик нагрузки между Filebeat и обработчиком.

В случае, когда между Filebeat и обработчиком стоит Kafka, такой проблемы нет. Все события изначально отправляются в Kafka, где они хранятся до тех пор, пока не будут запрошены сервисом обработки данных. Последний, в свою очередь, будет запрашивать их только тогда, когда у него есть ресурсы. При этом, если в Kafka начнут копиться события, потребуется включить копию этого обработчика на свободных серверах. На данный момент решение о развертывании дополнительной инсталяции сервисов принимает разработчик, но в будущем, при повышении количества серверов, возможно реализовать сервис, который будет управлять этим процессом. Для развертывания дополнительного сервиса никаких дополнительных настроек конфигурации указывать не нужно.

Go

Обработчики данных имеют следующие особенности, которые могут повлиять на выбор языка программирования:

1) Множество маленьких сервисов
2) Работа с JSON и XML

3) Использование многопоточности.

Каждую из перечисленных задач можно эффективно решить при помощи Go. При этом не исключено, что и другие языки программирования могут справиться с этими задачами. Однако при выборе языка программирования из списка известных автору языков (Python, C#, Java) Go показался самым подходящим для этого проекта: Java и C# чаще всего используются для разработки крупных монолитных проектов, чем для создания маленьких сервисов, а Python не обладает статической типизацией. Статическая типизация имеет свои плюсы и минусы, так что выбор таких языков для разработки крупных проектов определяется желанием разработчиков использовать такие языки, как и в случае этого проекта. Так или иначе, каждый новый сервис может быть написан на том языке программирования, который лучше подойдёт для очередной задачи. Если в будущем сервис будет необходимо поместить в сложную систему со большим объёмом бизнес-логики, то для разработки этой системы возможно использовать Java и при этом взаимодействовать с разрабатываемой системой без каких-либо трудностей.

3.2 Особенности реализации системы

Для обеспечения масштабируемости системы все сервисы можно запустить параллельно на нескольких серверах. Чтобы запуск нового сервиса был максимально простым, используются файлы конфигурации и технология контейнеризации.

При помощи конфигурационных файлов настраиваются все внешние компоненты, которые используются в этой системе. Для Filebeat конфигурация требует особенной настройки, поэтому рассмотрим её детально. Для того, чтобы Filebeat самостоятельно решал, в какую тему Kafka отправить каждое событие, необходимо описать правила в конфигурационном файле. Для журналов событий в формате edX главным критерием, по которому события можно отнести к разным темам, является поле “event_type”. Для

реализации этого распределения (который имеет формат YAML) добавляется поле `output.Kafka`, для него устанавливается хост, по которому доступна Kafka, и сжатие, при этом важно согласовать его при извлечении события из Kafka в обработчике и использовать такой же алгоритм. Далее устанавливается параметр `topics`. Это массив, каждый элемент которого определяет правила попадания в определенную тему. Например, чтобы отправлять события видео в тему с видео, использовалась следующая конфигурация (листинг 1).

```
- topic: "VideoEvents"
      when.or:
        - equals:
            event_type: "seek_video"
        - equals:
            event_type: "pause_video"
        - equals:
            event_type: "stop_video"
        - equals:
            event_type: "play_video"
```

Листинг 1. Конфигурация Filebeat для передачи событий просмотра видео в Kafka

Контейнеризация сегодня используется во множестве проектов, поскольку она позволяет упростить множество моментов во время разработки. Особенно выгодно использовать контейнеры при наличии большого количества сервисов. Для развертывания нескольких сервисов необходимо убедиться, что они будут работать на всех операционных системах, на которых они могут быть развернуты, учесть все проблемы заранее, написать скрипты, которые бы устанавливали необходимые зависимости, подготавливали и запускали эти сервисы. Ситуация меняется при использовании контейнеров (Docker-

контейнеров). В этом случае среда настраивается при помощи Dockerfile и docker-compose файлов. На сервере, где будет развернута система, необходимо только наличие Docker, который доступен на всех самых популярных операционных системах (Windows, macOS, Linux). При этом для разработки не придётся менять среду персональных компьютеров, что тоже является преимуществом этого способа разработки.

Первый контейнер в системе развёртывался из образа docker.elastic.co/beats/Filebeat. Для добавления в систему логов нужно поместить файл конфигурации и файлов логов в соответствующие им директории, чего можно достичь настройкой параметра volumes в docker-compose файле (листинг 2).

```
Volumes:
  - ./config/Filebeat.yml:/usr/share/Filebeat/Filebeat.yml:ro
  - type: bind
    source: ./build/logs
    target: /usr/share/Filebeat/edx-logs
    read_only: true
```

Листинг 2. Конфигурация docker-compose для добавления логов в Filebeat

Таким образом файл конфигурации и папка с логами будет видна внутри контейнера, но не будет туда продублирована, что позволит не тратить лишнюю память. В данном примере в папке config располагается файл конфигурации Filebeat.yml; в папке build/logs располагаются файлы логов. Помимо этого, необходимо настроить Filebeat конфигурацию (листинг 3).

```
Filebeat.inputs:
    - type: log
      enabled: true
      paths:
        - /usr/share/Filebeat/edx-logs/*.log
      json.keys_under_root: true
```

Листинг 3. Конфигурация входов Filebeat

Эта настройка указывает, что нужно брать все файлы из папки /usr/share/Filebeat/edx-logs с расширением .log и отслеживать появление новых событий в них.

Для Kafka использовались два контейнера: контейнер для Kafka wurstmeister/Kafka и контейнер с zookeeper, без которого Kafka не работает, wurstmeister/zookeeper. Настроить в данном контейнере нужно было только инициализацию тем, в которые будет писать события Filebeat (листинг 4).

```
KAFKA_CREATE_TOPICS: "VideoEvents:1:1, TestEvents:1:1, Sequen
tialEvents:1:1, BookmarksEvents:1:1, LinksEvents:1:1"
```

Листинг 4. Конфигурация инициализации тем Kafka

Цифры после названия тем обозначают replication factor и количество partition, которые необходимы для обеспечения надежности и оптимизации системы, когда количество сервисов, читающих одну тему, начинает расти. На данный

момент количество partition равно единице, потому что текущая нагрузка не требует увеличения сервисов, читающих одну очередь, а replication стоит оставлять равным единице, потому что данные хранятся в том месте, где они генерируются и могут быть восстановлены.

Для ELK использовались контейнеры:

- docker.elastic.co/Elasticsearch/Elasticsearch для Elasticsearch
- docker.elastic.co/Kibana/Kibana для Kibana.

В контейнер с Kibana необходимо передать конфигурационный файл (листинг 5).

```
volumes:
    - type: bind
      source: ./build/Kibana/config/Kibana.yml
      target: /usr/share/Kibana/config/Kibana.yml
      read_only: true
```

Листинг 5. Передача файла конфигурации Kibana в контейнер

В конфигурации указывается название сервера, название хоста Kibana; адрес Elasticsearch (можно указать несколько) – это необходимо для того, чтобы Kibana могла автоматически брать данные из индексов Elasticsearch и анализировать их.

Разработанные сервисы также имеют свои конфигурации и контейнеры. Для них используется базовый контейнер golang, в который затем помещаются файлы кода и запускаются внутри него. Сервисы имеют конфигурацию, в которой указывается, какой адрес имеет Kafka и Elasticsearch.

3.3 Обработка данных edX

Из Kafka события необходимо забирать, выделять из них нужные поля и формировать из них объекты удобные для анализа. На Go существуют библиотеки для взаимодействия с необходимыми компонентами. Так, были использованы: библиотека github.com/olivere/elastic для взаимодействия с ElasticSearch и библиотека github.com/segmentio/Kafka-go для взаимодействия с Kafka. Один из возможных видов анализа, который нужно проводить регулярно – определение количества просмотров каждого фрагмента видео. Для этого нужно выделить моменты для каждого видео, в которых нажимались кнопки паузы и воспроизведения видео. Для этого рассматриваются 4 вида событий: pause_video, stop_video, seek_video, play_video. Они приводились к типам, которые в Go определяются следующей структурой (листинг 6).

```
type VideoEventDescription struct {
    EventTime string    `json:"event_time"`
    VideoTime float64   `json:"video_time"`
    Username  string    `json:"username"`
    VideoID   string    `json:"video_id"`
    EventType EventType `json:"event_type"`
}
```

Листинг 6. Структура события просмотра видео

EventType в данном случае – переопределённый тип строки, который принимает 2 значения: pause и play. Очевидно, что play_video приводился к типу play, а pause_video и stop_video к pause. Seek_video приводился к pause, так как платформа считает, что seek_video – это перемотка с дальнейшей паузой при воспроизведении (в точке, к которой перематывали). Для данного

вида анализа не важно, что перед паузой происходила перемотка, поэтому извлекались только сведения о новом моменте видео.

