Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки 09.04.04 – Программная инженерия
Направленность (профиль) образовательной программы Управление разработкой программного обеспечения

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
«____» _____2021 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

на тему: Обработка больших данных с целью учета количества обучающихся

Исполнитель
студент группы 957-ом _____ А.А. Ковшик
(подпись, дата)

Руководитель
доцент, канд. техн. наук _____ С.Г. Самохвалова
(подпись, дата)

Руководитель научного
содержания программы
магистратуры _____ И.Е. Еремин
(подпись, дата)

Нормоконтроль
инженер кафедры _____ В.Н. Адаменко
(подпись, дата)

Рецензент _____ С.В. Щербаков
(подпись, дата)

Благовещенск 2021

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ
Зав. кафедрой
_____ А.В. Бушманов
«____» _____

З А Д А Н И Е

К выпускной квалификационной работе студента Ковшик Алексея Анатольевича

1. Тема дипломной работы: Обработка больших данных с целью учета количества обучающихся.

(утверждена приказом от 15.04.2019 №847-уч)

2. Срок сдачи студентом законченной работы: 25.06.2019 г.

3. Исходные данные к выпускной квалификационной работе: отчет по практической подготовке, нормативная документация, специальная литература.

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов): аргументирование необходимости разработки, анализ области больших данных, выбор метода обработки больших данных, выбор подходящего фреймворка, выбор стеммеров, разработка алгоритма преобразования информации, проектирование структуры программы, разработка и тестирование программы, оценка её надежности.

6. Консультанты по дипломной работе: ????

7. Дата выдачи задания: 15.04.2019 г.

Руководитель дипломной работы: Самохвалова С.Г, декан факультета математики и информатики, доцент, кандидат технических наук.

Задание принял к исполнению: _____

РЕФЕРАТ

Магистерская работа содержит 89 с., 61 рисунок, 43 источника.

ПРЕОБРАЗОВАНИЕ, ИНФОРМАЦИЯ, САЙТ, МОДЕЛЬ, ПРОЕКТ, ПРОГРАММНЫЙ ПРОДУКТ, ОБРАЗОВАНИЕ, НАПРАВЛЕНИЕ ПОДГОТОВКИ, ОБУЧАЮЩИЙСЯ, ТОКЕНИЗАЦИЯ, ОБРАБОТКА, ТАБЛИЦА

В работе выполнен анализ области больших данных, с целью обработки и получения текстовой информации с сайтов образовательных организаций.

Цель магистерской диссертации: разработка программного продукта для учета количества обучающихся на основе выбранного метода обработки больших данных.

Работа включает в себя:

- аргументирование необходимости разработки;
- анализ области больших данных;
- выбор метода обработки больших данных;
- выбор походящего фреймворка и стеммеров;
- выбор стеммеров;
- разработка алгоритма преобразования информации;
- проектирование структуры программы;
- разработка и тестирование программы, оценка её надежности.

Результатом магистерской работы является разработанный программный продукт, позволяющий осуществить учет количества обучающихся по определённым специальностям/направлениям подготовки и визуализировать эту информацию. Данная программа поможет пользователям выделить наиболее популярные направления подготовки/специальности. Образовательным организациям программа позволит прогнозировать рост популярности определенного направлении подготовки/специальности и при необходимости вести статистический учет.

СОДЕРЖАНИЕ

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей магистерской диссертации использованы ссылки на настоящие стандарты нормативные документы:

ГОСТ 2.103-68 ЕСКД Стадии разработки;

ГОСТ 2.104-68 ЕСКД Основные надписи;

ГОСТ 2.105-95 ЕСКД Общие требования к текстовым документам;

ГОСТ 2.111–2013 ЕСКД. Нормоконтроль;

ГОСТ 7.1-2003 СИБИД. Библиографическая запись. Библиографическое описание. Общие требования и правила составления;

ГОСТ 34.601-90. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания;

ГОСТ 34.603-92 Информационная технология. Виды испытаний автоматизированных систем;

ГОСТ Р ИСО/МЭК 12207-2010 Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программных средств;

ГОСТ 15971-90 Системы обработки информации. Термины и определения;

ГОСТ Р ИСО/МЭК 20546-2019 Информационные технологии. Большие данные. Обзор и словарь;

ГОСТ 20886-85 Организация данных в системах обработки данных. Термины и определения;

ГОСТ Р 52872-2019 Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме;

ГОСТ Р 57193-2016 Системная и программная инженерия. Процессы жизненного цикла систем;

ГОСТ Р МЭК 60950-23-2011 Оборудование информационных технологий. Требования безопасности. Часть 23. Оборудование для хранения больших объемов данных.

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

Парсинг – процесс автоматического сбора информации с необходмых источников, по определенным критериям;

ПО - программа или множество программ, используемых для управления компьютером (ISO/IEC 26514:2008);

Стеммер – алгоритм, при помощи которого выполняется стемминг;

Стемминг – определение основы для необходимого слова.

Токенизация – это процесс разделения письменного языка на предложения-компоненты;

Фре́ймворк – платформа, которая определяет структуру программной системы. На её базе формируется разрабатываемый продукт;

Data mining – множество методов поиска и анализа данных с целью выявления в них неизвестных знаний;

Dataframe – табличная структура данных. Ее задача — позволить использовать многомерные структуры данных. Dataframe состоит из упорядоченной коллекции колонок, каждая из которых содержит значение разных типов;

MapReduce – наиболее популярная модель для распределенной обработки информации.

Pandas – библиотека написанная на Python. Она используется для анализа и обработки данных;

Python – универсальный язык программирования общего назначения, он ориентирован на структурированность, и читаемость кода.

ВВЕДЕНИЕ

Объемы информации в настоящее время возрастают в быстром темпе, тем самым актуализируя проблему обработки таких данных. Оперативная обработка этих данных одно из перспективных направлений во множестве сфер деятельности, в том числе и в сфере образования.

Наибольшее количество информации в глобальной сети и не только, храниться в виде текстовой информации. Преобразовать эти данные проблематично из-за необходимости знаний лингвистики, а готовые решения для одной предметной области плохо реализуемы в других областях.

Цель магистерской работы: Разработка программного продукта для учета количества обучающихся на основе выбранного метода обработки больших данных.

Выполнение работы состоит из следующих этапов:

- анализ области больших данных и аргументация разработки;
- выбор метода обработки больших данных;
- выбор походящего фреймворка и стеммеров;
- разработка алгоритма преобразования информации;
- проектирование структуры программы, разработка и тестирование программы, оценка её надежности.

Научная новизна работы состоит в применении выбранных методов и средств, для обработки текстовой информации, в сфере образования.

Практическая значимость определяется возможностью программы помочь выявить востребованные на текущий момент специальности/направления подготовки, вести статистику на их основе и принимать решения о перспективе популярности того или иного направления подготовки/специальности.

Результатом магистерской работы является разработанный программный продукт, выполняющий обработку больших данных и предоставляющий пользователю необходимую информацию.

1 ХАРАКТЕРИСТИКА БОЛЬШИХ ДАННЫХ В СФЕРЕ ОБРАЗОВАНИЯ

1.1 Большие данные в различных сферах деятельности

В настоящее время цифровые технологии присутствуют практически во всех областях жизни любого человека, тем самым увеличивая объемы вырабатываемой информации. С хранением и обработкой настолько большого количества данных уже нельзя справиться при помощи стандартных методов, именно поэтому в 2013 оксфордский английский словарь ввел определение «Big data», которое можно сформулировать следующим образом «Данные очень большого размера, как правило, в том смысле, что представляют серьезные трудности в материально-техническом обеспечении по манипуляциям и управлению ими; (также) направление вычислений с использованием такого типа данных». В России наибольшую популярность получило следующее определение: большие данные – набор информации, по объему превосходящей жесткий диск одного персонального устройства и не поддающейся обработке классическими инструментами, применяемыми для меньших объемов.

Определив, что считается большими данными, нужно понять ценность этого механизма для определенных сфер деятельности и мира в общем. Компания IDC Digital Universe осуществила исследования, согласно которым в 2020 году в мире создано 40 зеттабайт информации, а к 2025 году это количество возрастет до 160 зеттабайт (рисунок 1) [1].

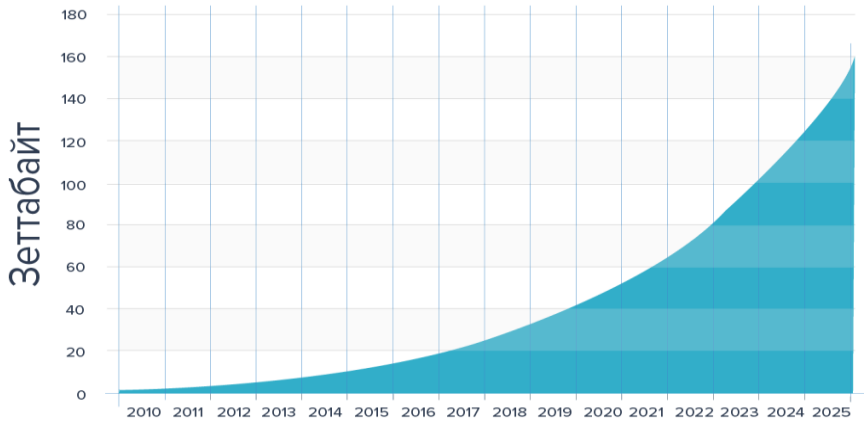


Рисунок 1 – Размер информации в мире по данным IDC Digital Universe

Большие данные вызвали интерес в таких сферах как:

- биология и медицина, поскольку происходит накопление информации в области нейронаук из-за большого количества результатов, полученных с использованием аналитических методов с высоким разрешением, а также информации, выработанной попытках исследования работы мозга, и т.д;

- в геодезии и фотограмметрии обработка больших данных происходит при исследовании снимков почвы и аэрокосмических снимков;

- одно из наиболее заинтересованных направлений – журналистика, поскольку огромное количество источников информации в сети интернет дает уникальные возможности для развития современной журналистики;

- в статистических исследованиях главным преимуществом больших данных является своевременное получение объемных массивов информации с наименьшими финансовыми и временными затратами.

Компанией «Геолайн технологии» был осуществлен опрос, результаты которого представлены на рисунке 2:

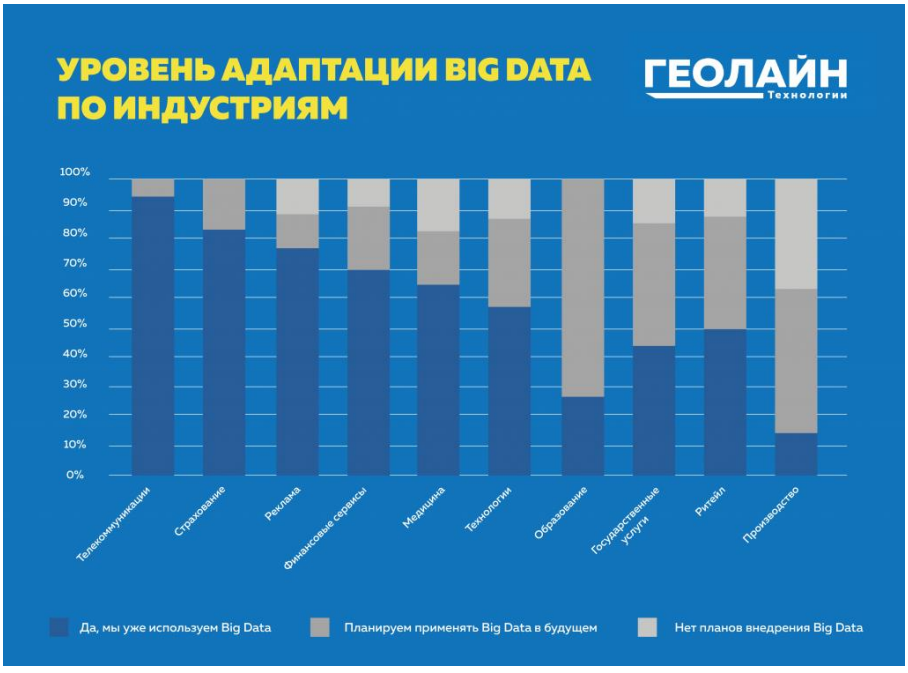


Рисунок 2 – Предполагаемый прогноз от внедрения больших данных в различных сферах деятельности

Из рисунка 2 видно, что многие сферы деятельности уже используют большие данные, а некоторые только планируют их внедрять. Рассматриваемая нами

область образования только начала использовать эту технологию, но планирует максимально адаптироваться к её использованию в будущем.

Большие данные используются для решения следующих задач [2]:

- совершенствование группировки клиентов;
- улучшение продаж;
- прогноз ситуации на рынке;
- развитие выбора товаров и услуг;
- разработка наиболее оптимальных управленческих решений;
- повышение производительности труда;
- эффективная логистика.

В данной работе осуществляется анализ данных с сайтов образовательных организаций с целью учета количества обучающихся по направлениям подготовки/специальностями. Это позволит собрать статистические данные, которые помогут спрогнозировать популярность того или иного направления подготовки/специальности, и отслеживать её.

Чтобы определить методы обработки больших данных, необходимо определить тип данных, которые будут обрабатываться. В разных сферах деятельности превалируют свои типы данных, сопоставим на схеме области применения и степень использования определенного вида информации:

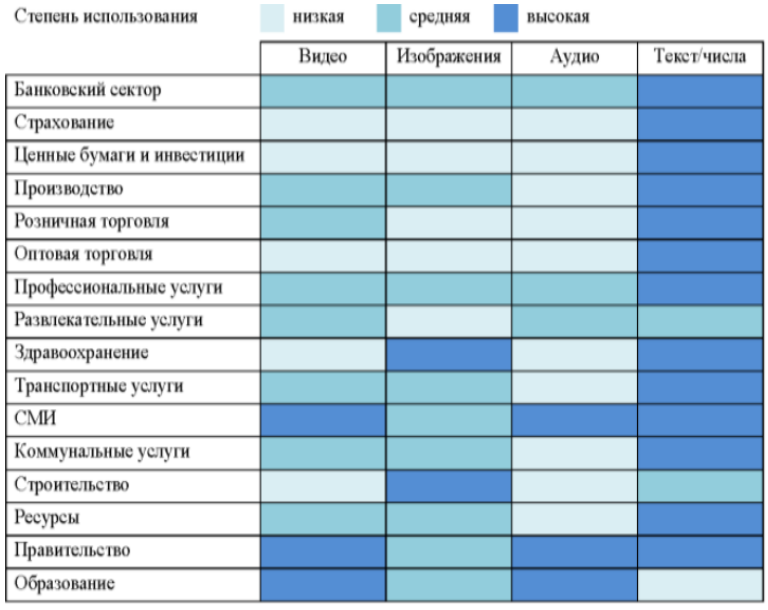


Рисунок 3 – Степень использования типов информации в разных сферах деятельности

Как видно из диаграммы (рисунок 3) наиболее распространенным видом информации практически для любой сферы являются текстовые и числовые данные соответственно их обработка имеет наиболее важное значение. Однако в сфере образования этот тип данных согласно диаграмме менее популярен, потому что в статистике учитывались только данные непосредственно используемые в образовательно процессе.

Данные подлежащие обработке в данной работе – текстовые, поскольку на сайтах образовательных организаций информация о количестве обучающихся выложена в табличном виде. Следовательно, для работы программы необходимо выбирать метод для обработки текстовых данных.

Компанией Economist Intelligence Unit были проведены исследования, для визуализации преимуществ от использования больших данных. Результаты исследования показаны на рисунке 4.