Событие сначала необходимо прочитать из Kafka. Для этого необходимо создать объект типа Kafka.Reader с указанием адреса Kafka и темой, из которой он будет читать. Дополнительно можно указать partition и минимальное, максимальное количество байт, получаемых из Kafka. Тогда у объекта Kafka.Reader можно вызвать метод ReadMessage, который вернет следующее сообщение в очереди.

Затем событие необходимо преобразовать к объекту, который уже будет записываться в БД. Существует два способа: превратить полученное сообщение из Kafka в map или создать новый тип. И тот, и другой способ позволяют решить задачу, однако был выбран второй из-за возможности не описывать логику обращения к полям map и извлечению полей из вложенных объектов, а просто описать структуру одного события. Для описания структуры события видео создан следующий тип (листинг 7).

```
// VideoLog is a definition of a log object with event type
// "play_video", "pause_video", "stop_video" and "seek_video"
type VideoLog struct {
    Username     string        `json:"username"`
    EventType    string        `json:"event_type"`
    Time         string        `json:"time"`
    Event        VideoEventLog `json:"event"`
    VideoContext LogContext    `json:"context"`
}

// VideoEventLog is a definition of an event object within
// VideoLog
type VideoEventLog struct {
    CurrentTime float64 `json:"currentTime"`
```

```
    OldTime     float64 `json:"old_time"`
    ID          string  `json:"id"`
}
```

Листинг 7. Структура события просмотра видео в журнале событий

К сообщению, полученному из Kafka, применяется метод json.Unmarshal, в который дополнительно передаётся структура объекта, который нужно получить. Затем поля этого объекта переносятся в объект, который далее необходимо отправить в Elasticsearch.

Помимо видео, рассматриваются ещё несколько видов событий:

1. События добавления элементов курса в закладки. Среди таких событий выделяются: имя пользователя, который добавил эту закладку, время создания закладки, ID и ID курса элемента, который был добавлен в закладки и специальный флаг IsAdded, который показывает, обозначает ли это событие добавление в закладки или наоборот отмечает убранную закладку. Для ссылок рассматривался ID курса, где произошел переход по ссылке; имя пользователя, нажавшего на ссылку; время нажатия на ссылку; текущий URL и URL ссылки;
2. События, связанные с ответами на контрольные задания, определяются ID задания и курса, именем пользователя, который отвечает на вопросы теста, полученным результатом и максимальным возможным результатом;
3. Sequential, что означает группу объектов курса (модуль курса), определяет событие перехода между последовательными элементами курса (например, переход к следующей лекции) и обладает ID старого и нового элемента, ID курса, именем пользователя и временем перехода.

После обработки события помещаются в Elasticsearch. После обработки данные можно анализировать в следующем слое.

3.4 Обработка файлов структуры курсов

Для обработки структуры курсов было принято решение создать отдельную библиотеку. Любой исследователь или программист, желающий взаимодействовать со структурой курсов, обязательно столкнется с задачей хранения структуры в виде структуры своего языка программирования. При этом структуру курса получают не только администраторы платформы, имеющие доступ к базе данных, но и пользователи системы в роли “создатель курсов”. Так, ЦРЭОР, будучи клиентом НПОО и не имея доступ к их базам данных, может получить структуры курсов из меню создания курса. Аналогично может поступить любой другой создатель онлайн-курсов. По этой причине количество исследователей этого вопроса потенциально высоко.

В первую очередь, рассмотрим, что из себя представляет структура курса (рис. 3).

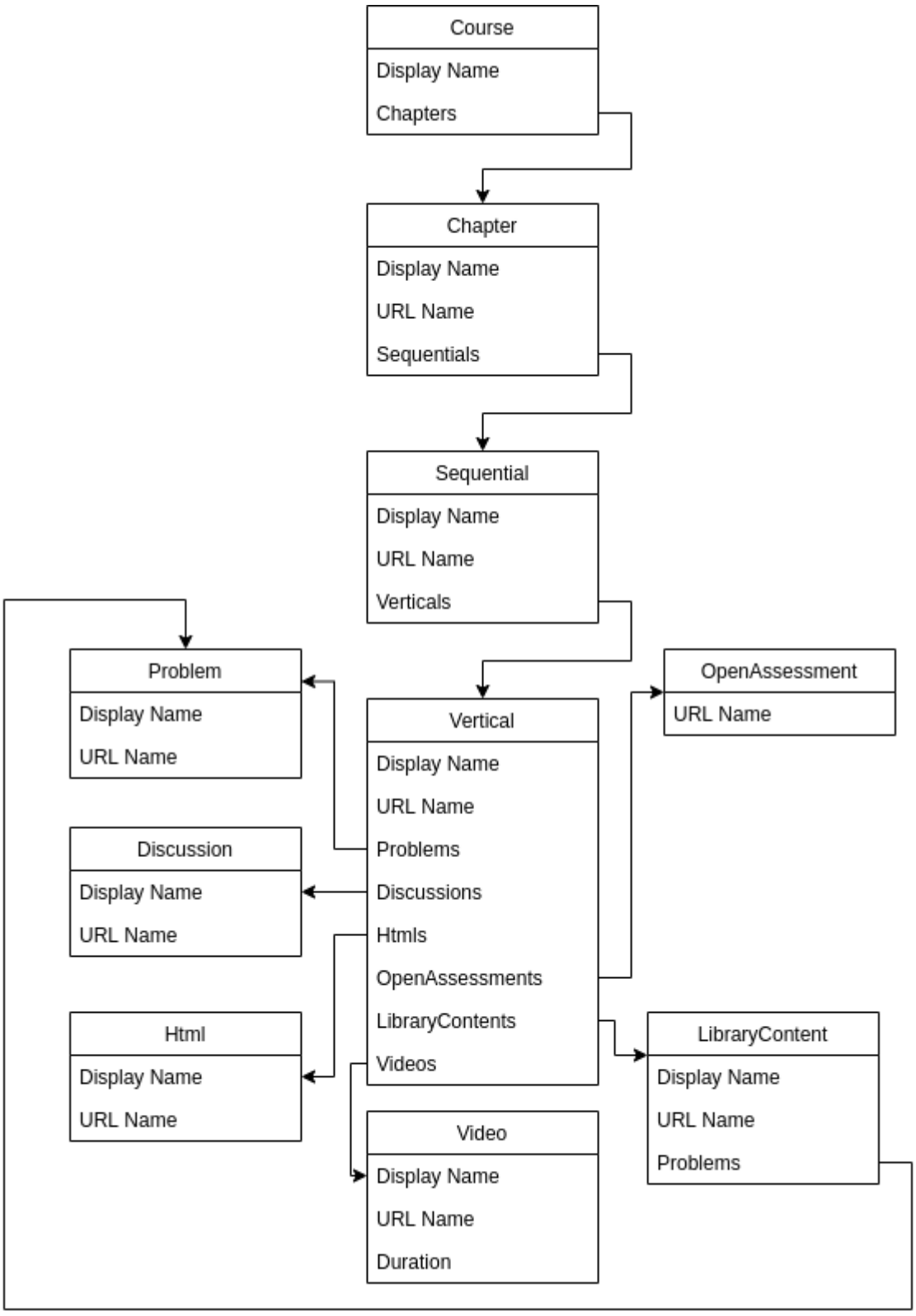


Рис. 3 Структура онлайн-курса

Таким образом, курс представляет из себя иерархию: курс (course), модуль(chapter), урок (sequential), блок (vertical). Эта иерархия использовалась при анализе маршрута, которым следуют слушатели в рамках прохождения онлайн-курса.

Структура извлекается в виде архива с XML файлами. Каждый файл представляет собой объект с указанием его потомков согласно иерархии. Проблема, которая возникает при парсинге файлов заключалась в том, что стандартная XML библиотека Go предполагает, что объект со всеми его потомками лежит в рамках одного файла, но у edX каждый объект содержит ссылки на своих потомков, информация о которых хранится в отдельных файлах. Из-за этого необходимо проходить итеративно по потомкам на каждом уровне иерархии и обрабатывать каждый отдельный файл во вспомогательную структуру, а затем переносить значения полей вспомогательной структуры в финальную.

3.5 Выводы по главе 3

Описанный в этой главе прототип позволяет обеспечить постоянную отправку данных из платформ для онлайн-обучения в систему анализа данных, где данные обрабатываются и записываются в базу данных Elasticsearch для последующего анализа. Для каждого сервиса использовалась контейнеризация в виде Docker-контейнеров. Это позволяет упростить разработку и развертывание системы. Добавление нового обработчика данных будет производиться путем создания нового контейнера и указанием адреса Kafka и Elasticsearch в его конфигурации. В ситуации, когда обработчик перестанет справляться со своей нагрузкой, необходимо развернуть такой же обработчик на свободном сервере. Благодаря Kafka, он не будет конфликтовать с уже запущенным сервисом.

Глава 4. Анализ данных онлайн-курсов

В этой главе представлены некоторые виды анализа данных, которые реализованы на сегодняшний день и которые используются в ЦРЭОР для исследования своих онлайн-курсов.

4.1 Связь географического распределения и успеваемости

На данный момент для реализации прототипа анализа географического местоположения слушателей использовался язык Python. По итогам прототипирования полученные результаты могут быть реализованы на любом языке для встраивания в систему в качестве анализатора, который будет использоваться в системе.

Для решения задачи применялись следующие технологии:

1) NumPy. В данном исследовании необходимо выполнять различные математические преобразования, которые уже реализованы в NumPy;
2) Matplotlib для построения графиков оценок;
3) Csv. Базы данных, в которых содержится информация об IP-адресах, имеют формат csv.

Для визуализации использовались Google Maps, а именно:

1) Google Maps JavaScript API для отображения полученной информации на карте. Однако для этого потребуется знать координаты исследуемых регионов, поэтому необходим следующий инструмент;
2) Google Maps Geocoding API выполняет сопоставление координат географического объекта его названию.

Для исследования среднего балла необходимо использовать события отправки на проверку контрольных заданий, то есть результат работы problem logs parser.

Также, представляется интересным географическое местонахождение пользователя. Для этого можно добавить в обработку событий сдачи контрольных заданий считывание IP адреса или создать дополнительный

обработчик и дополнительный индекс в Elasticsearch, который будет сохранять только IP адрес и username. Так как для каждого пользователя генерируется достаточно много событий, то IP адрес в первом сценарии будет дублироваться большое количество раз. Поэтому на данный момент был выбран второй вариант. При этом необходимо учесть, что одни и те же события будут обрабатываться несколько раз. В будущем необходимо оценить, как такой повторный анализ будет влиять на нагрузку системы и выбрать оптимальный вариант, исходя из возможностей серверов.

Из событий сдачи контрольных заданий выбираются данные о полученной оценке и максимальной возможной оценке. Так как максимальное число возможных баллов за задание везде разное, в качестве показателя оценки использовалось соотношение weighted_earned/weighted_possible, имеющее значение от 0 до 1.

Поскольку каждому пользователю сопоставлен IP адрес, можно получить словарь, имеющий в качестве ключей IP адреса, а в качестве значений — средний балл учащегося.

Для того, чтобы сопоставить IP адресу регион, необходимо воспользоваться существующими базами данных, которые содержат необходимую информацию. Такие базы данных распространяют различные сервисы. В данной работе был взят ip2location (URL: https://www.ip2location.com/, дата визита 02.06.2020). Он предоставляет несколько видов баз данных с нужной информацией. Для проводимого исследования достаточно данных, взятых из таблицы под названием “DB3.LITE”, которая имеет 6 столбцов: первый IP адрес в блоке, последний IP адрес в блоке, код страны, название страны, название региона и название города.

Первый и второй столбцы отсортированы по возрастанию. Это помогает ускорить операцию сопоставления IP-адресам регионов. Для этого необходимо взять словарь с IP-адресами и средними баллами. Каждый IP-адрес, который находится в ключах словаря, необходимо преобразовать в тип

данных int, так как в таком формате они хранятся в базе данных. После преобразования словарь сортируется по ним в порядке возрастания.

С подготовленными данными процедура проверки региона выполняется в цикле по всем ключам словаря в порядке, полученном после сортировки. Каждая итерация цикла заключается в следующем:

1) Проверить, принадлежит ли текущий IP адрес интервалу от значения в первом столбце текущей строки базы данных до значения во втором столбце. Если да, перейти на шаг 2. Иначе взять следующую строку базы данных и повторять этот процесс, пока адрес не будет содержаться в интервале.
2) Если третий столбец текущей строки базы данных имеет значение ‘RU’, то к ключу со значением региона (находится в пятом столбце) добавить все элементы текущего элемента словаря с IP адресами. Иначе переходить к следующей итерации цикла.

Следующий код позволяет выполнить эту процедуру (листинг 8).

```
for ip in sorted(ipint_mark.keys()):
    cur_row = next(ip_db)
    while ip > int(cur_row[1]):
        cur_row = next(ip_db)

    if cur_row[2] == 'RU':
        if cur_row[4] in report_regions.keys():
            report_regions[cur_row[4]].extend(ipint_mark[ip])
        else:
            report_regions[cur_row[4]] = ipint_mark[ip]
```

Листинг 8. Составление словаря оценок по регионам

В результате описанных действий был составлен словарь, который имеет в качестве ключей регионы РФ, а в качестве значений – список оценок за каждую попытку решения контрольного задания пользователями с IP соответствующих этим регионам. Далее необходимо выяснить, как правильно считать средний балл по региону.

При рассмотрении среднего балла необходимо исключить результаты регионов, в которых не набралось достаточного количества оценок и среднее сильно меняется при добавлении новых.

Пусть a – сумма оценок за n оценок в регионе. Тогда b – оценка, не входящая в сумму для a. Рассмотрим изменение среднего значения при добавлении оценки b:

$$\left|\frac{a}{n} - \frac{a+b}{n+1}\right| = \left|\frac{an + a - an - bn}{n^2 + n}\right| = \left|\frac{a - bn}{n^2 + n}\right| = \left|\frac{a}{n^2 + n} - \frac{b}{n+1}\right| \leq \left|\frac{a}{n^2 + n}\right|$$

Такое неравенство имеет место быть в силу ограничений:

$$\begin{cases} a \in [0, n] \\ b \in [0, 1] \\ n \in N \end{cases}$$

Максимальное значение $\max(a) = n$ следовательно:

$$\left|\frac{a}{n^2 + n}\right| \leq \left|\frac{n}{n^2 + n}\right| = \left|\frac{1}{n+1}\right| \underset{n \to \infty}{\longrightarrow} 0$$

Таким образом изменение среднего при добавлении новой оценки стремится к 0. Поэтому при достаточно высоких n оценка меняется на малое значение. В качестве этого значения было взято 0,001. Количество оценок n, при котором изменения оценки меньше 0.001, равно 999.

Среди рассмотренных логов имеется 54 региона с более чем 999 оценками. Для них было посчитано среднее. Эти результаты можно отобразить на карте для наглядности.

Для отображения результатов на карте использовались Google Maps JavaScript API [15]. Чтобы не просто отобразить регионы, а отразить и величину среднего балла, использовались heatmaps [16]. Для каждого региона были найдены

координаты на карте с помощью сервиса Geocoding API [17], входящего в Google Maps API.

Инструкцию по отображению можно найти в документации Google Maps, однако для адаптации требуется внести следующие изменения.

Перед отображением необходимо провести преобразование: для каждого среднего балла a значение для размера круга, отображающего величину среднего балла, необходимо брать $2^{25(a-0.5)}$для того, чтобы различия между оценками были хорошо заметны визуально. Эти изменения необходимо отобразить в функции getCircle. Для отображения она должна выглядеть следующим образом (листинг 9).

```
function getCircle(avgMark) {
    return {
        path: google.maps.SymbolPath.CIRCLE,
        fillColor: 'red',
        fillOpacity: .2,
        scale: Math.pow(2, (avgMark-0.5)*25),
        strokeColor: 'white',
        strokeWeight: .5
    };
}
```

Листинг 9. Функция сопоставления параметров круга для отображения на карте

Помимо этого, требуется изменить данные, т. е. сформировать GeoJSON из словаря со значениями средних баллов по региону. Код, с помощью которого данные добавляются на карту имеется в документации (листинг 10).

```
function eqfeed_callback(results) {
    map.data.addGeoJson(results);
}
```

Листинг 10. Формирование GeoJSON по данным

В связи с этим GeoJSON объект подаётся в качестве аргумента функции eqfeed_callback. Сам объект выглядит следующим образом (листинг 11).

```
{
    "type":"FeatureCollection",
    "features":[]
}
```

Листинг 11. Пример объекта GeoJSON

Features – массив, состоящий из объектов следующего вида (листинг 12).

```
{
    "type":"Feature",
    "properties":{
      "avgMark": 0.71
    },
    "geometry":{
      "type":"Point",
      "coordinates":[66.948278, 56.9634387]
    }
}
```

Листинг 12. Пример объекта массива features

После этого можно сослаться на файл с этими данными в качестве источника (листинг 13).

```
script.src = ‘points.geojsonp’
```

Листинг 13. Подключение файла с данными о точках в основном файле JavaScript

Распределение баллов изображено кругами на карте на рисунке 4.