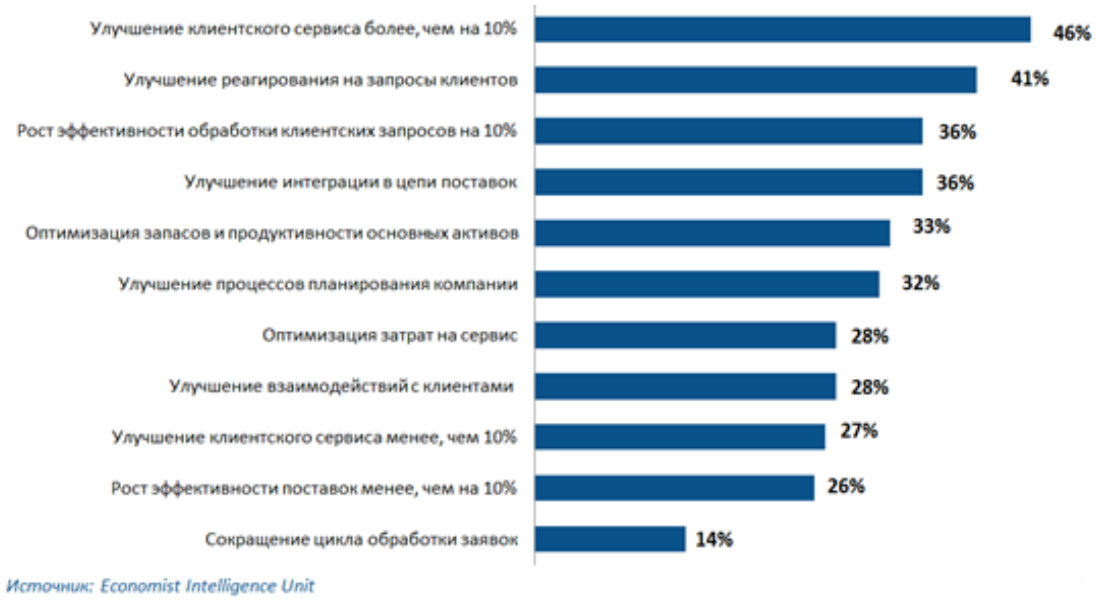


Рисунок 4 – Последствия от использования больших данных

Согласно результатам представленным на рисунке 4, использование больших данных вызывает только положительные последствия для организаций, которые их используют. Кроме того, ситуация в мире, приводит к необходимости работать удаленно с использованием мировой сети, следовательно, актуальность больших данных возрастает во всех сферах деятельности, и наиболее существенно в сфере образования.

Таким образом можно сделать вывод, что тема магистерской диссертации особенно актуально на текущий момент времени. Это доказывает позиция государства, которое в этом году решило разработать и внедрить суперсервис «Поступление в вуз онлайн».

1.2 Характеристики и принципы работы

Образовательные организации на протяжении многих лет сохраняют и используют множество данных, в своих системах. Эта информация включает в себя данные о: обучающихся, оценках, посещаемости, преподавателях, направлениях подготовки/специальностях, а также различный образовательный контент (текст, аудио, видео) и т.д. Эти данные необходимо постоянно обрабатывать, анализировать и хранить так, чтобы польза от них была максимальной [3].

Big data в образовании охватывает три аспекта (рисунок 5):

- объем – т.е данные о количестве обучающихся и об учебных заведениях, накопленные за многие года их существования. Такая информация, позволяет максимально эффективно организовать учебный процесс;

- скорость – т.е в зависимости от скорости с которой данные изменяются, увеличивается контроль процесса обучения и появляется возможность быстро реагировать на любые изменения;

- разнообразие – информация в образовательных организациях настолько разнообразна, что позволяет преподавателям использовать не только текст и фото в своей работе, но и видеоматериалы, 3D-модели и другие необычные типы данных [4].

Рисунок 5 – Три аспекта Big data в электронном обучении

Целесообразно рассмотреть возможность использования методов data mining, для достижения конечной цели исследования. Data mining в сфере электронного обучения используется для решения следующих задач:

а) *Классификация* – распределение действий или событий, по классам, которые заранее определены. В Data mining классификация основывается на использовании следующих моделей: деревья решений, нейронные сети, метод k-means.

К примерам задач классификации в процессе электронного образования можно отнести:

1) классификация электронных материалов;

2) классификация тестовых заданий.

б) *Регрессия*, используется для прогнозирования. Применение регрессии дает возможность разработать модель влияния параметров друг на друга, а также создать зависимость выходных данных от входных переменных.

в) *Кластеризация* – отбор объектов с максимально похожими признаками и свойствами, и добавление их в кластеры. Похожие объекты хранятся внутри одного кластера, а в разных кластерах – объекты должны обладать разными признаками.

г) *Анализ данных социальных сетей*, играет важную роль. В социальных сетях хранятся наиболее полные данные об обучающихся ведь в них они общаются, передают информацию друг другу. Часто в социальных сетях создаются группы для обмена информацией об учебном процессе. Общаясь в социальной сети, её пользователи находятся в комфортных для них условиях, поэтому информация, полученная этим способом, более полна и приближена к объективной. Из социальных сетей можно получить данные о хобби, планах, перемещениях, активности и тд [5].

Эти задачи могут использоваться и в других сферах деятельности, но в текущей работе они рассмотрены в сфере образования.

Графически описанные задачи data mining в электронном обучении изображены на рисунке 6.

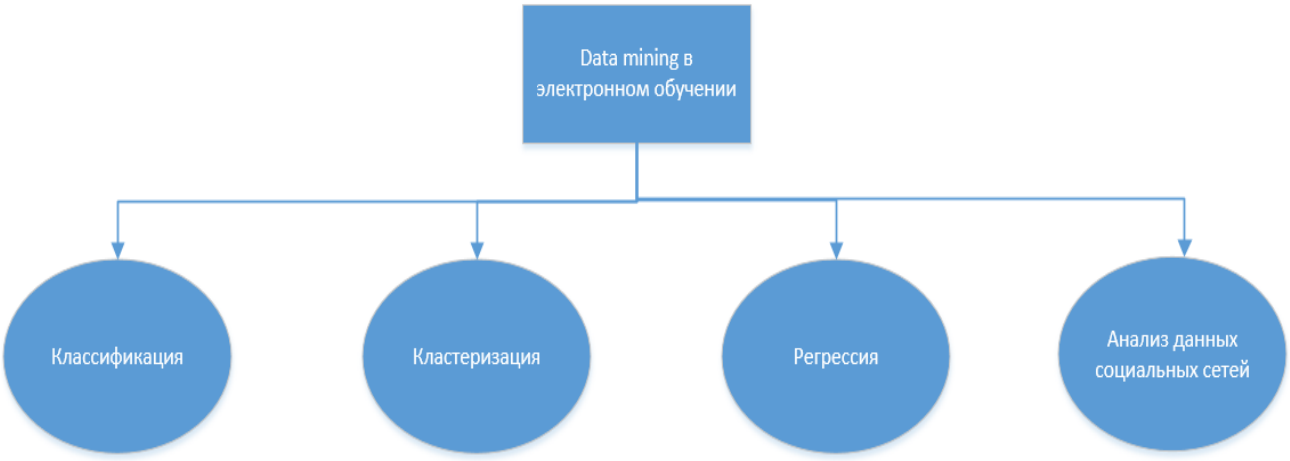


Рисунок 6 – Задачи data mining в электронном обучении

Не всю информацию в мире можно отнести к большим данным, для определения такой информации существуют определенные характеристики. Кроме того, анализ не всей информации представляется возможным. Понятие больших данных заложено в их характеристиках и все они умещаются в три V [6]:

- объем (от англ. volume). Данные измеряются в величине физического объема «документа», подлежащего анализу;

- скорость (от англ. velocity). Количество данных постоянно увеличивается. Для того, чтобы не замедлять процесс анализа, скорость обработки этих данных должна увеличиваться для получения своевременных результатов;

- многообразие (от англ. variety). Данные могут быть разного формата. Могут быть разрозненными, структурированным или структурированными частично. Например: текстовая информацию, фото и видео файлы и т.д.

В различных источниках иногда добавляют еще характеристики достоверности и жизнеспособности.

Характеристики больших данных позволяют сформулировать основные выводы для работы с ними. Существуют следующие принципы работы с big data:

- горизонтальная масштабируемость. Большие данные подразумевают использование огромного количества информации таким образом любая программа, которая подразумевает обработку больших данных, должна иметь возможность доработки и расширения. Если объем данных увеличивается, то нужно

увеличить количество аппаратного обеспечения пропорционально количеству выросшего объема данных;

- отказоустойчивость. В соответствии с предыдущим принципом, существует возможность использования ресурсов нескольких ЭВМ в кластере. Например, hadoop-кластер yahoo имеет более 42000 машин. Из-за различных факторов одна или несколько машин может выйти из строя. Необходимо предусмотреть возможность замены одних ресурсов на другие без существенных изменений в работу системы;

- локальность данных. При работе с большими данными нет возможности хранить информацию только на одной машине, поэтому в этом случае часто используются алгоритмы распределенной обработки данных. Если пересылать данные на одну машину, выделенную непосредственно для обработки, то затраты на передачу этой информации, будут очень существенны. Принцип локальности гласит, что лучше выполнять обработку информации на той машине, где она хранится, а после этого при необходимости пересылать её [7].

1.3 Существующие методы решения задачи анализа и обработки

1.3.1 Обзор методов кластеризации

Для того чтобы достигнуть цели работы, необходимо выбрать метод кластеризации, стеммер и фреймворк для работы. Поскольку каждый из этих элементов выполняет свои функции, то их следует рассмотреть отдельно.

На текущий момент времени нет единой классификации алгоритмов кластеризации. Опишем некоторые существующие виды алгоритмов и выберем алгоритм, наиболее подходящий для данной работы.

Во многих языках программирования процесс кластеризации автоматизируется при помощи библиотек, тем самым облегчая задачу программиста. Язык python предоставляет для создания кластеризации библиотеки pandas, и scikit-Learn. Их функционала достаточно для решения поставленной задачи.

По способу обработки данных методы делятся на иерархические и неиерархические (рисунок 7).

Рисунок 7 – Виды методов кластеризации по способу обработки данных

Результатом иерархического метода является иерархия, которая выполняет разделение множества исходных объектов на произвольное число кластеров. *Иерархические методы* можно разделить на восходящие и нисходящие.

Нисходящие алгоритмы выполняются по принципу «Сверху-вниз» (Дивизимные). Объекты помещаются в кластер, который в затем делится на малые кластеры. Примеры: Rock, chameleon, cure [8].

Наиболее распространены алгоритмы по принципу «Снизу-вверх» (Агломеративные), которые помещают каждый объект в отдельный кластер, а впоследствии их объединяют. Самые популярные алгоритмы: Single-link и Complete-link.

- single-link – на каждом шаге объединяет два кластера с наименьшим расстоянием между двумя любыми представителями;

- complete-link - на каждом шаге объединяет два кластера с наименьшим расстоянием между двумя наиболее удаленными представителями [9].

Неиерархические(плоские) методы строят одно разбиение объектов на кластеры. В свою очередь данные методы могут быть разделены по способу анализа данных на: четкие и нечеткие.

Четкие методы разбивают исходное множество объектов на несколько непересекающихся подмножеств. Любой объект из исходного множества должен принадлежать только одному кластеру.

Нечеткие методы позволяют одному и тому же объекту принадлежать одновременно нескольким кластерам, но с различной степенью.

Нечеткая кластеризация в некоторых случаях выглядит более натурально. Как пример можно рассмотреть ситуацию, когда элементы расположенных на пересечении кластеров.

Самыми распространенными методами четкой кластеризации являются:

а) **K-means** - простой метод разделения данных на «К» различных непересекающихся кластеров, на основе наблюдений должно быть определено фиксированное число кластеров, они должны быть максимально различны друг от друга. В процессе работы алгоритм относит каждое наблюдение к одному из кластеров [10];

1) достоинства: скорость работы и время выполнения, легкость изучения и понимания;

2) недостатки: медленная скорость работы при больших объемах информации, нужно задавать число кластеров, сбой работы при выбросах.

б) **PAM** – улучшенный метод работы К-средних. В начале работы указывается количество кластеров и множество элементов. На выходе выдается разбиение множества элементов по кластерам. В отличие от К-средних, элементы делятся относительно медианы кластера, а не центра;

1) достоинства: совпадают с К-средних, но при сбоях работы более устойчив;

2) недостатки: нужно определить число кластеров перед началом работы, медленно работает при большом объеме данных.

в) **Самоорганизующиеся карты Кохонена** – в основном предназначены для обнаружения новых явлений, проведения разведочного анализа, и кластеризации многомерных векторов.

1) достоинства: используется нейронная сеть, сеть самоорганизуется, обучение сети без учителя, гарантированный ответ, прост в реализации;

2) недостатки: Работа ограничена числовыми данными, необходимость задавать количество кластеров.

г) **CLOPE** – алгоритм применяется ко множеству транзакций. Самая полезная для алгоритма информация о каждом кластере сохранена на протяжении работы всего алгоритма. Для этого алгоритма не нужно указывать количество кластеров, он сам их создает. Из-за этого увеличивается качество группировки в кластерах и производительность системы в общем [11];

1) достоинства: сам выбирает количество кластеров, расширяемость, скорость сканирования, набор данных сканируется минимальное количество раз.

д) Среди методов нечеткой кластеризации наиболее популярным алгоритмом является **Fuzzy C-means**, представляющий собой модификацию метода K-means. Данный метод позволяет разбить существующее множество на заданное количество нечетких множеств. Он отличается от метода k-means тем, что позволяет рассчитать степень принадлежности для каждого кластера.

1) достоинства: Возможность добавлять элементы, расположенные на пересечении кластеров;

2) недостатки: Сложность вычислений, нужно определить количество кластеров.

Кроме того, существуют следующие классификации методов:

а) По количеству применений:

1) выполняется один раз;

2) может выполняться много раз.

б) По возможности масштабирования объема обрабатываемых данных:

1) расширяемые;

2) нерасширяемые.

в) По времени выполнения:

1) потоковые (on-line);

2) не потоковые (off-line).

Для реализации поставленной задачи был выбран метод k-means из-за простоты понимания и быстроты работы, он дает возможность предварительного разбиения на группы большого набора данных, после чего можно провести более мощный кластерный анализ подкластеров, кроме того, он является наиболее популярным методом кластеризации. Работа данного метода будет рассмотрена при формировании программного решения задачи

1.3.2 Обзор стеммеров

Поскольку в русском языке преобладает словообразование при помощи аффиксов, он допускает использование стеммеров. В работе будет произведен анализ русскоязычного текста, поэтому следует определить наиболее популярные русскоязычные стеммеры, и выбрать несколько из них для осуществления стемминга. Это выполняется для того, чтобы исключить дублирование, возникающее из-за неверного падежа слов, при объединении таблиц.

Стеммер Портера. Был создан Мартином Портером в 1980 году для анализа текстов на английском языке, в будущем функционал был расширен для некоторых других языков. Стеммер был написан для большинства индоевропейских языков, в том числе и русского [12].

В этом алгоритме для определения основы используются суффиксов и правила, заданные вручную. Базы данных с основами слов и словарями не используются.

В алгоритме заложены определенные правила, с которыми он проверяет соответствие на каждом этапе. Всего существует пять этапов. В каждом этапе удаляется суффикс, от которого образованно слово, а то, что осталось снова проверяется вышеупомянутыми правилами. Переход на другой этап выполняется только в том случае если полученное слово удовлетворяет правилам. Когда слово не удовлетворяет правилам алгоритм, удаляет другой суффикс и так происходит до согласования с правилом. На первом шаге удаляется максимальный суффикс, от которого образованно слово. На втором — буква «и». На третьем — суффикс, от которого образованно слово. На четвёртом — суффиксы превосходных форм, «ь» и одна из двух «н» [13].

Основной недостаток этого стеммера, заключается в том, что он может удалить лишнее. Это может сделать невозможным получение правильной основы слова. Из-за возможных изменений корня стеммер может работать некорректно.

Stemka. Андрей Коваленко разработал этот стеммер в 2002 году. Этот стеммер предполагалось использовать, как анализатор морфологии русского языка [14].

Он основан на модели вероятностей: выбранные слова разделяются стеммером на пары, состоящие из двух последних букв основы слова и суффикса слова. Если подобная пара добавлена в модель до этого, то её значимость возрастает. Если такой пары нет, то она добавляется в модель.

Все полученные данные сортируются по значимости. Если вероятность использования модели меньше 0,0001, то они удаляются. В результате работы получается набор возможных окончаний с их предшествующими символами. Создаются таблицы перехода, в которых данные добавляются «слева-направо». Анализируемые слова проверяются при помощи сформированных таблиц перехода. Существует особое правило для этого алгоритма: в основе, которую нельзя изменить должна существовать хотя бы одна гласная буква.

MyStem. Создан Ильей Сегаловичем в 1998 году, был передан в собственность компании Яндекс.

В начале работы алгоритм использует дерево суффиксов во входном слове, и на его основе анализируются возможные границы между основой и суффиксом. Следующим этапом для всех возможных основ является проверка её наличие в словаре, либо нахождение схожих с ней основ (проверяется длинна «хвоста»). Если слово словарное — алгоритм заканчивает работу, иначе — переходит к следующему разбиению [15].

Рассматриваемое слово может не иметь нужный вариант основы, например в том случае если их нет в словаре. В этом случае помимо основы слова, берется суффикс и при помощи основы генерируется модель возможного изменения

слова. Гипотеза сохраняется, а в том случае если она уже существовала, её важность увеличивается. Если в словаре не существует анализируемого слова, то его окончание уменьшатся на единицу и дерево снова проверяется на существование новых гипотез [16].

На следующем этапе рассматривается "хвост" и если его длина равна двум, то гипотезы начинают сортировку. Удаляются гипотезы, у которых важность в пять раз выше самого большого веса. Результат работы стимера – набор гипотез для несуществующего или одна гипотеза для словарного слова.

В правилах компании Яндекс указано что этот стеммер может использоваться для коммерческих целей. Компания напоминает, что стеммер не может использоваться для создания и распространение спама, оптимизации поиска сайтов и систем функционал которых аналогичен продуктам компании Яндекс. Исходный код не распространяется. Для установки стеммера нужно скачать и распаковать архив.

Ильей Сегаловичем было произведено сравнение этих стеммеров, в статье “A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine”, согласно ей было произведено сравнение стеммеров, где выходными данными были пары PPMV (форма+основа).

Чтобы уменьшить образец выборки, они оставили только словоформы, которые встречаются в корпусе с частотой от 10 до 100, т.е. являются наиболее близкими к специфике темы (рисунок 8).

Module	Total PPMVs	PPMVs with 10-100 freq
Canonical	484462	70028
Stemka (aggressive)	628154	82895
Stemka (conservative)	108248	18014
Mystem	718216	93863
Snowball	519262	72568

Рисунок 8 – Количество пар, размещенных в корпусе

После они убрали канонические PPMV, (взятые напрямую из мэппинга) и получили список добавленных и удалённых PPMV для каждого стеммера. Это выполнялось для измерения максимального эффекта (как положительного, так и отрицательного) влияния стеммера на поисковую систему. Основной задачей было определить, насколько не подходят PPMV, добавленные отдельным стеммером и насколько хороши те PPMV, которые стеммер исключил, по сравнению с каноническими PPMV (рисунок 9).

Module	**Added PPMV**	**Hosts**	**Lost PPMV**	**Hosts**
Stemka conserv.	136	46	1384	997
Stemka aggres.	787	210	493	269
Snowball	643	219	487	308
Mystem	778	403	41	7

Рисунок 9 – Расчет новых множеств для PPMV

В результате авторы сделали вывод, что хоть число PPMV определенных Mystem больше, они намного точнее и количество нерелевантных хостов меньше, как и количество потерянных хостов. Следует так же упомянуть, что стеммер Snowball является вариантом стеммера Портера.

Таким образом в работе было решено использовать стеммеры Портера и Mystem, поскольку Mystem выдает более точные результаты, а стеммер портера и stemka выдали похожие результаты на основании исследования, несмотря на то что в их основе лежат разные алгоритмы.

1.3.3 Обзор фреймворков

Особенностью работы с большими данными является то, что многие фреймворки, используемые при обработке и анализе данных не исключают использования друг друга, а могут работать вместе. Примером является Spark, работа которого дополняет функции фреймворка Hadoop.

Работа Spark должна контролироваться менеджером кластера и распределенной системы хранения данных. Для того чтобы управлять кластерами, можно

использовать встроенные средства: Hadoop YARN или Apache Mesos. Для распределенного хранения данных могут использоваться сторонние инструменты. Это расширяет возможности системы и позволяет пользователю устанавливать свои решения для использования технологий Big Data. Например, установка spark поверх фреймворка Hadoop: такая совокупность программы дает возможность ускорить выполнение программы до 100 раз быстрее в RAM и до 10 раз быстрее на ROM [17].

В фреймворке Spark используется абстракция под названием «Устойчивый распределенный набор данных» (RDD), это группа объектов только для чтения, которые распределены по узлам кластера. С помощью RDD происходит распределение исходной информации по кластерам. RDD используются для выполнения преобразований и действий.

Преобразования не предназначены для возврата значений. Они меняют метаданные и возвращают новый RDD. Примером преобразований можно считать операции: map, filter, flatMap, и т.д.

В модуле MapReduce, фрейморка Hadoop, присутствуют только одноименные функции, а spark помимо них предлагает другие возможности. Поэтому следует использовать spark только для замены модуля Hadoop MapReduce.

Архитектура spark состоит из трех следующих компонентов (рисунок 10):

- хранилище данных;
- интерфейс программирования;
- менеджер кластера.

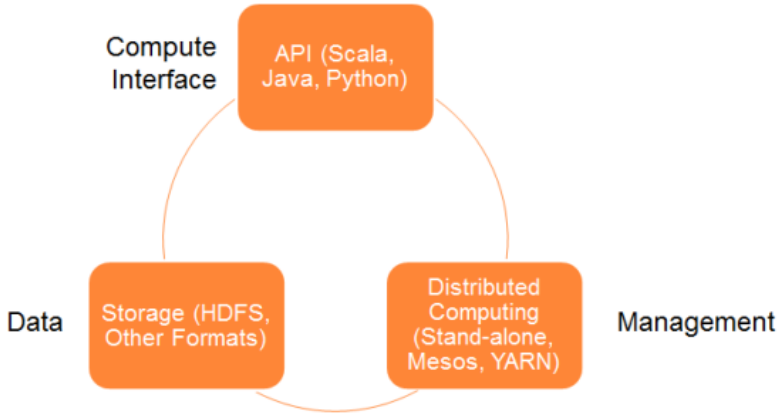


Рисунок 10 – Архитектура Apache Spark

Как видно из рисунка 10, spark по умолчанию поддерживает три языка программирования: Scala, Java и Python. Кроме того, он включает в себя несколько библиотек:

- spark SQL – для выполнения sql запросов;

- spark Steaming – для добавления возможности обработки потоковых данных;

- spark MLib – для добавления возможности машинного обучения;

- graphX – для распределенной обработки графов.

Можно сделать вывод, что использовать spark в «чистом» виде следует при машинном обучении, но все свои преимущества он раскрывает «поверх» установки на Hadoop.

Вторым наиболее популярным фреймворком является Storm. Он отличается от других фрейморков для обработки больших данных своим подходом.

Hadoop это система, которая принципиально осуществляет только пакетную обработку. Информация передается в файловую систему Hadoop (HDFS) и распределяется между машинами кластера. После обработки полученные данные снова помещаются в файловое хранилище, но с новым типом данных определенным пользователем.

Топологии storm могут преобразовывать незавершенные потоки данных. Эти преобразования, в отличие от фреймворка Hadoop, выполняются постоянно. Данные обрабатываются по мере их поступления на кластеры т.е эта система предназначена для обработки данных в режиме реального времени. Кроме того, она не зависит от языка программирования что, несомненно, является плюсом этой технологии [18].

Эта система не может использоваться для разработки приложений под кластеры Hadoop, но разработчики пытаются разработать библиотеки для решения этой проблемы.

Кластер Apache Storm, состоит из следующих компонентов [19]:

- ведущий узел с запущенной системной службой Nimbus, она создает задачи для машин и отслеживает их работу;

- выделенные узлы, на которых запущена служба супервизор. Она распределяет задачи между рабочим узлам и управляет ими.

Storm не может отследить состояние кластера, поэтому им используется модуль ZooKeeper которая связывает Nimbus с супервизорами (рисунок 11).

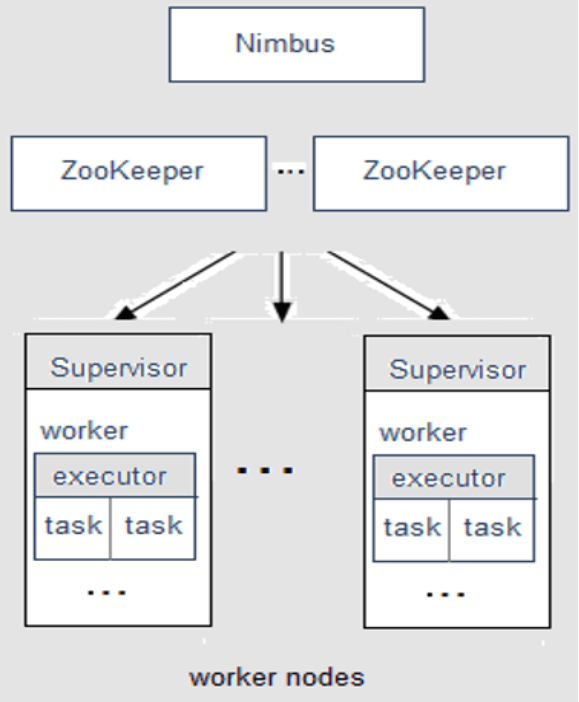


Рисунок 11 – Архитектура Apache Storm

Как и в map reduce, работу данной системы можно представить, как совокупность определенных элементов: воронок (spouts) и сит (bolts). Воронки создают потоки данных в форме неизменяемых пар «ключ-значение» (кортежи *(tuples)*), а сита преобразуют эти потоки, создают связь с внешними базами данных, разделение, объединение.

Поток – это неограниченный конвейер кортежей, а воротки являются источниками потоков. Они преобразуют данные в кортеж потоков и отправляют их на обработку в сита. В целом эта платформа работает, как механизм для преобразования данных. Однако в отличие от MapReduce данные обрабатываются в режиме реального времени, а не пакетами. Кроме того, основным отличием от Hadoop предоставляет возможность гарантированной обработки сообщений т.е каждый «tuples», испускаемый из «bolt», будет обработан, если он не обработан в течении определенного времени, то storm повторяет отправку (рисунок 12).

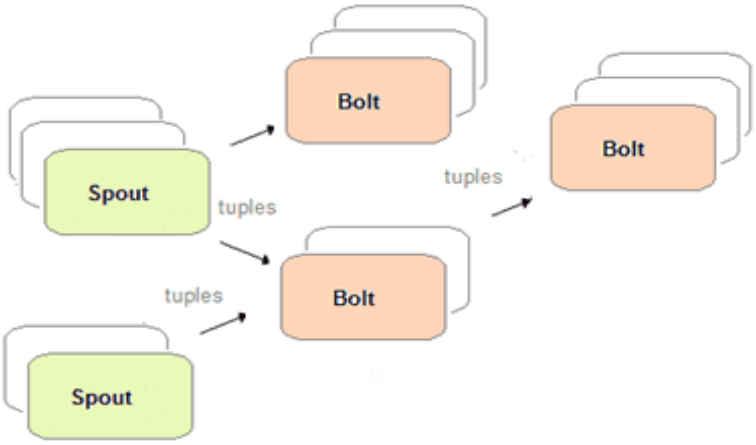


Рисунок 12 – Топология Apache Storm

Обобщая все вышесказанное можно выделить следующие преимущества Apache Storm:

- масштабируемость;
- гарантия обработки;
- поддерживает много языков программирования;
- простота развертывания;
- отказоустойчивость.

Несмотря на все эти преимущества у нее есть следующие недостатки:

- поддерживает только однократную доставку сообщений;
- не может управлять состоянием приложений;
- отсутствует настройка возможности обработки событий в разных периодах.

1.4 Обзор существующих программных решений

Перед проектированием структуры программы, необходимо проанализировать рынок на предмет программ, с похожими функциональными возможностями, чтобы понять их преимущества и недостатки, и учитывать их при разработке.

Наиболее популярная программа - *IBM SPSS Statistics*. Данная программа предназначена для статистической обработки данных, и является одним из лидеров рынка в этой области, основной её функционал – проведение прикладных исследований в общественных науках.

Программа предоставляется на условиях покупки лицензии, которая делиться на несколько изданий – base, standard, professional, premium. Для сравнения функционала версий стоит привести примеры функции, которые есть в версии base и версии premium, пользователи версии base могут проверять основные гипотезы, подготавливать диаграммы и графики, выполнять кластерный анализ. Пользователи premium могут выполнять кросс-валидацию, точные тесты и анализ RFM.

Есть вариант подписки годовой или месячной, под названием IBM SPSS Statistics Subscription Base Edition, функционал которой совпадает с базовой версией покупки программы, для расширения функционала так же можно докупить три дополнения. Преимуществом является то, что существует специальная лицензия программы для учебных целей.

Безусловно данный продукт является лидером на рынке, но у него есть несколько существенных недостатков, один из которых – недоступность для обычного пользователя, для того чтобы даже узнать цену программ и подписки необходимо связываться с официальным дистрибьютором, что не совсем удобно, согласно данным мультисервисного провайдера syssoft, цена подписки составляет от 7990 рублей в месяц.

Проанализировав продукт можно выделить следующие преимущества:

- аналитика больших данных;
- прогнозное моделирование;
- статистический анализ;
- отчетность;
- несколько вариантов подписки;
- кроссплатформенность.

Но также есть несколько недостатков:

- возможность программирования для опытных пользователей, использующих языки python и R;
- сложность получения для обычного пользователя;
- усложненная модель распространения, в том числе с дополнениями.

Второй программный продукт – *Statistica*. Разработчиком этого продукта является компания StatSoft. Этот пакет программ реализует функции анализа данных, управления данными, добычи и визуализации данных.

Обработку больших данных выполняет не столько базовая версия программы, как несколько модулей включенных в расширенные издание программы. В отличие от IBM SPSS Statistics, программу нельзя приобрести по подписке, только покупка, более того количество версий и дополнений программы гораздо больше, чем у конкурента, и так же, как и у них у Statistica, есть версия для учебных организаций.

На сайте организации StatSoft расположена статья под названием «Революция Big Data: как извлечь необходимую информацию из «Больших Данных»?», где авторы описывают своё видение работы с большими данными. Они упоминают о такой технологии как Map-reduce и системе Hadoop, которые были использованы в данной работе [20].

Компания заявляет, что при работе с большими данными, выбранная платформа должна использовать технологию map-reduce. Платформа STATISTICA Enterprise и Decisioning дает все возможности для максимально продуктивной работы с большими данными.

К преимуществам программы следует отнести:

- кластерный и статистический анализ;

- анализ главных компонент и классификация;

- есть веб-версия;

- есть дополнительная возможность для добычи данных над текстами.

Недостатки программы:

- все преимущества программы доступны только в расширенной версии;

- цена от 38 тысяч рублей, за каждый из продуктов. Версия для учебных целей стоит от 8 тысяч.

- усложненная модель распространения, в том числе с дополнениями.

2 РЕШЕНИЕ ЗАДАЧИ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ В СФЕРЕ БОЛЬШИХ ДАННЫХ

2.1 Преобразование информации

Первое что следует выполнить при работе с большими данными, это определить вид информации, с которой необходимо работать и описать процесс преобразования этой информации к виду, пригодному для машинной обработки.

В данной работе исследуемые данные – текстовая информация. Следовательно, необходимо осуществить кластеризацию текста, данная задача разбивается на две ключевые составляющие:

- преобразование информации в другую модель (векторную, матричную и тд);

- математическая задача кластеризации.

Анализ большого количества текстовых данных необходимо начинать с предварительной обработки документов, которая состоит из нескольких этапов:

- токенизация — разбиение текста на отдельные слова. Полученные слова переводятся в нижний регистр, так же включает в себя удаление специальных символов и знаков пунктуации, и цифр при необходимости;

- стемминг (англ. stemming) — полученные слова приводятся к основе, учитывая морфологию слова.

- удаление стоп-слов – к ним относятся союзы, местоимения, предлоги и другие «связующие» части речи;

Стеммером называется алгоритм, при помощи которого реализуется стемминг. Большинство стеммеров осуществляют усечение окончаний, суффиксов и префиксов, основываясь на особенностях языка. Одним из самых распространённых стеммеров является стеммер Портера. Он находится в свободном доступе на сайте автора.

После обработки множество полученных документов следует представить в виде матрицы, где каждый документ – вектор в пространстве.