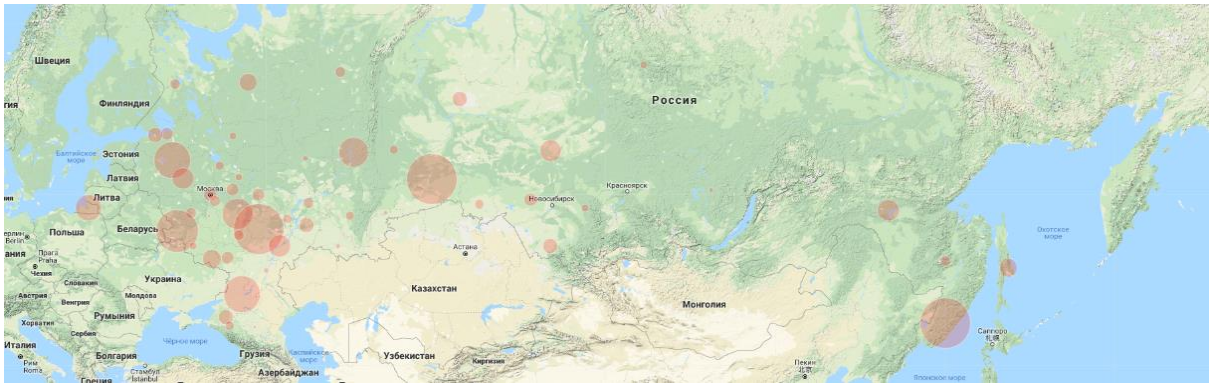


Рис. 4. Распределение средних баллов по РФ

Очевидно, что не во всех местах на Земле (и даже в одной стране) одинаковое состояние образовательной сферы. Онлайн-курсы больше не прикрепляют человека к определенному месту для прохождения занятий, однако это накладывает на них дополнительную ответственность: помогать адаптироваться людям, у которых меньше доступа к качественному образованию. Для решения этой проблемы в будущем рекомендательной системы может быть, например, предусмотрен модуль, добавляющий некоторое количество образовательных материалов для тех регионов, где постоянно наблюдается сравнительно низкое количество баллов, которые получают слушатели.

4.2 Маршруты слушателей

В статье “Mining theory-based patterns from Big data: Identifying self-regulated learning strategies in Massive Open Online Courses”, рассмотренной в разделе “Обзор литературы”, приводится исследование того, каким маршрутом слушатели проходят онлайн-курс. Поскольку в адаптивных курсах ключевым механизмом является выстраивание персонализированной траектории, то сначала необходимо исследовать то, какие траектории слушатели выбирают самостоятельно и какие результаты они получают.

Для этого исследования необходимы записи о событиях, связанных с просмотром видео и с решением тестов. Помимо этого, обязательно нужна структура курса. С её помощью определяется порядок элементов курса.

Каждый элемент курса edX имеет свой уникальный ID в виде строки из 32 символов. С его помощью происходит связь логов и структуры курса.

Определение порядка элемента в курсе выполняется последовательным перебором всех элементов курсов с увеличением счётчика. Этой процедуре соответствует следующий код (листинг 14).

```
for _, chapter := range courseStructure.Chapters {
    for _, sequential := range chapter.Sequentials {
        for _, vertical := range sequential.Verticals {
            for _, video := range vertical.Videos {
                result[video.URLName] = currentItemNumber
                currentItemNumber++
            }
            for _, problem := range vertical.Problems {
                result[problem.URLName] = currentItemNumber
                currentItemNumber++
            }
        }
```

```
    }
}
```

Листинг 14. Вычисление порядковых номеров элементов онлайн-курса по его структуре

Сперва необходимо получить список имен пользователей, для которых нужно построить траекторию прохождения. Для этого из Elasticsearch из индексов видео и контрольных заданий извлекаются уникальные значения поля username.

Далее в цикле по всем именам из Elasticsearch извлекаются все документы из индексов видео и контрольных заданий для данного пользователя. Документы уже отсортированы по дате, поэтому следующим, завершающим, этапом станет последовательный перебор. Его необходимо осуществлять вместе со счётчиком, который будет увеличиваться каждый раз, когда у события на новой итерации ID отличается от предыдущего, то есть человек перешёл от одного события к другому. Если увеличивать счётчик просто на каждой итерации, то получится уже другая, не лишенная смысла картина, однако в этом исследовании такой цели не стоит. Для нового ID находится порядок в курсе и это позволяет сгенерировать новую точку для графика: координатой x этой точки выступает номер элемента в курсе, а координатой y – счётчик, который увеличивается каждый раз, когда возникает новый ID. То есть график будет отображать по оси X номер элемента в курсе, а по оси Y – номер действия слушателя.

Для построения анализа маршрута прохождения курса использовалась библиотека построения графиков на языке JavaScript Plotly.

Графики имеют следующий вид (рисунок 5)

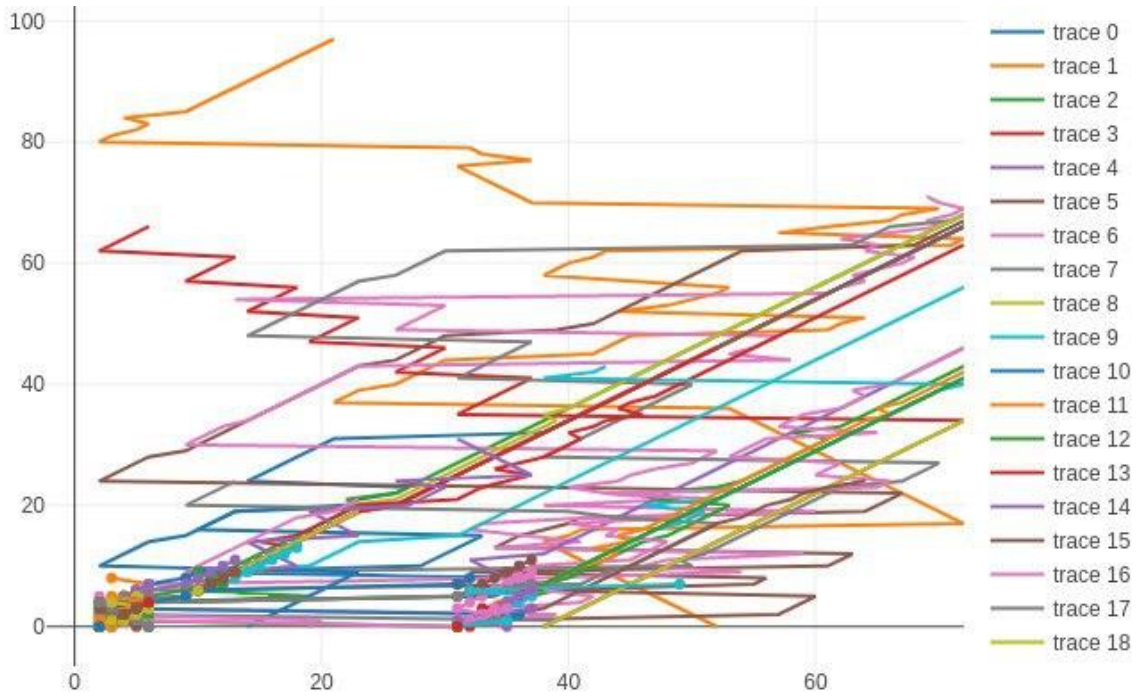


Рис. 5 График маршрута прохождения курса

Если кривая имеет линейный вид, то слушатель проходит курс в порядке следования материалов. На представленном графике можно видеть несколько кривых, которые соответствуют этому описанию. Остальные графики имеют переломы, то есть соответствующие кривым слушатели возвращались назад к предыдущим материалам. Их тоже можно разделить на несколько типов: хаотичное движение по курсу и длинные возвраты назад (этому соответствуют графики, где для двух соседних по Y координате точек координата X сильно отличается). Второй тип соответствует второму типу слушателей из статьи, а остальные – третьему типу. Для точного анализа типов таких слушателей необходимо рассмотреть точное количество переходов вперед, количество переходов назад и расстояния, которые проходят слушатели.