2.2 Сравнение методов классификации и кластеризации

Поскольку при работе с большими данными объем информации огромен, то необходимо разбить информацию на определенные группы, это является следующим шагом при выполнении рассматриваемой задачи. Для осуществления поставленной задачи рассмотрим понятия классификации и кластеризации, выявим их отличия и выберем какой из этих методов должен использоваться в работе.

Кластеризация выполняет автоматическое разбиение элементов некоторого множества на группы в зависимости от их сложности. Элементами множества может быть любая информация. Сами группы называются кластерами.

Классификация – системное распределение изучаемых предметов, явлений, процессов по каким-либо существенным признакам для удобства их исследования. Выполняется группирование исходных понятий и их сортировка в таком порядке, который отражает их похожесть [21].

Данные понятия очень схожи, но задача кластеризации – это логическое продолжение задачи классификации, т.е при кластеризации группы создаются непосредственно при получении результата, а для классификации нужно заранее определить необходимые группы.

Таким образом для выполнения поставленной задачи следует использовать метод кластеризации, потому что нет четкого представления о составе и группах данных, а отбор ручным способом сложен и трудоемок. Кроме того, для осуществления классификации нужно создать обучающую выборку, т.е можно сделать вывод, что классификация занимает больше времени. В работе было решено решать задачу кластеризации, из-за скорости её работы и отсутствия необходимости задавать количество кластеров.

Задачи, которые выполняет кластеризация, во многом совпадают с задачами Data mining, опишем эти задачи:

- обобщение данных, при помощи кластеризации производит отбор только тех данных, которые нужны пользователю;

- сегментирование потребителей, позволяет осуществить анализ предполагаемых действий групп потребителей. Это необходимо в таких областях как журналистика, маркетинг и т.д;

- анализ данных социальных сетей. Важность данной задачи была описана при рассмотрении области образования. Она помогает представить взаимосвязь пользователей в виде модели, где узлами являются группы друзей или сообщества;

- интеллектуальный анализ, как пример нахождение аномальных значений;

- вычисление значений меры сходства между двумя объектами;

- создание групп сходных объектов.

2.3 Алгоритм работы выбранного метода кластеризации (K-means)

Следует более подробно рассмотреть работу выбранного алгоритма кластеризации – k-means.

Как и говорилось ранее, метод создает к-групп из набора объектов таким образом, чтобы члены одной группы были наиболее однородными. Алгоритм пытается минимизировать суммарное квадратичное отклонение точек кластеров от их центров (формула (1)) [22]:

$$V = \sum_{i=1}^{k} \sum_{x \in S_i} (x - \mu_i)^2 \quad (1)$$

где k – требуемое число кластеров;

S_i – полученные кластеры;

μ_i – центры масс векторов x, из кластера S_i.

В начале работы алгоритма необходимо задать число кластеров, которые необходимо сформировать. Затем происходит разбиение всех объектов на k кластеров.

Весь алгоритм можно представить как последовательность четырех шагов:

- выбирается k произвольных точек, служащие в дальнейшем, как начальные центры кластеров (Центройды - μ_i);

- для каждого центройда вычисляются расстояния до точек(x). Для каждой итерации нужно переопределять расстояние между записями и центрами кластеров, это необходимо для определения принадлежности записи к кластеру (могут использоваться различные метрики в зависимости от ситуации, как пример, Евклидово расстояние и расстояние Манхэттена);

- формируются кластеры, для каждого центройда отбирается подмножество точек с минимальным расстоянием до центров по формуле (2):

$$argmin \sum_{i=1}^{k} \sum_{x \in S_i} (x_i - \mu_k)^2 \quad (2)$$

где k – требуемое число кластеров;

S_i – полученные кластеры;

μ_k – центры масс векторов, из кластера S_i.

- Выполняется пересчет цетройдов, по формуле (3):

$$\mu_k = \frac{1}{S_k} \sum x_k \quad (3)$$

где S_k – количество точек в кластере К.

3 и 4 шаги выполняются пока работа не будет прервана, либо пока не будет выполнено условие сходимости, алгоритм останавливается, когда границы кластеров и расположение центройдов не перестанут изменяться, при смене итерации.

Вычислительные затраты в алгоритме увеличиваются пропорционально количеству записей определяются по формуле (4):

$$k*n*I \quad (4)$$

где k- количество кластеров (заранее задано);

n – число записей;

I – число итераций (заранее задано);

Нельзя однозначно сказать о том требует ли обучения этот метод, поскольку метод не уточняет количество кластеров, а исследует самостоятельно, к какому кластеру относится объект.

В некоторых языках программирования существуют множество библиотек для упрощения процесса кластеризации. Это очень удобно, поскольку позволяет существенно сократить временные затраты на разработку кода.

2.4 Парсинг, как средство для сбора больших данных

Для того, чтобы накопить большие данные необходим источник информации, который будет генерировать данные с некоторой периодичностью, либо постоянно. Таким источником на текущий момент является глобальная сеть. Данные имеющие ценность для пользователя необходимо собирать из глобальной сети, а в этом может помочь парсинг.

Парсинг – это принятое в информатике определение синтаксического анализа.

Для создания парсера создается математическая модель сравнения лексем с формальной грамматикой, описанная одним из языков программирования.

Можно использовать уже готовые парсеры, но большинство программистов хотят изменять в будущем свой парсер под определенные нужды, либо добавлять в него новые возможности. Для парсинга, в настоящее время, наиболее удобен язык Python. Он используется такими порталами, как Youtube, Facebook, Instagram, Google.

Самое сложное при работе с парсером, это постоянная доработка и зависимость от конкретной структуры сайтов и области исследования. Для того чтобы получить информацию с сайта и обработать её в дальнейшем, нужно досконально знать структуру сайта, поскольку необходимо явно описывать теги и классы элементов, из которых информацию нужно получить.

В том случае если нужно получить всю информацию, наиболее простой способ – полная выгрузка сайта и её постраничный анализ, что невозможно в условиях прикладного использования т. к. количество сайтов, в котором нужно

провести парсинг не ограничиться несколькими десятками, и для хранения этой информации нужно место на жестком диске. Кроме того, нужно постоянно загружать копию сайтов с некоторой периодичностью из-за обновления информации. Для анализа информации в зависимости от времени нужно будет хранить копии за определенный период времени, что неудобно, гораздо выгоднее занести нужные данные в БД и удалить избыточную информацию, либо осуществлять поиск ссылок на странице и осуществлять поиск информации сразу.

2.5 Фреймворк Hadoop

Начать разработку следует с выбора платформы, на которой будет реализован продукт. Поскольку необходимо проанализировать большое количество информации, платформа должна предоставлять инструменты для распределенной обработки данных, в пункте 1.3.3 “Выбор фреймворка”, были рассмотрены наиболее популярные фреймворки, предоставляющие эту возможность, но было принято решение использовать фреймворк Hadoop, преимущества которого следует обосновать.

Hadoop – это целый набор библиотек, утилит и фреймворк. Совокупность этих инструментов используется для выполнения следующих задач: хранение, обработка и управление данными. Многие ученые и аналитики считают, что эта технология – одна из основополагающих при работе с большими данными.

На текущий момент Hadoop можно разделать на 4 полноценные части, каждая из которых выполняет свою задачу [23]:

- распределенная файловая система «Hadoop Distributed File System» (HDFS), отвечает за хранение данных. Эта система состоит из двух частей: сервера «NameNode», необходимого для управления пространством имен файловой системы и доступом клиентов, и «DataNodes», выполняющего функции чтения, записи, удаления и т.д. HDFS хранит информацию в блоках. Блоки, в которых HDFS хранит информацию файла все одного размера кроме последнего;

- MapReduce это базис для написания приложений, выполняющих обработку больших объемов структурированных и неструктурированных данных параллельно на кластере, в который включены необходимые машины;

- Apache Hadoop YARN, осуществляет управление данными, рабочими нагрузками, многоуровневым обслуживанием, безопасностью и функциями высокой доступности, он позволяет устранить высокую задержку ввода-вывода и добавляет следующие новые модели обработки: пакетная, интерактивная, потоков, графов. С появлением YARN подход MapReduce превратился просто в распределенное приложение и теперь называется MRv2. MRv2 – это просто реализация;

- Hadoop Common – набор библиотек и утилит, которые используются в родственных проектах, она предоставляет инструменты для чтения данных, хранимых в файловой системе HDFS. Основная задача – создание инфраструктуры.

Совокупность этих возможностей предоставляет следующие преимущества:

- минимизация времени обработки информации;

- низкая стоимость оборудования;

- повышение отказоустойчивости. Технология позволяет построить отказоустойчивое решение;

- расширяемость;

- данные при работе могу быть не структурированы;

- высокая мощность вычислений.

Помимо этого, исключительной особенностью работы с Hadoop является возможность при необходимости использовать другие фреймворки, в этом случае они заменяют собой один из модулей Hadoop.

2.6 Выбор языка программирования

В качестве платформы для написания программы был выбран фреймворк Hadoop, он написан на языке Java, поэтому программы, написанные на этом языке, можно запускать в нём непосредственно как задание MapReduce. В том случае если разработчику более близки такие языки как Python и C#, то необходимо использовать потоковую передачу Hadoop.

Потоковая передача использует модули сопоставления и редукции, через потоки «STDIN» и «STDOUT», т.е работа этих модулей заключается в построчном считывании информации из «STDIN» и записи в «STDOUT». Строки должны быть в формате «Ключ-значение».

Пользователю предоставляется возможность работать с более удобным для него языком программирования, без вреда на производительность программы, поскольку каждый из языков программирования обладает своими преимуществами в сравнении друг с другом.

а) Python обладает следующими преимуществами:

1) простота – легкость чтения и понимания языка.

2) совместимость с различными платформами.

3) множество библиотек.

4) поддержка процедурно-ориентированного и объектно-ориентированного программирования.

5) динамическая типизация (возможность не указывать тип переменных);

б) Преимущества Java:

1) статически типизированный язык (необходимо указание типа переменных);

2) кроссплатформенность;

3) создание сетевых приложений.

в) Преимущества C#:

1) стандартная библиотека лучше, чем у Python;

2) лучшая производительность;

3) удобная среда заработки Visual studio;

4) наиболее удобен для разработчика, не нужно переучиваться.

При рассмотрении таких программ как «mapper» и «reducer» (функции реализующие map и reduce), можно увидеть, что код на языке python более ёмок и лаконичен, чем другие варианты, несмотря на необходимость использования потоков «STDIN» и «STDOUT», поэтому было решено использовать язык python.

Кроме того, язык python имеет огромное количество библиотек для анализа данных.

Преимущества языков программирования можно представить при помощи схемы, изображенной на рисунке 13.

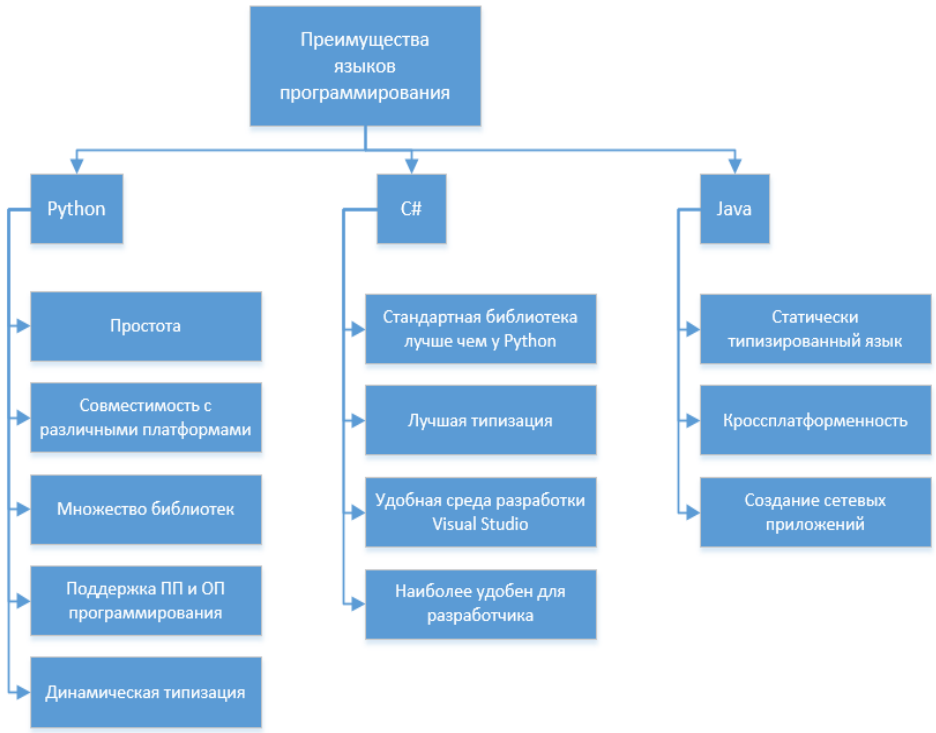


Рисунок 13 – Преимущества языков программирования

Для того чтобы разобраться в синтаксисе следует привести пример написания «mapper» (программа реализующая функцию map), и «reducer» (программа реализующая функцию reduce) для каждого языка и оценить сложность написания.

На языке python mapper выглядит следующим образом (рисунок 14):

```
#mapper.py
import sys

def do_map(doc):
for word in doc.split():
    yield word.lower(), 1

for line in sys.stdin:
    for key, value in do_map(line):
        print(key + "\t" + str(value))
```

Рисунок 14 – Mapper на языке Python

Reducer на языке python представлен на рисунке 15:

```python
#reducer.py
import sys

def do_reduce(word, values):
    return word, sum(values)

prev_key = None
values = []

for line in sys.stdin:
    key, value = line.split("\t")
    if key != prev_key and prev_key is not None:
        result_key, result_value = do_reduce(prev_key, values)
        print(result_key + "\t" + str(result_value))
        values = []
    prev_key = key
    values.append(int(value))

if prev_key is not None:
    result_key, result_value = do_reduce(prev_key, values)
    print(result_key + "\t" + str(result_value))
```

Рисунок 15 – Reducer на языке Python

Рассмотрев написание программы для этих, функция на языке python можно сделать вывод, что программа имеет достаточно простой синтаксис и код программы компактен, что удобно при дальнейшей работе с ней. Как видно из примеров (рисунок 14 и 15) для чтения строки она должна быть разделена на ключ и значение разделенные знаком табуляции.

В отличие от Python, программа в не модифицированном виде имеет внушительный размер, синтаксис её более сложен в написании, несмотря на импорт библиотек, которые предоставляют упрощенные возможности для реализации классов TokenizerMapper и IntSumReducer наследуемых от классов Mapper и Reducer.

Поскольку программа должна будет обрабатывать данные на распределенных машинах, важно чтобы платформа для разработки обеспечивала удобный

интерфейс разработки приложений в консоли. Jupiter notebook это удобный инструмент для представления проектов data science.

Реализация функций на языке C# представлена на рисунке 16:

```
public interface IKeyValuePair<TKey, TValue>
{
    Ссылок: 0
    TKey Key { get; set; }
    Ссылок: 0
    TValue Value { get; set; }
}
Ссылок: 0
public abstract class GenericMapReduce<TSource, TKey, TValue, TResult>
{
    Ссылок: 0
    public abstract IEnumerable<MapReduce.Core.KeyValuePair<TKey, TValue>> Map(TSource values);
    Ссылок: 0
    public abstract TResult Reduce(MapReduce.Core.KeyValuePair<TKey, IEnumerable<TValue>> value);
    Ссылок: 0
    public IEnumerable<TResult> MapReduce(IEnumerable<TSource> source,
    Func<TSource, IEnumerable<MapReduce.Core.KeyValuePair<TKey, TValue>>> map,
    Func<KeyValuePair<TKey, IEnumerable<TValue>>, TResult> reduce)
    {
        Contract.Requires<ArgumentNullException>(source != null);
        Contract.Requires<ArgumentNullException>(map != null);
        Contract.Requires<ArgumentNullException>(reduce != null);
        var mapResults = new ConcurrentBag<KeyValuePair<TKey, TValue>>();
        Parallel.ForEach(
        source,
        item =>
        {
            foreach (var result in map(item))
                mapResults.Add(result);
        });
        var reduceSources = mapResults.GroupBy(item => item.Key, (key, values) => new KeyValuePair<TKey, IEnumerable<TValue>>(key, val - ues.Select(value => value.Value))
        var reduceResult = new ConcurrentBag<TResult>();
        Parallel.ForEach(
        reduceSources,
        item => reduceResult.Add(reduce(item)));
        return reduceResult;
    }
}
```

Рисунок 16 – MapReduce на языке C#

Как видно из рисунка 16 программа на C# обладает теми же недостатками, что и программа на Java, а именно большой объем и сложный синтаксис, кроме того, как и в программе на Python, необходимо представить входные данные как пары “ключ-значение”.