Такое исследование на данный момент выполнено в виде прототипа. Для этого использовался язык программирования Python и существующий сервис, который позволяет строить маршрут, описанный в этом параграфе. Для

достижения результата необходимо дополнительно получить оценку каждого пользователя и рассчитать количество переходов назад и вперед в рамках курса. Эксперимент проводился на данных с осенней сессии 2019 года по курсу “Философия”, на которой учились только студенты СПбГУ. Все они сдавали экзамен в присутствии преподавателя, поэтому в качестве измерения знаний человека использовались оценки за итоговый тест.

Оценки можно получить из оценочных листов, которые предусмотрены в edX. При помощи библиотеки pandas оценочный лист можно превратить в DataFrame, в рамках которого легко получить необходимые сведения. Для этого баллы за экзамен переводятся в 100-бальную систему (в оригинале они хранятся в рамках значений от 0 до установленного веса баллов за экзамен) (листинг 15).

```
grades['exam'] = grades[final_test_col_name] \
                 .apply(lambda x: 100*x/final_test_weight)
```

Листинг 15. Перевод значений столбца оценок в 100-бальную шкалу

После этого рассматривается маршрут каждого пользователя и отмечаются переходы назад (листинг 16).

```
for user in curves:
    total_moves = len(curves[user]['x']) - 1
    return_moves = 0
    point_number = 0
    prev_item_number = -1
    while point_number < len(curves[user]['x']):
        if prev_item_number > curves[user]['x'][point_number]
:
```

```
            return_moves+=1
        prev_item_number = curves[user]['x'][point_number]
        point_number+=1
    if total_moves != 0:
        count_returns[user] = 100*(return_moves/total_moves)
```

Листинг 16. Подсчет доли переходов назад для каждого слушателя

Создаётся новая колонка с процентом переходов назад (листинг 17).

```
grades_total['returns'] = grades_total['Username'].apply(
    lambda x: count_returns.get(x, 0)
)
```

Листинг 17. Добавление доли переходов назад в таблицу с данными о слушателях

По полученным данным строятся графики. Помимо графика процентов переходов строился также график количества действий пользователей (рисунок 6).

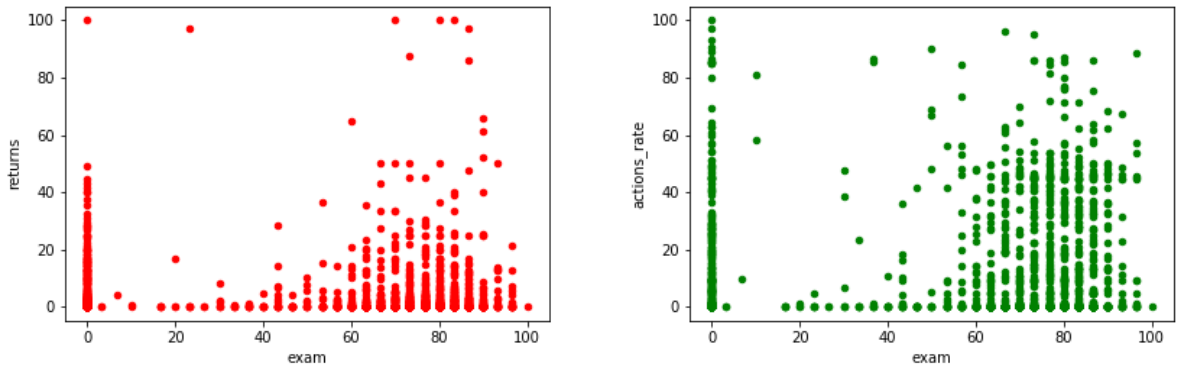


Рис. 6 Графики зависимости оценки за итоговый тест от количества переходов назад

На графике слева координата x каждой точки отражает оценку за экзамен (от 0 до 100); координата y – процентов переходов назад от общего количества переходов. На графике справа координата x каждой точки отражает оценку за

экзамен, а координата y – количество действий (было взято соотношение (количество действий конкретного пользователя/максимальное количество действий у одного пользователя в выборке).

Для полученных данных были посчитаны корреляции (таблица 1).

	Пирсона	Спирмена	Кендалла
% возвратов – оценка	0.03	0.05	0.04
Количество действий – оценка	0.05	0.08	0.06

Таблица 1. Корреляции между количеством возвратов и между количеством переходов назад и оценками слушателей

Можно заметить, что корреляция переходов от количества переходов назад или от количества действий незначительная (близка к 0). При таких обстоятельствах в рекомендательной системе можно выстраивать траектории соответственно: если слушателю лучше сначала посмотреть на контрольный тест, а потом уже решить смотреть материалы, то не нужно мешать ему в этом, но при этом не нужно заставлять слушателей сначала смотреть тесты, а потом учиться.

4.3 Стратегия просмотра видео

Видеолекции составляют большую часть онлайн-курса. В каждом курсе от 6 до 12 часов лекций. Большинство материала курса обычно приходится на эти лекции. В связи с этим необходимо исследовать поведение слушателей во время просмотра видео. В этом исследовании будут рассматриваться переходы слушателей между фрагментами одного видео. Так, некоторые могут смотреть видео последовательно, не пропускать куски; другие могут смотреть видео “по диагонали”: посмотреть немного в начале видео, немного в середине и

немного в конце; остальные пересматривают фрагменты видео. При этом не обязательно такая стратегия сохраняется для всех видео в курсе. Возможно, что сама лекция устроена таким образом, что фрагменты необходимо смотреть несколько раз для лучшего понимания. В этом параграфе будет рассмотрено это поведение и будет проведен анализ того, как влияет поведение слушателя во время просмотра видео на его финальную оценку за курс.

Для того, чтобы узнать, какие фрагменты слушатели смотрели больше, а какие меньше, необходимо для каждого момента видео сопоставить количество просмотров этого фрагмента.

Просмотр видео можно представить в виде отрезков [начало просмотра, конец просмотра]. Таким образом, если слушатель начал смотреть видео с начала и, не прерываясь, досмотрел до конца, то точка “начало просмотра” будет равна нулю, а точка “конец просмотра” будет равна длине видео. Однако это самый простой сценарий просмотра видео. Если слушатель посмотрел видео целиком, а потом решил, что он не понял вторую половину видео, то он вернется и создаст ещё один отрезок с середины видео и до конца видео. Такие отрезки можно обнаружить при помощи событий просмотра видео. Всего было выделено 2 вида событий: запуск воспроизведения видео и остановка. Эти точки обозначают все границы отрезков для этого видео (старт соответствует началам отрезков, а паузы – концам).

Алгоритм для решения этой задачи следующий:

1) Отсортировать все границы отрезков с сохранением их статуса (начало отрезка или конец отрезка);
2) Инициализировать счётчик количества активных просмотров нулем;
3) В цикле перебирать все границы отрезков и
    a) Увеличивать счётчик, если граница является началом отрезка;
    b) Уменьшать счётчик, если граница является концом отрезка;
    c) Если текущая точка отличается от предыдущей, сохранить пару (граница отрезка, значение счётчика).

4) После завершения цикла сохранить пару (последняя граница отрезка, значение счетчика).

Алгоритм корректен. Его работа заключается в подсчёте количества воспроизведений для каждого момента видео (обозначим момент времени буквой t). Количеством воспроизведений момента является количество интервалов, показывающих начало и конец воспроизведения слушателем. Доказательство корректности сводится к доказательству того, что цикл на шаге 3 алгоритма решает задачу, и что он корректен. Цикл находит все пары точек $(x;\ y)$ такие, что $x \in [0; T]$ (где T – продолжительность видео) – момент видео, а y – количество воспроизведений этого момента. При этом эти точки можно использовать для поиска количества воспроизведений любого момента $t \in [0; T]$, так как оно равно значению y точки с максимальным x таким, что $x \leq t$. Таким образом, цикл действительно решает задачу, так как он находит количество воспроизведений для каждого момента видео t. Теперь необходимо доказать его корректность. Для этого необходимо рассмотреть инварианту цикла. Обозначения:

- A – множество событий для данного видео отсортированных по моменту видео t. Его размер равен n;
- X – множество точек $(x;\ y)$, которые необходимо найти;
- c – счётчик, значения которого используются для вычисления значений координат y точек множества X.

Инвариантой цикла является следующее утверждение: в начале каждой итерации $i \in [1, n]$ сохранены все точки, обозначающие изменение количества просмотров, для моментов времени на луче $[0, t)$. Докажем её корректность:

- На этапе инициализации цикла это верно, так как никаких точек не должно быть сохранено;
- На этапе поддержки цикла сохранено столько точек, сколько уникальных моментов времени t встретилось среди рассмотренных

событий A. Сохраненные точки позволяют узнать точное количество воспроизведений момента t, так как:

- Для всех отрезков с началом и концом меньшим или равным t значение счетчика увеличилось и уменьшилось, то есть не изменилось;
- Для всех отрезков с началом меньшим или равным t и концом большим t значение счетчика было увеличено и не уменьшалось, то есть значение счетчика равно количеству отрезков этого вида;
- Для всех отрезков с началом и концом большими, чем t, значение счетчика еще не менялось.