Таким образом можно сделать вывод что наиболее оптимальным языком для написания данного продукта является язык Python, т. к. он емок и лаконичен, и часто используется при написании такого рода программ, из-за быстроты обработки данных, не смотря на необходимость использования потоковой передачи Hadoop.

2.7 Структура MapReduce

MapReduce – наиболее популярная модель для распределенной обработки данных. Эта технология предложена компанией Google, для обработки больших

объёмов данных на компьютерных кластерах [24]. Как и говорилось ранее, это модуль, в котором будет происходить написание программы.

Поскольку полученный массив данных огромен, необходимо выполнять анализ на распределенных машинах, эта технология позволяет выполнить это автоматически. Технология универсальна и пригодна для использования во многих областях: индексация веб контента, подсчет слов, обработка данных и т.д.

Принцип работы MapReduce представлен на рисунке 17.

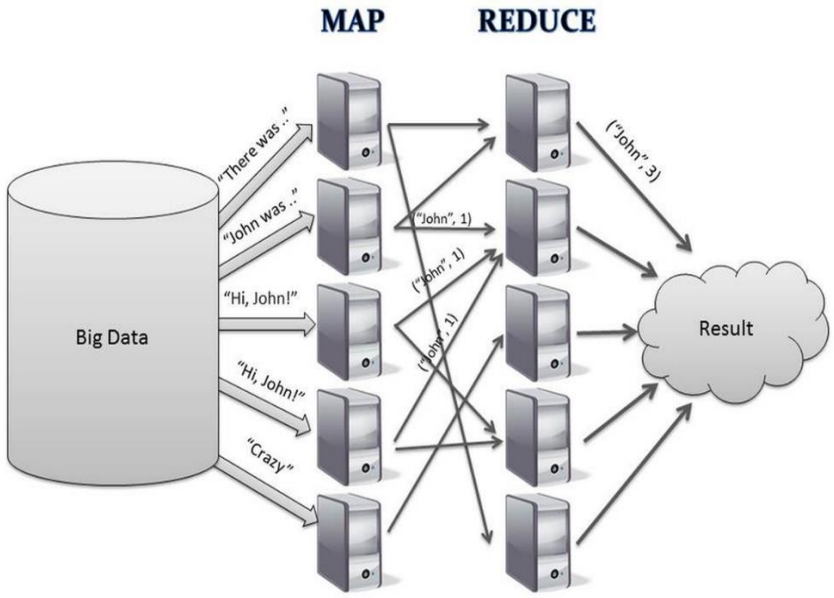


Рисунок 17 – Схема работы MapReduce

Как видно из рисунка 17, данные в MapReduce делятся и обрабатываются на отдельных машинах (при необходимости). После этого происходит объединение результатов обработки данных. Весь процесс обработки данных происходит в 3 стадии:

- стадия «Map». На этой стадии пользователем определяется функция map, отвечающая за предварительную обработку и фильтрацию данных. Соответственно стадия выполняет задачи, для которых создана функция. Для каждой записи на входе программы применяется функция, определенная на этом этапе;

- стадия «Shuffle». Скрыта для пользователя. На этой стадии результаты работы функции map делятся по «корзинам». Корзина соответствует одному

ключу вывода стадии map. В дальнейшем созданные корзины подаются на вход функции, создаваемой на следующей стадии;

- стадия «Reduce». Результат работы предыдущей стадии, попадают на вход функции «reduce». Функция reduce разрабатывается пользователем и предназначена для определения финального результата для отдельной «корзины». Множество всех значений, возвращённых функцией reduce, определяет финальный результат алгоритма MapReduce. Функции map и reduce могут работать одновременно на разных узлах [25].

Соответственно, для того чтобы произвести обработку, необходимо написать функции для стадии map и reduce и запрограммировать их.

К преимуществам MapReduce можно отнести высокую гибкость, простоту и удобство использования, она позволяет скрыть от пользователя всю ненужную информацию по организации вычислений происходящий на кластерной системе. Кроме того, данная парадигма используется такими компаниями как:

- google, для параллельных вычислений очень больших массивов данных, в кластерах;

- яндекс, для анализа и обработки данных с сайтов;

- nvidia, для распараллеливания вычислений на видео-ядрах.

Парадигму mapreduce можно реализовать различным образом. Каждая компания имеет свои закрытые реализации моделей этой технологии.

2.8 Выбор и обоснование модели жизненного цикла

Чтобы осуществить разработку программы наиболее правильно и продуктивно, следует описать процессы, действия и ход разработки программы в целом т.е выбрать модель жизненного цикла программы.

Выбор модели жизненного цикла разработки программы, и её содержание, выполняется на основании рекомендаций из ГОСТ Р ИСО/МЭК 12207-2010.

На текущий момент наиболее распространены три типа моделей жизненного цикла: каскадная (рисунок 18), итерационная (рисунок 19) и спиральная (рисунок 20).

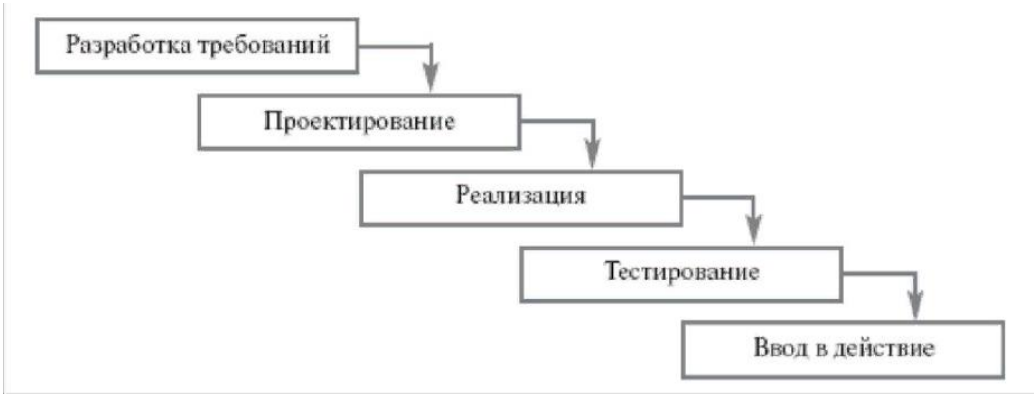


Рисунок 18 – Каскадная модель жизненного цикла

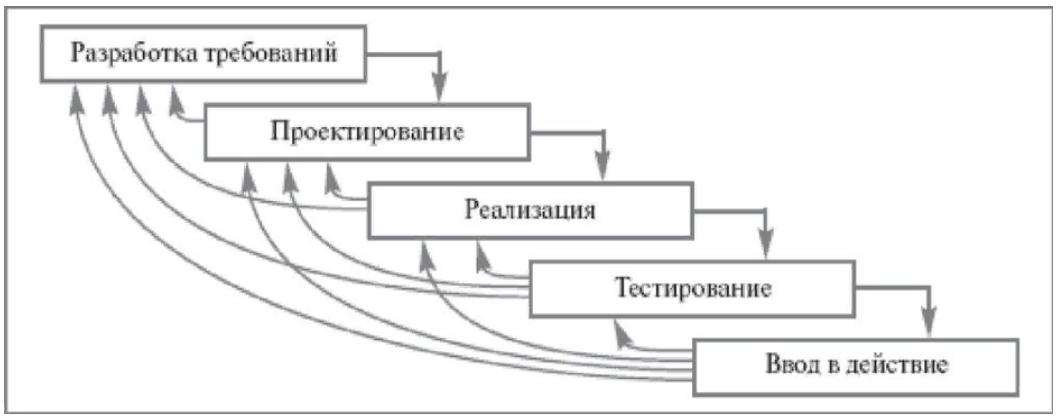


Рисунок 19 – Итерационная модель жизненного цикла

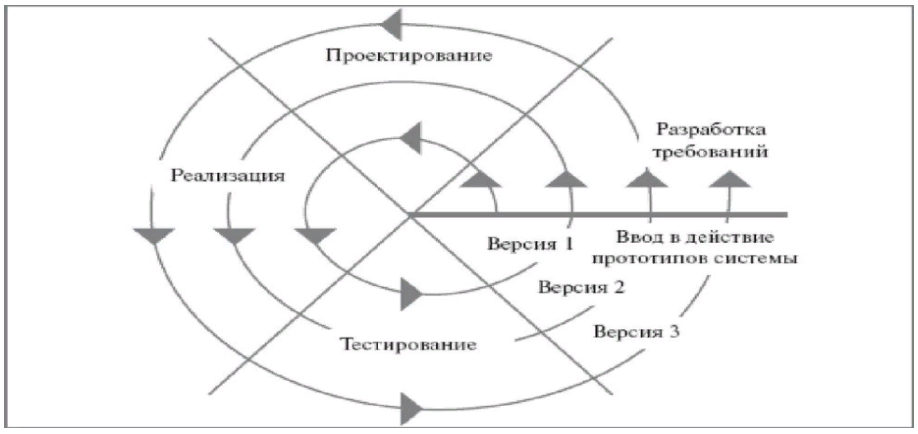


Рисунок 20 – Спиральная модель жизненного цикла

Представленные модели состоят из одинаковых процессов, но отличаются друг от друга преимуществами и недостатками.

Каскадная модель, представленная на рисунке 18, не подходит для разработки выбранной программы, поскольку она не соответствует реальным условиям разработки программного продукта, хоть и имеет весомое преимущество –

строго фиксированный порядок разработки. Такой порядок разработки невозможен, поскольку в случае возникновения ошибок необходимо вернуться на тот этап разработки, где ошибка возникла и устранить её, а это возможно только при наличии обратной связи. Отсутствие возможности параллельно разрабатывать различные функции тоже ограничивает работу этой модели, но не в данном случае, поскольку в программе результат работы одной функции является входной информацией для другой.

Спиральная модель, представленная на рисунке 20, устраняет главный недостаток каскадной модели, но в её использовании так же есть недостатки:

- создание нескольких вариантов программы;
- определение точки перехода на новый виток спирали;
- для поддержания версий продукта необходимо достаточно времени, а это не укладывается в рамки курсовой работы.

Таким образом наиболее оптимальная модель жизненного цикла – итерационная (рисунок 19). Поскольку функции программы выполняются последовательно друг за другом, после разработки какой-либо функции, и возникновении ошибки, можно вернуться на один из предыдущих этапов и исправить её. Создание межэтапных корректировок обеспечивает меньшую трудоемкость разработки по сравнению с каскадной моделью. Эта модель не требует работы с несколькими вариантами программы, и большого количества времени. Явно представлен, конец разработки, что отличает эту модель от спиральной. Таким образом, разработку программы было решено выполнять при помощи итерационной (каскадной с обратной связью) модели. Это обеспечит наибольшую эффективность разработки.

3 РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА

3.1 Проектирование взаимодействия модулей и их связи

Разработанный программный продукт состоит из совокупности модулей. Поэтому удобно представить структуру программы в виде контекстной модели, которую в последующем можно будет разложить на составляющие и продолжить их анализ.

Для того чтобы обработать данные, необходимо получить информацию с сайтов образовательных организаций. Поскольку данные, которые будут обрабатываться, находятся в открытом доступе, то нет необходимости в разработке модуля разграничения прав доступа.

Для получения конечного результата, на вход программы подаются определенные данные. Входными данными для программного продукта являются: ссылки на веб-страницы образовательных организаций с таблицей, одна или несколько необходимых диаграмм, количество специальностей/направлений подготовки необходимое для отображения, параметры для группировки. В качестве алгоритма кластеризации используется алгоритм k-means, реализуемый при помощи библиотек pandas и scikit-learn.

Работа программы регламентируется библиотекой стеммеров, а также mapreduce. За визуализацию отвечает библиотеки seaborn и matplotlib. Поскольку для работы программы необходима поддержка многомерных массивов, используется библиотека numpy, а за обработку и анализ данных отвечает библиотека pandas. Механизмами в данной системе являются: программное и аппаратное обеспечение, пользователи и фреймворк hadoop.

Выходными данными является обработанные данные в виде таблицы и графическое отображение информации в виде диаграмм, которые выбрал пользователь.

Таким образом контекстную модель программы, наиболее наглядно можно изобразить при помощи диаграммы IDEF0 показанной на рисунке 21.

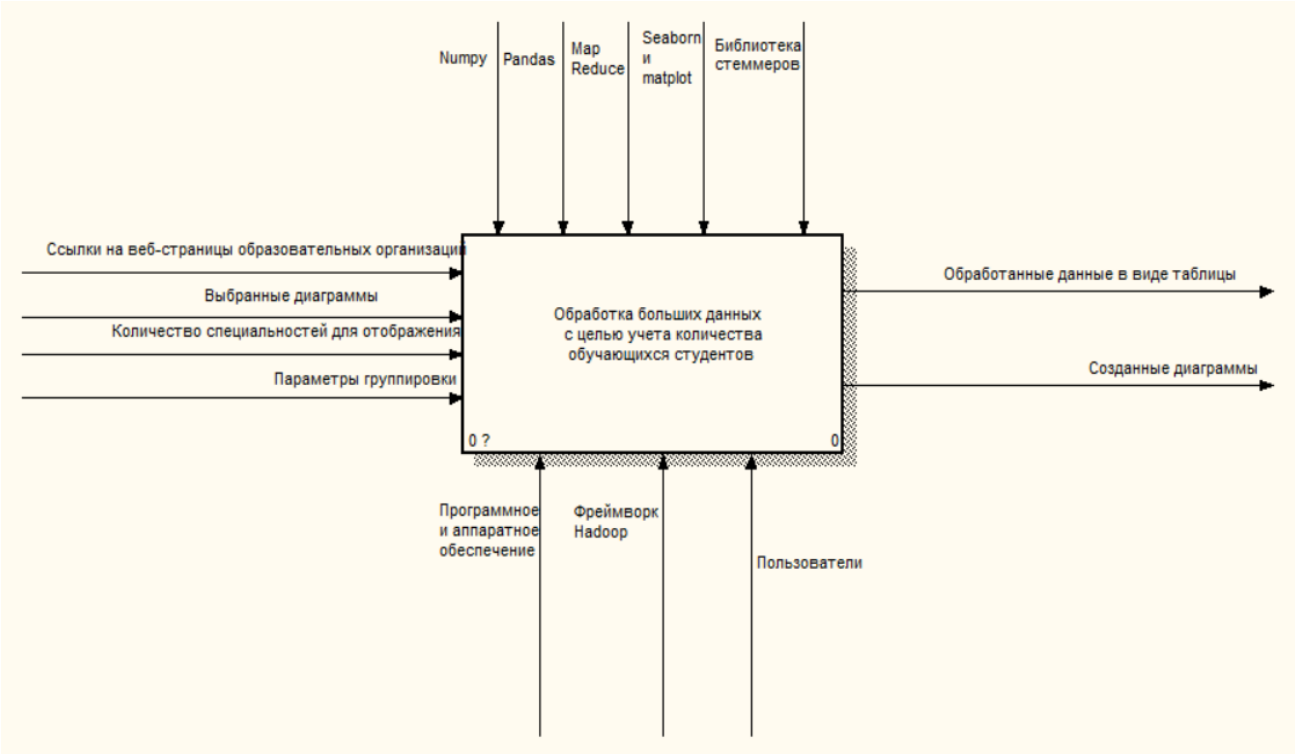


Рисунок 21 – Контекстная модель программы

В первую очередь пользователю необходимо выбрать среди каких вузов он хочет знать численность обучающихся по направлениям подготовки/специальностям, определить вид графиков, которые он хочет получить и количество отображаемых направлений/специальностей.

Чтобы лучше понять работу программы нужно провести декомпозицию контекстной диаграммы представленной на рисунке 21, на внутренние модули и определить входные и выходные данные для них.

В первую очередь пользователю необходимо решить среди каких вузов он хочет узнать численность обучающихся по направлениям подготовки/специальностям, и определить вид графиков, которые он хочет получить и количество отображаемых направлений подготовки/специальностей.

Ссылки на вузы являются входной информацией для модуля выбора и работы парсеров, в нем, в зависимости от вида ссылки, количества столбцов и уровня столбцов выбирается парсер и выгружает нужную таблицу, таким образом в модуль преобразования информации передаются не обработанные таблицы.

Осуществив декомпозицию, модель программы примет вид, представленный на рисунке 22.

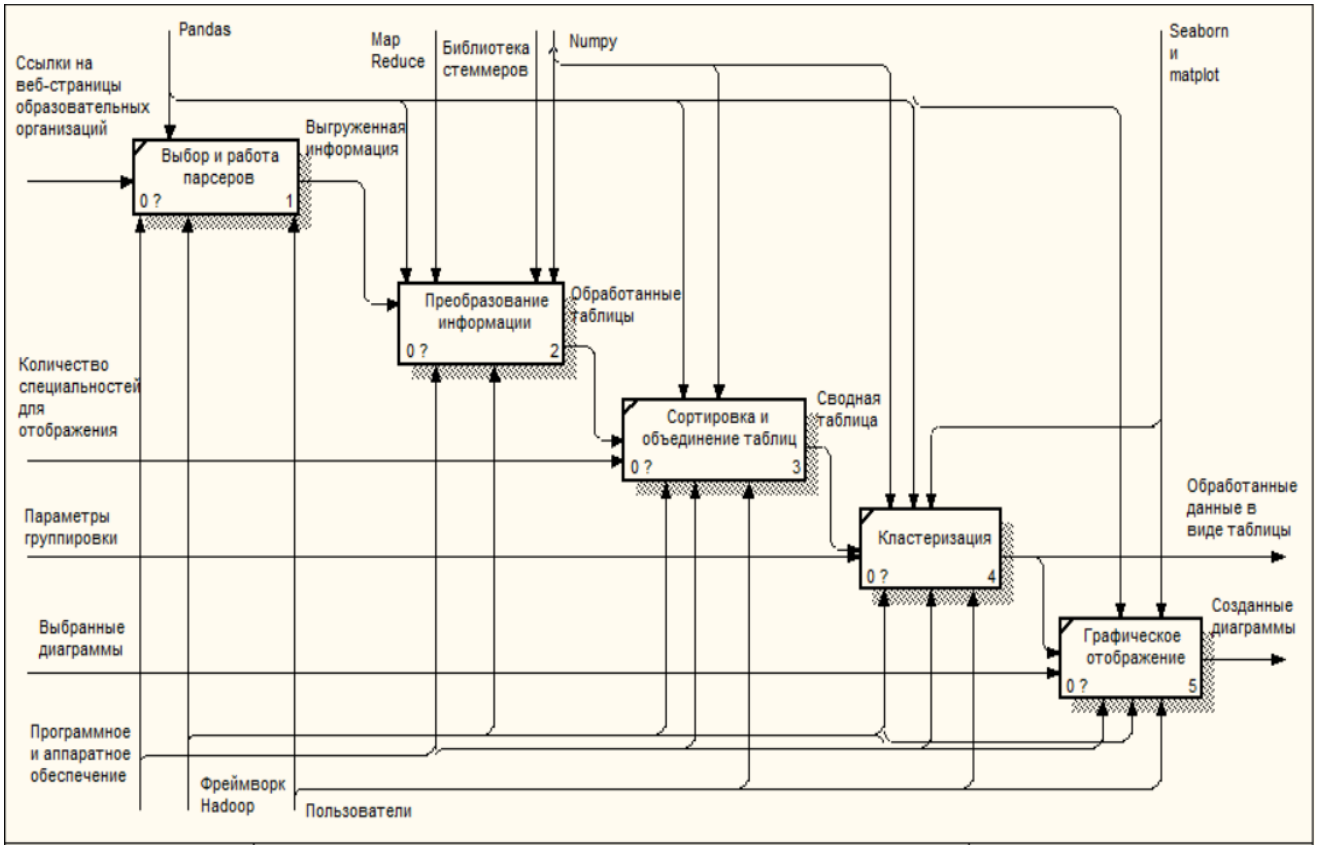


Рисунок 22 – Декомпозиция контекстной модели

После того как был выбран парсер для каждого сайта, происходит выгрузка всей необходимой информаций. Затем она передается в модуль преобразования информации.

Модуль преобразования информации необходим, для того чтобы привести полученные таблицы к единой структуре, удалить лишнюю информацию и т.д. Рассмотрим более подробно работу модуля преобразования информация поскольку он является основным модулем программы, и без него работа программы была бы невозможна. Декомпозиция функций этого модуля представлена на рисунке 23.

Выгруженная информация из открытых источников попадает в модуль преобразования информации, где на первом этапе выполняется токенизация, т. е разбиение текста на отдельные слова. Этот процесс включает в себя перевод в

нижний регистр, удаление знаков пунктуации, специальных символов и цифр при необходимости.

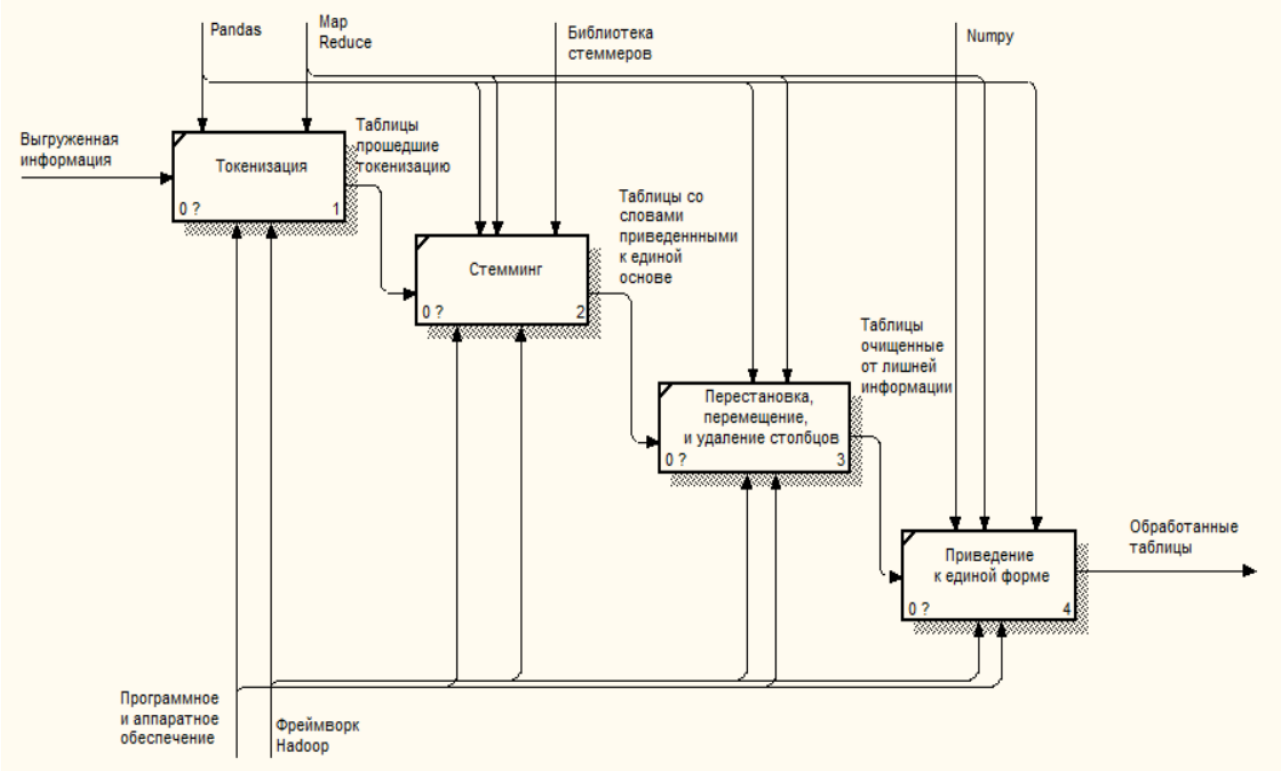


Рисунок 23 – Функциональная декомпозиция модуля преобразования информации

Таблицы прошедшие токенизацию, поступают на вход подмодуля выполняющего стемминг, в задачу этого подмодуля входит приведение слов к единому падежу и основе, чтобы не было проблем с дублированием информации.

Таблицы прошедшие стемминг, поступают в подмодуль, который перемещает, удаляет и переставляет столбцы в таблице.

Последним этапом преобразования информации является приведение таблицы к единой форме, ведь если объединить таблицы без этого пункта, то в сводной таблице появятся множество фактически одинаковых столбцов, с немного разными названиями.

В следующем модуле, пользователю выбирает сколько направлений подготовки/специальностей ему нужно отобразить, обработанные таблицы сортируются и объединяются в одну сводную таблицу, которая передается в модуль, осуществляющий группировку и кластеризацию.

В модуле кластеризации пользователь должен явно указать, какую информацию он хочет сгруппировать и по какому признаку. После указания всех настроек таблица принимает конечный вид и отображается пользователю.

Результаты таблицы используются модулем графического отображения, для составления диаграмм, а пользователь выбирает, какие диаграммы составлять, а какие нет.

Таким образом на выходе программы выводится таблица с результатами, и необходимые для пользователя диаграммы.

3.2 Описание функций программы

Перед проектированием структуры программы, следует описать функции которые программа будет выполнять.

Чтобы обработать большие данные, необходимо накопить достаточно количество источников информации. Это одна из наиболее сложных задач, потому что необходимо соблюдать баланс между количеством и качеством информации. Накопление бесполезной информации повлечет за собой только лишние вычислительные затраты на её сортировку, и хранение. Таким образом программа должна обеспечивать:

а) Получение информации из распределенных источников.

б) Обработку текстовой информации, которая включает в себя:

1) разбиение предложений на отдельные слова;

2) изменение регистра и удаление лишних символов и цифр, если такие присутствуют;

3) приведение слов к основе, на основе стеммера. Для того чтобы в последующем удалять лишнюю информацию, при сортировке;

4) преобразование одних типов данных в другие.

в) Сортировку данных и удаление ненужной информации (Например: строки, которые полностью нулевые или имеют тип «Nan»).

г) Приведение различных таблиц к единому виду, для того чтобы обеспечить возможность анализа данных.

д) Создание полной таблицы, которая включает в себя данные со всех источников.

е) Выполнение кластеризации по какому-либо признаку.

ж) Отображение полученных результатов о количестве обучающихся в виде таблицы.

и) Визуализация результатов на диаграммах и графиках.

к) Возможность выбора количества элементов на графике, поскольку при отображении всей информации графики будут сложно поддаваться анализу.

3.3 Получение и хранение информации

Перед обработкой информации необходимо определить, где можно получить эту информацию, и хранить её. Поскольку программный продукт должен рассчитывать численность обучающихся студентов, то объектом исследования становятся сайты вузов.

В соответствии с пунктом «В» приказа Федеральной службы по надзору в сфере образования и науки от 14 августа 2020 года N 831 «Об утверждении Требований к структуре официального сайта образовательной организации в информационно-телекоммуникационной сети "Интернет" и формату представления информации», в подразделе «Образование» на сайте вуза необходимо вести учет численности обучающихся по реализуемым образовательным программам. Поскольку эта информация находиться в открытом доступе, ничего не мешает использовать её для проведения статистических исследований и анализа.

Существует несколько путей получения этой информации, а именно:

а) Пользователь вручную отбирает таблицы на сайтах, сохраняет их локально, а затем обрабатывает при помощи программы.

б) Пользователь вручную отбирает таблицы, распределяет их между машинами, производит обработку, а затем складывает полученные результаты.

в) Пользователь выгружает таблицы автоматизировано, но это усложняет задачу, потому что необходимо писать парсер, который будет перебирать на сайте кучу таблиц, а для такого перебора по ссылкам, нужно делать копию сайта

и сохранять локально. А после осуществлять поиск и преобразование таблиц. Такой подход не верен так как предполагается, что сайтов будет достаточно много и при успешной работе программы количество сайтов планируется расширять.

г) Создать адаптивный парсер, который проверяет определенное местоположение. В том случае если он не находит таблицу в том пути, который ему задали, выбирает другой путь, так же заранее заданный. Для этого способа получения информации нужно знать определенный пул адресов, на которых может быть расположена информация.

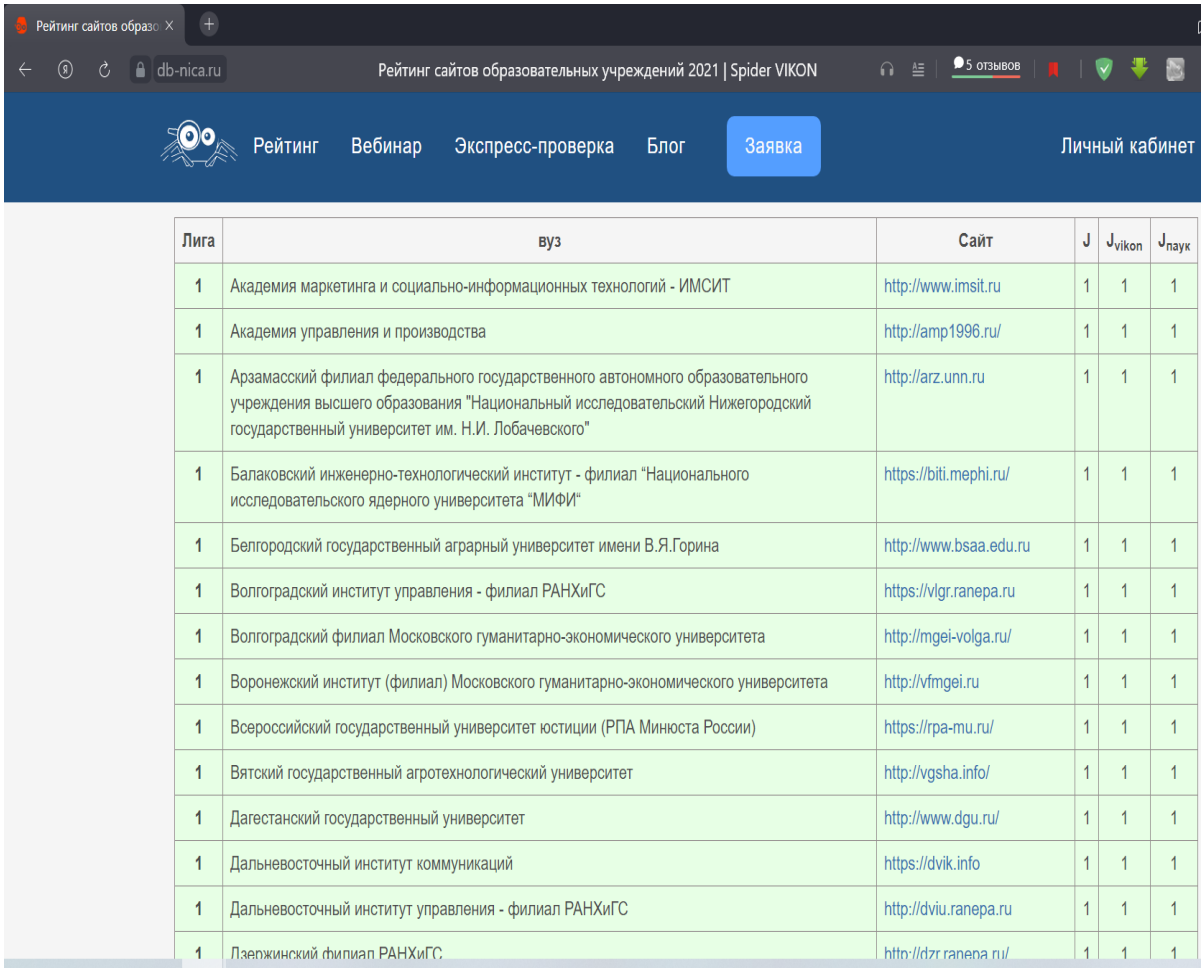
Самый подходящий способ «г», но для его реализации должна быть достаточно устойчивая структура сайта. Поскольку в вышеупомянутом приказе описана информация, которая, должна быть расположена на сайте, то можно сделать вывод, что этот контент контролируется Федеральной службой по надзору в сфере образования и науки.