Что означает, что каждая новая точка обозначает количество интервалов, содержащих момент t;

- При завершении цикла записывается последнее значение счетчика и момент времени последнего события, чтобы учесть последнюю точку.

Реализация произведена при помощи Elasticsearch, который позволяет получить все события определенного видео, отсортированные по возрастанию момента в видео, и при помощи реализации алгоритма на языке Go. Затем точки возвращаются во front-end приложение, где по ним строится график при помощи библиотеки Plotly (рисунок 7).

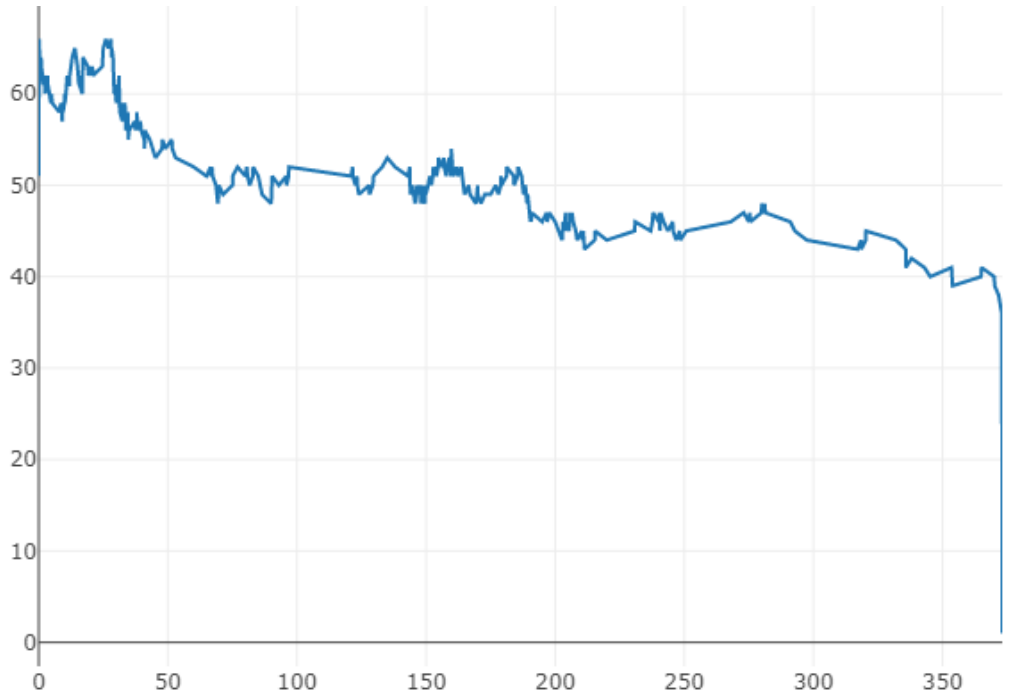


Рис. 7 График просмотра видео

На графике можно увидеть количество просмотров каждой секунды видео. Из таких графиков можно выделить две особенности видеоматериалов: моменты, которые пропускаются и моменты, которые необходимо пересматривать несколько раз. Неравномерность графика может говорить о присутствии одного из этих факторов. Определение того, какой именно фактор имеет место быть, осуществляется путём замера количества уникальных слушателей, которые смотрели выбранный видеоурок, что нетрудно посчитать одним запросом к Elasticsearch (листинг 18).

```
GET video_event_description/_search
{
  “query”: {
    “term”: {
      “video_id”: {
```

```
        "value": "d585cf06eb8f49ca8ddaf945d272b887"
      }
    }
  },
  "aggs" : {
      "unique_users" : {
          "cardinality": { "field" : "username"}
      }
  }
}
```

Листинг 18. Запрос для получения пользовательских имён, посмотревших видео по его ID

Такое исследование может помочь создателям курсов. Если они видят, что какой-то фрагмент видео часто пересматривается, то вероятно, с ним что-то не так. Возможно, там допущена ошибка и слушатели пытаются понять, что же пытался сказать автор – такая проблема легко решается добавлением примечания в видео в этом фрагменте, но для этого о существовании такого фрагмента необходимо знать заранее. Даже если слушатели рано или поздно скажут об этой ошибке на форумах обсуждений, существует ненулевая вероятность, что их поведение предупредит об ошибке раньше, что позволит не лишиться слушателей и не получить низкие рейтинги.

В контексте адаптивного обучения такая информация тоже может быть полезной. Ожидается, что есть зависимость между тем, как слушатели смотрят видео, и оценкой, которую они в итоге получают. То есть слушатели, которые смотрят видео больше одного раза, получают лучшую (или наоборот худшую) оценку. Рассмотрим глобально, как слушатели смотрят видео и какие оценки они получают (рисунок 8).

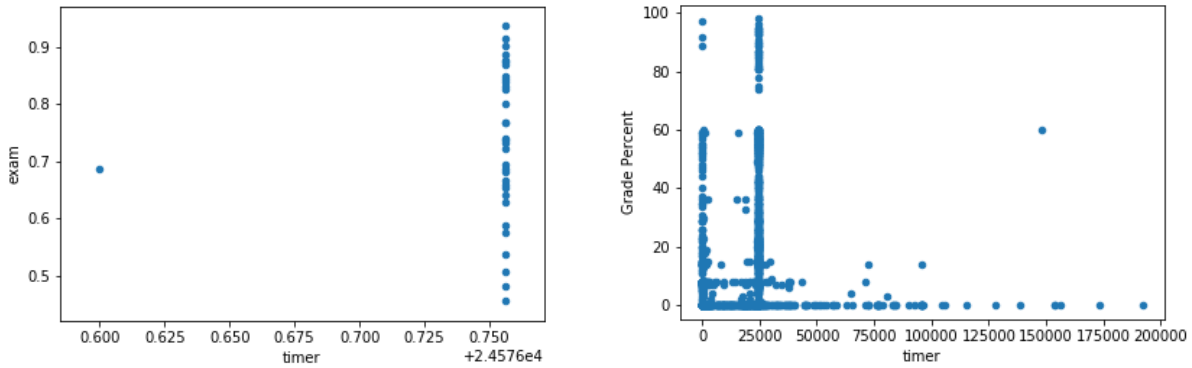


Рис. 8 Зависимость итоговой оценки от методики просмотра видео

На графике слева изображены только слушатели, которые сдавали экзамен и смотрели видео, а справа – все слушатели. По оси X измеряется количество секунд, которые слушатель потратил на просмотр видео курса, а по оси Y – оценка (за экзамен на левом графике и общая на правом графике).

Можно отметить несколько выводов из рассмотренного:

1) Нормальное поведение при просмотре видео: смотреть его от начала до конца и не перематывать. Это видно по столбцу около отметки 25.000 секунд. Это число характеризует общую продолжительность видео на курсе;
2) Есть вольные слушатели – люди, которые только слушают лекции, но не решают задания;
3) Все слушатели, которые сдавали экзамен по этому курсу, посмотрели каждое видео один раз. Один слушатель перематывал видео, но всё равно успешно сдал экзамен (70 баллов соответствует оценке удовлетворительно);
4) Существуют слушатели, которые не смотрели видео вообще, но при этом успешно сдали экзамен. Скорее всего это связано с тем, что слушатели учились на предыдущей сессии, но не успели сдать экзамен (так что они не смотрели видео снова). Возможно и то, что слушатели знали предмет до прохождения курса и им необходимо было только получить сертификат.

Таким образом, поставленная гипотеза не подтвердилась в рамках этого курса. Были взяты ещё несколько курсов и наблюдалась такая же тенденция. Возможно, это связано с небольшими выборками – на каждой сессии всего лишь 10-20 человек полностью завершают курс. На сегодняшний день выросло количество слушателей (особенно в рамках пандемии и перехода на дистанционное обучение), но эти данные ещё не собраны до конца (на момент написания работы сессии ещё не завершились). Созданные методики предстоит применять и к другим данным в будущем для поиска зависимости. Если такой зависимости не будет обнаружено, то такой анализ всё равно останется полезным инструментом для разработчиков онлайн-курсов.

4.4 Сравнение слушателей на общих сессиях и студентов

Слушатели курсов ЦРЭОР СПбГУ сильно отличаются. Одна группа – “внешние слушатели”. Любой желающий может зайти на образовательные платформы и начать изучение курсов СПбГУ. Они могут учиться так долго, как они захотят (даже если они не успеют пройти курс за одну сессию, они могут пройти курс на следующей сессии). Единственный момент, который может теоретически обязать человек проходить курс – покупка сертификата слушателем. В таком случае он должен пройти экзамен и перейти через проходной балл, что потребует от человек приложить усилия.

Вторая группа – студенты ВУЗов. Самая большая группа слушателей такого типа – студенты СПбГУ, которые осваивают онлайн-курс в качестве дисциплины в университете. Это сильно меняет условия – все слушатели имеют меньший разброс по возрасту; они учатся в одном университете; мотивация пройти курс крайне высокая; у слушателей меньше преград, чтобы общаться между друг другом. Можно привести ещё множество факторов, отличающих эти группы друг от друга. Рассмотрим, какие отличия можно увидеть по проведенным исследованиям.

Географическое распределение

Это сравнение лишено смысла, поскольку все учащиеся СПбГУ находятся в городе Санкт-Петербург.

Маршруты слушателей

Рассмотрим отношение количества переходов назад по материалам курсов к общему количеству переходов и оценок, которые получили учащиеся. Для этого будет рассмотрен курс по философии. В параграфе, где приведено описание этого исследования, брались данные о студенческой сессии, так что сперва необходимо рассмотреть данные об общей сессии (таблица 2).

	Пирсона	Спирмена	Кендалла
% возвратов – оценка	-0.15	-0.21	-0.14
Количество действий – оценка	0.77	0.87	0.70

Таблица 2. Корреляции между количеством возвратов и между количеством переходов назад и оценками слушателей-студентов университета

Можно заметить два отличия от сессии со студентами: присутствует высокая корреляция оценки слушателя от количества действий на курсе, что может свидетельствовать о том, что хорошую оценку на курсе получают только те, кто прикладывает усилия; корреляция между возвратами назад всё ещё низкая, но она поменяла свой знак. Второе наблюдение может стать очередным доказательством мнения исследователей из статьи “Mining theory-based patterns from Big data: Identifying self-regulated learning strategies in Massive Open Online Courses”, рассмотренной в разделе “Обзор литературы”, так как возвраты назад присущи не только студентам, которые методично

заглядывают в конец курса, а потом возвращаются к текущему моменту, но и тем, кто не имеет стратегии и не достигает успехов в изучении курса.

То, что слушатели на общих сессиях учатся на онлайн-курсе намного больше, может говорить о том, что студенты списывают на экзамене. При этом у них действительно гораздо больше возможностей прибегать к жульничеству. Эти данные могут стать основанием для пересмотра текущих методов проведения промежуточных аттестаций.

Методики просмотра видео

В процессе сравнения данных по студенческим сессиям для студентов с сессиями для всех слушателей было сделано следующее наблюдение: видео курса смотрела меньшая часть учащихся. Таким образом, высокие баллы могли получить как те, кто смотрел все видео курса, так и те, кто не смотрел видео вообще. Корреляция между количеством просмотренных секунд и полученной оценкой не наблюдается. Те факторы, которые были перечислены в начале раздела 4.4, могли способствовать этому. Например, возможность находиться близко друг к другу физически могла привести к тому, что многие смотрели видеолекции вместе и событие фиксировалось только для одного пользователя. Но основным выводом из такого распределения данных может стать то, что на экзаменах имели место случаи жульничества. На этой сессии студенты сдавали экзамен очно в присутствии преподавателя, задача которого – следить за тем, как проходит экзамен.

Выводы из этой ситуации уже были сделаны: следить за группами студентов, которые сдают экзамены, достаточно сложно, но при этом на общих сессиях большинство слушателей, которые сдавали экзамен, смотрели все видео курса. Контроль прохождения экзамена этих слушателей обеспечивают системы прокторинга и прокторы. Возможностей для списывания в таких обстоятельствах меньше. Сейчас такие системы внедряются для проведения промежуточных аттестаций по онлайн-курсам и для студентов.

4.5 Выводы по главе 4

В этой главе были описаны методы, которыми удалось дополнить популярные виды анализа данных онлайн-курсов. Помимо сбора общих сведений об оценках и активности слушателей, были рассмотрены детализированные данные о стратегиях просмотра курсов и при просмотра видео.

Анализ географии и его связи с оценкой слушателей может стать одним из алгоритмов для принятия решения о дополнительной нагрузке на студента. Если какой-то регион продолжает показывать результаты ниже средних или в этом регионе отмечаются аномально низкие оценками, то возможно рассмотреть повышение количества материалов для этого региона, чтобы слушатели могли восполнить свою образовательную потребность.

На данный момент не было найдено корреляции между количеством возвращений к предыдущим материалам курса и оценкой слушателя. Если отсутствие этой зависимости подтвердится, то в адаптивной системе необходимо не закреплять порядок прохождения курса, а предлагать слушателю самому решать, какие материалы он будет изучать. Судя по статье “Mining theory-based patterns from Big data: Identifying self-regulated learning strategies in Massive Open Online Courses” слушатели могут возвращаться назад методично, когда они уходят посмотреть на то, какие задания им предстоит решить по итогам обучения на очередной неделе курса. В таком случае адаптивная система могла бы предлагать посмотреть на задания в начале недели, чтобы студент имел цель для обучения на новой неделе. Возможно, именно в рамках работы адаптивной системы удастся выявить зависимость между количеством переходов назад и оценкой слушателя.

Исследование методики просмотра лекций курса позволило обнаружить то, что большинство смотрит видео подряд, не пропускает никакие фрагменты и при этом не увлекается пересмотром видео. Большинство слушателей, которые сдают экзамен, смотрят все видео курса подряд. При этом, возможно, интерес к поведению во время просмотра видео, должен быть сосредоточен на тех, кто покинул курс. Возможно, если слушатель пытается понять материал с

нескольких попыток, это провоцирует его поменять курс. Это информация может пригодиться авторам курсов для своевременного обнаружения проблем на курсе.

Сравнение результатов анализа слушателей на общих сессиях и слушателей-студентов СПбГУ отчетливо показывает сниженную активность студентов во время обучения на курсах. При этом многие другие признаки говорят о том, что студенты не редко прибегают к списыванию во время прохождения экзаменов по онлайн-курсам. Эти данные могут только подтвердить такую гипотезу. Решение такой проблемы может заключаться в использовании систем прокторинга, которые значительно снижают вероятность жульничества. К сожалению, на момент написания этой работы только начинается сессия, в которую используются такие системы и для проведения экзаменов у студентов. После сбора этих данных они будут проанализированы при помощи созданных методик для сравнения не только общих сессий и сессий студентов, но и сессий студентов с использованием прокторинга и без.

Заключение

По итогам работы удалось создать прототип системы, которая позволяет анализировать данные и постоянно отправлять эти данные в системы, которые от них зависят. Прототип используется для повышения доступности данных онлайн-курсов для принятия решений о развитии ЦРЭОР и его курсов. Помимо этого, система будет интегрирована в разрабатываемую на данный момент систему адаптивных курсов.

Система состоит из трёх частей – сбор данных, обработка данных и их анализ. В качестве первого слоя взято готовое решение – Filebeat. Слои анализа данных и обработки данных являются набором сервисов, которые были разработаны в рамках этой работы. Архитектура системы выделяется возможностью к масштабированию – с ростом количества передаваемых для анализа данных можно без существенных затрат увеличивать мощность серверов путем дублирования сервисов на новые сервера. Разработка новых видов сервисов тоже не является трудной задачей – им необходимо подключиться к Kafka и Elasticsearch при помощи передачи конфигурации в существующие интерфейсы для этих компонент и использовать их для получения и хранения данных соответственно. Аналогичным образом масштабируются сервисы анализа данных.

Были изучены существующие методики анализа: готовые решения и теоретические исследования в научных статьях. Из них стало понятно, что текущий анализ показывает общие характеристики – успеваемость слушателей, их распределение. При этом малое внимание уделяется зависимостям между различными данными. С этой целью в главе 4 были представлены методы анализа связи различных характеристик. Было рассмотрено влияние географического местонахождения слушателей курсов, методика продвижения по курсу и методика просмотра видео. Для географического распределения даже в рамках страны удалось заметить различие между успеваемостью в разных регионах. Методика прохождения курса не имеет сильного влияния на успеваемость слушателей, но при этом

существуют разные подходы к построению маршрута прохождения курса и все они могут привести к получению положительных оценок, что несомненно стоит использовать при разработке адаптивной системы онлайн-курсов. Методика просмотра видео в большинстве случаев была одинаковой – слушатели смотрят видео от начала до конца, не пересматривая и не перематывая фрагменты. Это говорит о том, что в действительности на сегодня нет основания полагать, что изменение этого поведения является нормой. Таким образом, если слушателям не нужно смотреть видео целиком или если они их пересматривают несколько раз, необходимо обращать внимание на этот курс или этот контингент.