Помимо этого, существуют разработчики создающие шаблоны сайтов, которые соответствуют необходимым требования. Наиболее популярным разработчиком личных кабинетов для сайтов вузов является «Национальный фонд поддержки инноваций в сфере образования». Они предлагают программное решение «Vikon», предназначенное для интеграции и самопроверки информации об образовательной организации на официальном сайте образовательных учреждений, в разделе «Сведения об образовательной организации».

На сайте «db-nica.ru», создан рейтинг с оценками, показывающими насколько сайт совпадает с требованиями. Данный рейтинг численно отражает насколько сайт, соответствует требованиям, и разделяет сайты по лигам соответствия. Всего на текущий момент в рейтинге представлено 287 учебных организаций. Так, например сайт амурского государственного университета находиться только в восьмой лиге на сайте.

На текущий момент, необязательно проверять работу программы на большом количестве информации, для работы программы следует выбрать разные виды и структуры сайтов, для создания парсеров. Это позволит показать возможности программы наиболее полно.

Таблица с рейтингом сайтов представлена на рисунке 24.



Лига	ВУЗ	Сайт	J	J_{vikon}	$J_{наук}$
1	Академия маркетинга и социально-информационных технологий - ИМСИТ	http://www.imsit.ru	1	1	1
1	Академия управления и производства	http://amp1996.ru/	1	1	1
1	Арзамасский филиал федерального государственного автономного образовательного учреждения высшего образования "Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского"	http://arz.unn.ru	1	1	1
1	Балаковский инженерно-технологический институт - филиал "Национального исследовательского ядерного университета "МИФИ"	https://biti.mephi.ru/	1	1	1
1	Белгородский государственный аграрный университет имени В.Я.Горина	http://www.bsaa.edu.ru	1	1	1
1	Волгоградский институт управления - филиал РАНХиГС	https://vlgr.ranepa.ru	1	1	1
1	Волгоградский филиал Московского гуманитарно-экономического университета	http://mgei-volga.ru/	1	1	1
1	Воронежский институт (филиал) Московского гуманитарно-экономического университета	http://vfmgei.ru	1	1	1
1	Всероссийский государственный университет юстиции (РПА Минюста России)	https://rpa-mu.ru/	1	1	1
1	Вятский государственный агротехнологический университет	http://vgsha.info/	1	1	1
1	Дагестанский государственный университет	http://www.dgu.ru/	1	1	1
1	Дальневосточный институт коммуникаций	https://dvik.info	1	1	1
1	Дальневосточный институт управления - филиал РАНХиГС	http://dviu.ranepa.ru	1	1	1
1	Дзержинский филиал РАНХиГС	http://dzr.ranepa.ru/	1	1	1

Рисунок 24 – Рейтинг соответствия с сайта db-nica.ru

С этого рейтинга берется массив сайтов, для решения поставленной задачи. Выбраны сайты из разных лиг и разного вида сложности, для оценки правильности работы, на последующих этапах разработки.

Для решения проблемы с хранением информации, было решено парсить данные, это позволяет не только решить проблему, но и получить преимущество в долгосрочной перспективе использования программы. Если данные будут изменяться с течением времени, а они обязаны изменяться, ведь каждый год вузы добавляют новые учебные направления, и каждый год меняется количество студентов, поступивших на те или иные направления, то парсить данные более удобно, пользователю не нужно будет собирать таблицы вручную. Если пользователю необходимо получить актуальные данные, то он может просто запустить программу и она в режиме реального времени соберет информацию.

Поскольку было решено писать программный продукт на языке «Python», необходимо выбрать какой из парсеров следует использовать. Наиболее популярные парсеры на данный момент это «BeautifulSoup» и несколько инструментов из библиотеки «Pandas».

Для того чтобы парсить данные с помощью BeautifulSoup необходимо осуществлять потэговый анализ веб-сайта и циклом проходить по всему коду страницы. Это усложнит задачу поскольку разные сайты могут использовать различные теги для при создании.

Гораздо удобнее для чтения данных использовать функцию read_html библиотеки pandas. Эта функция предназначена для преобразования из формата веб-страниц, в формат dataframe – специальной структуры данных, предназначенной для хранения информации. Изначально dataframe создавался для хранения данных в табличном виде, поэтому использовать эту возможность гораздо удобнее, чем перебирать теги и формировать таблицу исходя из них. Кроме того, многие сайты используют html5, и, для того чтобы предотвратить возможные ошибки доступа и чтения модулей, следует использовать библиотеку html5lib.

Обращение к таблице происходит по её индексу, либо с помощью выбора фразы, которая точно присутствует в этой таблице, но её нет в других таблицах на странице. При чтении можно задать такие параметры как индекс таблицы, её шапку и множество других. После чтения обязательно указывать «[0]» для dataframe, чтобы отобразить его как таблицу.

3.4 Обработка информации

После того как был составлен пул адресов, где находятся таблицы. Все таблицы были помещены в список. Необходимо при помощи цикла брать адреса из списка и выбирать для него один из нескольких написанных парсеров. Наличие нескольких парсеров для работы программы обязательно, поскольку структура сайтов может отличаться друг от друга, как минимум из-за человеческого фактора при создании.

Существует несколько вариантов парсеров, для разного вида таблиц. На текущий момент определено три основных вида таблиц:

- таблица с мультииндексом в шапке, с семью столбцами. Такая таблица не учитывает количество иностранных граждан, общее количество, и в ней код и наименование разделены. Пример такой таблицы представлен на рисунке 25;

Информация о численности обучающихся по реализуемым образовательным программам за счет бюджетных ассигнований федерального бюджета, бюджетов субъектов Российской Федерации, местных бюджетов и по договорам об образовании за счет средств физических и (или) юридических лиц

Код	Наименование специальности, направления подготовки	Уровень образования	Форма обучения	Численность обучающихся за счет (количество человек):			
				бюджетных ассигнований федерального бюджета	бюджетов субъектов Российской Федерации	местных бюджетов	средств физических и (или) юридических лиц
-	Русская классическая гимназия	начальное общее образование	очная	-	-	-	91
-	Русская классическая гимназия	основное общее образование	очная	-	-	-	16
40.02.01	Право и организация социального обеспечения	среднее профессиональное образование	очная	66	-	-	459
40.03.01	Юриспруденция	высшее образование – бакалавриат	очная	243	-	-	182
40.03.01	Юриспруденция	высшее образование – бакалавриат	очно-заочная	54	-	-	58
40.03.01	Юриспруденция	высшее образование – бакалавриат	заочная	17	-	-	55
40.05.01	Правовое обеспечение национальной безопасности	высшее образование – специалитет	очная	98	-	-	301
40.05.01	Правовое обеспечение национальной безопасности	высшее образование – специалитет	очно-заочная	12	-	-	57
40.05.01	Правовое обеспечение национальной безопасности	высшее образование – специалитет	заочная	55	-	-	241
40.05.02	Правоохранительная деятельность	высшее образование – специалитет	очная	43	-	-	107
40.05.02	Правоохранительная деятельность	высшее образование – специалитет	очно-заочная	5	-	-	1
40.05.02	Правоохранительная деятельность	высшее образование – специалитет	заочная	12	-	-	35
40.05.04	Судебная и прокурорская деятельность	высшее образование – специалитет	очная	5	-	-	41

Рисунок 25 – Пример таблицы вуза ВГУЮ

- таблица с четырнадцатью столбцами и тремя уровнями в мультииндексе и учетом количества иностранных граждан. Такая таблица наиболее полная и развернутая. Пример таблицы представлен на рисунке 26;

№	Код	Наименование специальности, направления подготовки	Уровень образования	Форма обучения	Общая численность обучающихся	Численность обучающихся за счет (количество человек):							
						бюджетных ассигнований федерального бюджета		бюджетов субъектов Российской Федерации		местных бюджетов		средств физических и (или) юридических лиц	
						Всего	в том числе обучающихся, являющихся иностранными гражданами	Всего	в том числе обучающихся, являющихся иностранными гражданами	Всего	в том числе обучающихся, являющихся иностранными гражданами	Всего	в том числе обучающихся, являющихся иностранными гражданами
1	38.02.01	Экономика и бухгалтерский учет (по отраслям)	Среднее профессиональное образование - Программа подготовки специалистов среднего звена	Очная	14	0	0	0	0	0	0	14	0
				Заочная	19	0	0	0	0	0	0	19	0
				Очно-заочная	-	-	-	-	-	-	-	-	-
2	38.02.02	Страховое дело (по отраслям)	Среднее профессиональное образование - Программа подготовки специалистов среднего звена	Очная	0	0	0	0	0	0	0	0	0
				Заочная	-	-	-	-	-	-	-	-	-
				Очно-заочная	-	-	-	-	-	-	-	-	-
3	38.02.04	Коммерция (по отраслям)	Среднее профессиональное образование - Программа подготовки специалистов среднего звена	Очная	0	0	0	0	0	0	0	0	0
				Заочная	-	-	-	-	-	-	-	-	-
				Очно-заочная	-	-	-	-	-	-	-	-	-
4	38.02.06	Финансы	Среднее профессиональное образование - Программа подготовки специалистов среднего звена	Очная	27	0	0	0	0	0	0	27	1
				Заочная	-	-	-	-	-	-	-	-	-
				Очно-заочная	-	-	-	-	-	-	-	-	-

Рисунок 26 – Пример таблицы вуза ВГАТУ

- таблица с семью столбцами, с мультииндексом и объединенным кодом направления и наименованием специальности/направления подготовки. Пример таблицы представлен на рисунке 27;

Наименование специальности/ направления подготовки	Форма обучения	Численность обучающихся, чел.				Численность обучающихся, являющихся иностранными гражданами
		за счёт бюджетных ассигнований федерального бюджета	за счёт бюджетов субъектов Российской Федерации	за счёт местных бюджетов	за счёт средств физических и (или) юридических лиц	
Высшее образование - Бакалавриат						
01.03.02 Прикладная математика и информатика	Очная	26	0	0	0	0
	Заочная	0	0	0	0	0
	Очно-заочная	0	0	0	0	0
03.03.02 Физика	Очная	49	0	0	0	0
	Заочная	0	0	0	0	0
	Очно-заочная	0	0	0	0	0
09.03.01 Информатика и вычислительная техника	Очная	75	0	0	1	0
	Заочная	0	0	0	0	0
	Очно-заочная	0	0	0	0	0
09.03.02 Информационные системы и технологии	Очная	94	0	0	3	0
	Заочная	0	0	0	0	0
	Очно-заочная	0	0	0	0	0
09.03.04 Программная инженерия	Очная	0	0	0	0	0

Рисунок 27 – «Пример таблицы вуза АмГУ»

Из всех представленных таблиц наиболее правильно оформлена таблица на рисунке 26, но нельзя точно быть уверенным, что при парсинге все таблицы будут такими.

Из представленных примеров видно, что существуют различные примеры оформления таблиц и обработка их всех в таком виде сразу невозможна. Для того, чтобы получить эти таблицы, нужно написать разные парсеры и выбирать один из этих парсеров для каждого конкретного случая.

Обычно ссылка на таблицу на сайтах располагается по двум следующим путям:

а) «https://доменное имя/sveden/education»;

б) «https://доменное имя/sveden/education/study/».

На пути «а», обычно располагаются таблицы, состоящие из семи колонок. Путь «б» включает в себя таблицы из четырнадцати колонок, но бывают исключения, например сайт «АмГУ». Это необходимо учитывать при написании парсеров.

После парсинга часто возникают ситуации, когда появляются лишние столбцы, происходит сдвиг строк и появляются другие изменения, усложняющие анализ. Пример такой ситуации, возникающей при парсинге с таблицей, рассмотренной на рисунке 26, показан на рисунке 28.

In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import html5lib
import warnings
import matplotlib.pyplot as plt

warnings.filterwarnings('ignore')
pd.options.display.max_rows = 1000
pd.options.display.max_colwidth = 1000
pd.options.display.max_columns = 1000