Важным моментом исследования стало сравнение данных о внешних слушателях и студентов СПбГУ. В этих данных обнаружено сильное различие в их поведении во время просмотра курсов. Такое изменение поведения подтверждает возможные случаи списывания во время проведения промежуточных аттестаций.

Развитие работы будет идти по нескольким направлениям:

1) Расширение слоя обработки данных для использования данных других платформ;
2) Расширение слоя анализа данных для проведения новых исследований. Необходимо реализовать составление аналитических срезов и показателей, рассмотренных в главе 1;
3) Создание методов для построения персонализированных траекторий для слушателей онлайн-курсов по данным об их активности.

Источники

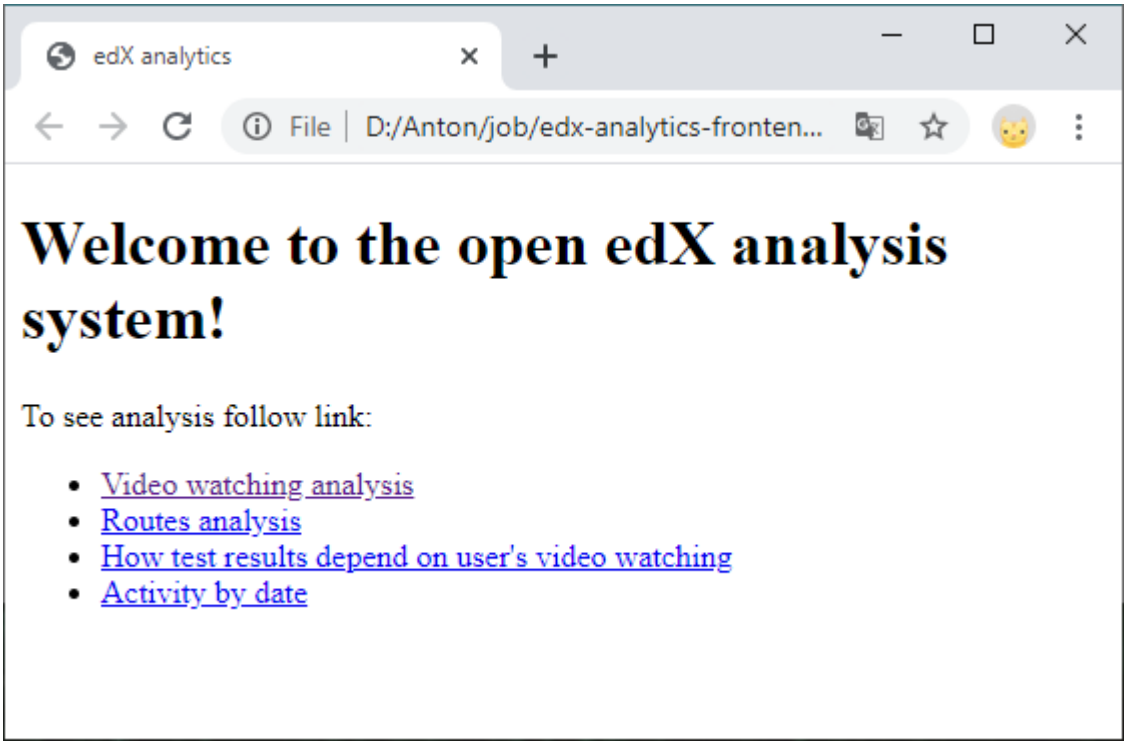
1. Understanding Item Analyses | Office of Educational Assessment // University of Washington Home URL: https://www.washington.edu/assessment/scanning-scoring/scoring/reports/item-analysis (дата обращения: 02.06.2020).
2. Matlock-Hetzel S. Basic Concepts in Item and Test Analysis // Annual Meeting of the Southwest Educational Research Association – 1997.
3. Guo P. J., Kim J., Rubin R. How video production affects student engagement: an empirical study of MOOC videos // Proceedings of the first ACM conference on Learning@ scale conference. – ACM, 2014. – С. 41-50.
4. Optimal Video Length for Student Engagement // edX URL: https://blog.edx.org/optimal-video-length-student-engagement (дата обращения: 02.06.2020).
5. АДАПТИВНОСТЬ: С ЧЕГО НАЧАТЬ И НУЖНО ЛИ? // EDUTAINME URL: http://www.edutainme.ru/post/adaptive-4/ (дата обращения: 02.06.2020).
6. Breslow, L., Pritchard, D. E., DeBoer, J., Stump, G. S., Ho, A. D., & Seaton, D. T. (2013). Studying learning in the worldwide classroom research into edX's first MOOC. Research & Practice in Assessment, 8, 13-25.
7. Maldonado-Mahauad, J., Pérez-Sanagustín, M., Kizilcec, R. F., Morales, N., & Munoz-Gama, J. (2018). Mining theory-based patterns from Big data: Identifying self-regulated learning strategies in Massive Open Online Courses. Computers in Human Behavior, 80, 179-196.
8. Открытое образование URL: https://openedu.ru (дата обращения: 02.06.2020).
9. Coursera URL: https://coursera.org (дата обращения 02.06.2020).
10. Student Events // edX Research Guide URL: https://edx.readthedocs.io/projects/devdata/en/stable/internal_data_formats/tracking_logs.html#student-events (дата обращения: 02.06.2020).

11. Overview of EdX Insights // Using edx Insights URL: https://open-edx-insights.readthedocs.io/en/latest/Overview.html (дата обращения: 02.06.2020).
12. What is the ELK Stack? // elastic URL: https://www.elastic.co/what-is/elk-stack (дата обращения: 02.06.2020).
13. Lightweight shipper for logs // elastic URL: https://www.elastic.co/beats/filebeat (дата обращения: 02.06.2020).
14. Apache Kafka URL: https://kafka.apache.org/ (дата обращения: 02.06.2020).
15. Overview | Maps JavaScript API | Google Developers // Google Maps Platform URL: https://developers.google.com/maps/documentation/javascript/tutorial (дата обращения: 02.06.2020).
16. Visualizing Data: Mapping Earthquakes | Maps JavaScript API // Google Maps Platform URL: https://developers.google.com/maps/documentation/javascript/earthquakes (дата обращения: 02.06.2020).
17. Developer Guide | Geocoding API | Google Developers // Google Maps Platform URL: https://developers.google.com/maps/documentation/geocoding/intro (дата обращения: 02.06.2020).
18. He, Jiazhen, et al. "MOOCs meet measurement theory: a topic-modelling approach." Thirtieth AAAI Conference on Artificial Intelligence. 2016.
19. Shi, Conglei, et al. "VisMOOC: Visualizing video clickstream data from massive open online courses." 2015 IEEE Pacific visualization symposium (PacificVis). IEEE, 2015.
20. Орлов А. С., Севрюков С. Ю. Разработка прототипа системы управления Центром развития электронных образовательных ресурсов СПбГУ // Смирнов Н. В. Процессы управления и устойчивость. 2018: СПбГУ, 2018. С. 333-337.

21. Севрюков С. Ю., Сорокина С. О., Орлов А. С. Оценка возможностей современных платформ онлайн образования в контексте анализа данных о поведении учащихся и их адаптивного обучения // Смирнов Н. В. Процессы управления и устойчивость. 2019: СПбГУ, 2019. С. 357 - 363.
22. Rethinking Higher Ed: A Case for Adaptive Learning URL: https://www.forbes.com/sites/ccap/2014/10/22/rethinking-higher-ed-a-case-for-adaptive-learning (дата обращения: 02.06.2020).
23. Адаптивные курсы URL: https://support.stepik.org/hc/ru/articles/360002316314 (дата обращения: 02.06.2020).
24. Адаптивное обучение и персонализация URL: https://edutechclub.sberbank-school.ru/node/5 (дата обращения: 02.06.2020).
25. Mining. Through educational data, enhancing teaching and learning through educational data mining and learning analytics: An issue brief // Proceedings of conference on advanced technology for education. 2012. 64.
26. Zhang X., Zhong S., et al. Entertainment for Education. Changchun: Springer Science & Business Media, 2010. 135.
27. Opening the Black Box of Adaptivity // Educause URL: https://er.educause.edu/blogs/2017/6/opening-the-black-box-of-adaptivity (дата обращения: 02.06.2020).

Приложения

Приложение 1 Главная страница front-end приложения анализа данных



Приложение 2 Страница с графиком активности во время просмотра видео слушателями