url = 'https://vgsha.info/sveden/education/study/'
df = pd.read_html(url,index_col=0)[0]
df
```

Out[2]:

№	Код	Наименование специальности, направления подготовки	Уровень образования	Форма обучения	Общая численность обучающихся	Численность обучающихся за счет (количество человек):			
						бюджетных ассигнований федерального бюджета		бюджетов субъектов Российской Федерации	
						Всего	в том числе обучающихся, являющихся иностранными гражданами	Всего	в том числе обучающихся, являющихся иностранными гражданами
1	38.02.01	Экономика и бухгалтерский учет (по отраслям)	Среднее профессиональное образование - Программа подготовки специалистов среднего звена	Очная	14	0	0	0	0
1	38.02.01	Экономика и бухгалтерский учет (по отраслям)	Среднее профессиональное образование - Программа подготовки специалистов среднего звена	38.02.01	Экономика и бухгалтерский учет (по отраслям)	Среднее профессиональное образование - Программа подготовки специалистов среднего звена	Заочная	19	0
1	38.02.01	Экономика и бухгалтерский учет (по отраслям)	Среднее профессиональное образование - Программа подготовки специалистов среднего звена	38.02.01	Экономика и бухгалтерский учет (по отраслям)	Среднее профессиональное образование - Программа подготовки специалистов среднего звена	Очно-заочная	-	-
2	38.02.02	Страховое дело (по отраслям)	Среднее профессиональное образование - Программа подготовки специалистов среднего звена	Очная	0	0	0	0	0
2	38.02.02	Страховое дело (по отраслям)	Среднее профессиональное образование - Программа подготовки специалистов среднего звена	38.02.02	Страховое дело (по отраслям)	Среднее профессиональное образование - Программа подготовки специалистов среднего звена	Заочная	-	-

Рисунок 28 – Парсинг таблицы вуза ВГАТУ

Как видно из рисунка 28, происходит дублирование данных в шапке, а также происходит дублирование специальностей/направлений подготовки очной и очно-заочной формы обучения, из-за этого добавляются два новых столбца, в конце с названием «Unnamed».

Это происходит из-за того, что при создании таблицы на сайте, разработчик помимо указания атрибута «rowspan», отвечающего за объединение ячеек, продублировал ту информацию, которая объединена в тегах.

Для того, чтобы избавиться от дублирования, следует осуществить проверку соответствия содержимого столбца, его типу. Таким образом проверяя столбец «Форма обучения», в нем должны содержаться только элементы текстового типа, а в последующих столбцах, где происходит дублирование должен быть тип «numeric». После преобразования этой информации, тип несовпадающих данных изменяется на «Nan».

Просто удалить эти столбцы не получиться т. к. в них есть информация об очной форме обучения. Чтобы избавиться от этого, необходимо все полученные Nan элементы переместить в конец строк, тем самым они окажутся в столбцах «Unnamed», и таблица примет не искаженный вид. После выполнения этой операции можно удалить все столбцы «Unnamed», тем самым порядок строк вернется к виду, представленному на сайте.

Далее необходимо удалить мультииндекс из шапки таблицы, потому что при создании общей таблицы будут возникать проблемы при объединении, и мультииндекс затруднит обращение к столбцам.

Для удаления мультииндекса нужно определить условие, при котором он возникает. В некоторых случаях, следует просто проверить есть ли во втором уровне мультииндекса название «Численность обучающихся» и если есть, то переместить, либо удалить этот уровень. Этот подход больше подходит для случая, когда уровень мультииндекса не превышает двух.

В том случае если уровень мультииндекса больше двух, можно просто использовать функцию «len(df.columns.levels)» для получения числа уровней и в том случае если он превышает определенное число отбросить нужные уровни.

Получив таблицы с одним уровнем в шапке, нужно привести их к одной, наиболее удобной и понятной форме для пользователя. Поскольку столбцы «Код» и «Наименование специальности» зачастую используются вместе, следует их объединить, это более удобно и просто чем разделять уже объединенные

столбцы. Логически код специальности/направления подготовки несет ту же нагрузку, что и наименование, а увеличение количества столбцов только усложнит работу парсеров.

Убедиться, в том, что поля ещё не объединены (такое используется, например на сайте АмГУ), возможно, создав условие для проверки имен всех столбцов, на наличие в них слова «Код» и различных её вариаций, после того как был удален мультииндекс эта задача не составляет большого труда. Элементы объединяемых столбцов сохраняются в переменные. Выполняется их суммирование и добавление между ними пробела. Затем создается новый столбец в датафрейме и в него построчно записываются, полученные элементы. Старые столбцы до объединения удаляется, а новый столбец помещается в датафрейм при помощи команды insert, с указанием его индекса, в данном случае индекс этого столбца будет равен нулю.

В том случае, если в таблице присутствует столбец «Общая численность обучающихся», его необходимо удалить, поскольку в этом столбце не учтены иностранные граждане. Можно было бы оставить этот столбец, производить выборку данных, об иностранных гражданах и затем складывать её с ним, но это может исказить расчеты. Поэтому удаление этого столбца необходимо, в том случае если он присутствует. Поиск такого столбца осуществляется таким же образом, что и поиск столбца «Код», описанный ранее.

Из рассмотренных примеров можно заметить, что даже если происходит удаление верхних уровней шапки столбцы искажаются и в определенных случаях на выходе получиться куча столбцов с названиями «Всего» и «В том числе обучающихся иностранных граждан», это показано на рисунке 29.

Имя	Уровень образования	Форма обучения	Всего	в том числе обучающихся, являющихся иностранными гражданами	Всего.1	в том числе обучающихся, являющихся иностранными гражданами.1	Всего.2	в том числе обучающихся, являющихся иностранными гражданами.2	Всего.3	в том числе обучающихся, являющихся иностранными гражданами.3	Сумма	Всего.4	в том числе обучающихся, являющихся иностранными гражданами.4
ценция	Высшее образование - Бакалавриат	Очная	0.0	0.0	0.0	0.0	0.0	0.0	112.0	9.0	121.0	NaN	NaN
ценция	Высшее образование - Бакалавриат	Заочная	0.0	0.0	0.0	0.0	0.0	0.0	32.0	0.0	32.0	NaN	NaN
ценция	Высшее образование - Бакалавриат	Очно-заочная	0.0	0.0	0.0	0.0	0.0	0.0	436.0	0.0	436.0	NaN	NaN
номика	Высшее образование -	Очная	0.0	0.0	0.0	0.0	0.0	0.0	66.0	9.0	75.0	NaN	NaN

Рисунок 29 – Дублирование названий при удалении уровней мультииндекса

Это искажает логический смысл этих столбцов, для того чтобы пользователь мог понимать то, о чем идет речь в этих столбцах, их следует привести к одному, утвержденному виду. Так, например можно переименовать первый столбец «Всего» в «Бюджетных ассигнований федерального бюджета», а столбец «В том числе обучающихся иностранных граждан» в «Бюджетных ассигнований федерального бюджета (Иностранцы)». И сделать эту процедуру со всеми столбцами. В некоторых таблицах встречаются столбцы, в которых в названиях добавляют словосочетание «За счёт», они имеют тот же смысл, но если в будущем их объединять, то будут созданы обе колонки «Бюджетных ассигнований федерального бюджета» и «За счет бюджетных ассигнований федерального бюджета». Один из этих столбцов будет наполнен данными только того сайта, с которого взята таблица, а все остальные данные по другим направлениям подготовки будет иметь тип «Nan». Пример такого случая показан на рисунке 30.

ёт ых ий го та	за счёт бюджетов субъектов Российской Федерации	за счёт местных бюджетов	за счёт средств физических и (или) юридических лиц	Численность обучающихся, являющихся иностранными гражданами	бюджетных ассигнований федерального бюджета	бюджетов субъектов Российской Федерации	местных бюджетов	средств физических и (или) юридических лиц	за счет бюджетных ассигнований федерального бюджета	за счет бюджетов субъектов Российской Федерации	за счет местных бюджетов	за счет средств физических и (или) юридических лиц
aN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
aN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
aN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
aN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Рисунок 30 – Искажение информации без переименования столбцов

Переименовав столбцы к единому формату, происходит избавление от недостатка, представленного на рисунке 30.

Для того чтобы получить сумму по строке для определенного направления подготовки/специальности осуществим преобразование элементов столбцов с информацией о количестве (т.е все столбцы кроме «№», «Имя», «Форма обучение», «Уровень образования») в числовой формат.

После преобразования, столбцы начиная с третьего (не учитываем столбец с номером строки) можно суммировать и записать их в новый созданный столбец

датафрейма с названием «Сумма». Таким образом получаются данные о количестве обучающихся по направлениям подготовки/специальностям в одном вузе.

Когда таблица приведена к единому формату, необходимо сделать так, чтобы ссылки на сайты автоматически подставлялись, а затем обрабатывались тем парсером, который написан для таблицы на этом сайте, это определяется из таких параметров, как количество столбцов, название столбцов, индекс таблицы на сайте и прочие. Обработанная таблица должна быть объединена с остальными обработанными таблицами.

Для того чтобы это сделать, в начале программы, после определения таких параметров, как отображение столбцов, их длина, реакция на ошибки и импорт библиотек, создаются два элемента «list», в первый помещаются ссылки на сайты, а второй остается пустым.

Заполнив список создается цикл, в котором перебираются все элементы списка до последнего, сначала происходит определение того, как заканчивается путь в сайте, затем определение длинны уровней мультииндекса и уже после этого длинны столбцов и названия столбцов. В зависимости от этих условий для каждой из используемых ссылок подбирается свой парсер.

После того как происходят преобразования таблицы и формирование столбца «сумма», выбираются те элементы сумма значений, которых равна нулю. Это может быть информация о форме обучения которая производится, но на ней нет обучаемых студентов (достаточно часто очно-заочная форма обучения), или лишняя информация (Например подзаголовки, либо заметки).

После такой обработки, каждая полученная таблица, в каждом парсере, при помощи функции «append» добавляется в пустой список, созданный в начале программы.

Таким образом пустой список наполняется всеми необходимыми данными из таблиц и становиться достаточно большим, после окончания работы всех парсеров при помощи функции concat «склеиваем» таблицы, помещенные в список, и передаем это в новый датафрейм, чтобы снова выполнять с ним операции.

После получения датафрейма с объединённой таблицей происходит упорядочивание столбцов таким образом, чтобы таблица была легко читаемой пользователем. Происходит сортировка строк столбца «Сумма» по убыванию, чтобы в последующем можно было напрямую обращаться к первым, или последним элементам таблицы, при построении графиков, а не писать функцию, которая будет обрабатывать достаточно большое количество строк.

Далее происходит сброс столбца с индексом, поскольку после сортировки, индексы перемешались и не упорядочены. Сброс происходит при помощи функции reset_index. После обработки общая таблица выглядит следующим образом (рисунок 31).

	Имя	Уровень образования	Форма обучения	Сумма	бюджетных ассигнований федерального бюджета	бюджетных ассигнований федерального бюджета (Иностранцы)	бюджетов субъектов Российской Федерации	бюджетов субъектов Российской Федерации (Иностранцы)	местных бюджетов	местных бюджетов (Иностранцы)	с
0	27.03.03 Системный анализ и управление	Высшее образование - Бакалавриат	Очная	5743.0	5716.0	17.0	0.0	0.0	9.0	1.0	
1	31.05.01 Лечебное дело	Высшее образование - Специалитет	Очная	1209.0	81.0	0.0	0.0	0.0	1032.0	96.0	
2	11.03.04 Электроника и наноэлектроника	Высшее образование - Бакалавриат	Очная	903.0	736.0	139.0	0.0	0.0	23.0	5.0	
3	40.03.01 Юриспруденция	Высшее образование - Бакалавриат	Очно-заочная	779.0	0.0	0.0	0.0	0.0	716.0	63.0	
4	27.03.04 Управление в технических системах	Высшее образование - Бакалавриат	Очная	775.0	485.0	185.0	0.0	0.0	72.0	33.0	
5	35.03.06 Агроинженерия	Высшее образование - Бакалавриат	Очная	747.0	455.0	288.0	0.0	0.0	2.0	2.0	
6	31.05.03 Стоматология	Высшее образование - Специалитет	Очная	726.0	30.0	0.0	0.0	0.0	614.0	82.0	
7	35.03.06 Агроинженерия	Высшее образование - Бакалавриат	Заочная	702.0	235.0	2.0	0.0	0.0	455.0	10.0	
8	09.03.01 Информатика и вычислительная техника	Высшее образование - Бакалавриат	Очная	676.0	476.0	29.0	0.0	0.0	168.0	3.0	
9	13.03.02 Электроэнергетика и электротехника	Высшее образование - Бакалавриат	Очная	610.0	414.0	145.0	0.0	0.0	44.0	7.0	
10	53.05.01 Искусство концертного исполнительства...	Высшее образование - Специалитет	Очная	604.0	526.0	0.0	0.0	0.0	78.0	0.0	
11	не предусмотрен Обеспечение экологической безо...	Дополнительное профессиональное образование - ...	Очная	603.0	0.0	0.0	0.0	0.0	603.0	0.0	
12	36.05.01 Ветеринария	Высшее образование - Специалитет	Очная	592.0	471.0	24.0	0.0	0.0	79.0	18.0	
13	09.03.02 Информационные системы и технологии	Высшее образование - Бакалавриат	Очная	591.0	374.0	23.0	0.0	0.0	188.0	6.0	

Рисунок 31 – Сводная таблица по численности обучающихся

Если пользователю необходима таблица с количеством обучающихся по всем формам обучения для конкретного направления подготовки/специальности, то после можно запустить сумму строк таблицы с группировкой по имени.

Например, если пользователю нужно получить число человек направления подготовки «27.03.03 Системный анализ и управление», для форм обучения «Очная», «Заочная», «Очно заочная» то он может использовать эту группировку. Пример сгруппированной таблицы представлен на рисунке 32.

	Имя	Сумма	бюджетных ассигнований федерального бюджета	бюджетных ассигнований федерального бюджета (Иностранцы)	бюджетов субъектов Российской Федерации	бюджетов субъектов Российской Федерации (Иностранцы)	местных бюджетов	местных бюджетов (Иностранцы)	иных средств	иных средств (Иностранцы)	ср физич и юридич
0	27.03.03 системный анализ и управление	5743.0	5716.0	17.0	0.0	0.0	9.0	1.0	0.0	0.0	
1	35.03.06 агроинженерия	3752.0	2610.0	325.0	0.0	0.0	761.0	56.0	0.0	0.0	
2	40.03.01 юриспруденция	3719.0	360.0	4.0	0.0	0.0	2615.0	125.0	0.0	0.0	
3	38.03.01 экономика	3109.0	90.0	10.0	0.0	0.0	2600.0	161.0	0.0	0.0	
4	36.05.01 ветеринария	2199.0	1531.0	69.0	0.0	0.0	523.0	76.0	0.0	0.0	
5	13.03.02 электроэнергетика и электротехника	1824.0	1054.0	160.0	0.0	0.0	250.0	37.0	0.0	0.0	
6	38.03.02 менеджмент	1662.0	48.0	3.0	0.0	0.0	1493.0	85.0	0.0	0.0	
7	31.05.01 лечебное дело	1552.0	81.0	0.0	0.0	0.0	1357.0	114.0	0.0	0.0	
8	31.05.03 стоматология	1291.0	30.0	0.0	0.0	0.0	1131.0	130.0	0.0	0.0	
9	35.03.04 агрономия	1257.0	883.0	104.0	0.0	0.0	196.0	74.0	0.0	0.0	
10	09.03.01 информатика и вычислительная техника	1094.0	652.0	32.0	0.0	0.0	402.0	7.0	0.0	0.0	
11	35.03.07 технология производства и переработки...	1058.0	743.0	24.0	0.0	0.0	288.0	3.0	0.0	0.0	
12	11.03.04 электроника и наноэлектроника	1002.0	769.0	141.0	0.0	0.0	84.0	8.0	0.0	0.0	
13	45.03.02 лингвистика	967.0	139.0	6.0	0.0	0.0	738.0	42.0	0.0	0.0	
14	09.03.02 информационные системы и технологии	952.0	558.0	26.0	0.0	0.0	317.0	6.0	0.0	0.0	
15	27.03.04 управление в технических системах	901.0	542.0	188.0	0.0	0.0	134.0	35.0	0.0	0.0	
16	42.03.01 реклама и связи с общественностью	833.0	155.0	7.0	0.0	0.0	633.0	30.0	0.0	0.0	

Рисунок 32 – Сводная таблица по численности обучающихся сгруппированная по столбцу «Имя»

Так же можно осуществить группировку по другим столбцам в зависимости от потребностей пользователя, но наиболее полезной является уже упомянутая группировка по имени.

При помощи функции nlargest, в последующем, можно указать какое количество строк необходимо вывести. Это полезно при построении графиков поскольку делает их более читаемыми, и отсеивает лишнюю информацию.

3.5 Создание и настройка кластера

Полагаясь на результаты работы, описанные в пункте 3.4, разработанный программный продукт выполняет свою задачу. Но будет ли выполнять про-

грамма свою функции за приемлемое время, при увеличении количества обрабатываемой информации (Например, если в программе будет обрабатываться не сотня ссылок, а десять тысяч). Эта проблема обычно игнорируется разработчиками, но для надлежащей работы программы она должна быть решена.

Именно для решения этой проблемы были рассмотрены такие фреймворки, как Apache Hadoop и Apache Spark. Ранее была рассмотрена возможность создания map reduce структуры для программы. Таким образом mapper бы преобразовывал каждую строчку исходного файла формата «.ру», а reducer бы собирал и считал то, что нам нужно. Обычно такой подход используется с локальными файлами т.е при запуске функции указываются mapper, reducer, исходный файл, папка с входными данными в среде HDFS, и выходная папка куда записывается результат.

Данный подход немного видоизменяется, применительно к разрабатываемой программе, поскольку исходных файлов у нас нет, а данные берутся сразу из глобальной сети интернет.

Перейдем к установке и созданию кластера для обработки программы. Данный этап был выполнен двумя методами, среди которых был выбран один для последующей работы.

Первый метод, это установка фреймворка Hadoop на локальную машину, и добавление в него других машин из сети, из всех добавленных в кластер машин могут быть использованы ресурсы для обработки.

Hadoop в первую очередь разработан под ОС Linux, но есть способы его установки и на ОС Windows. Для того чтобы фреймворк работал необходимо заранее установить Java SE и добавить её в переменные окружения Windows. После нужно скачать архив фреймворка с официального сайта «hadoop.apache.org», распаковать в удобную для пользователя директорию, добавит его в переменные окружения, и изменить файлы конфигурации «mapred-site.xml», «core-site.xml», «yarn-site.xml», «hdfs-site.xml», «hadoop-env.cmd» находящиеся в папке «*\etc\hadoop». Данные файлы отвечают за таки параметры как, отображение в веб-интерфейсе, создание учетной запись для пользователя, расположение hdfs в

локальной сети, расположение name node и data node на пк, и установка пути для java. Для примера покажем на рисунке 33 содержимое файла «hdfs-site.xml».

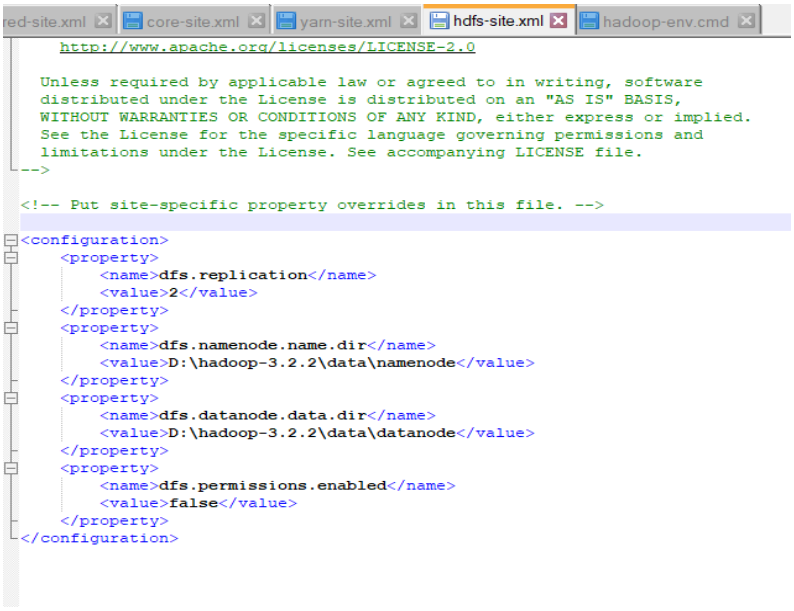


Рисунок 33 – Файл hdfs-site.xml

Как видно на рисунке 33, некоторые настройки зависят от того, в какую директорию у пользователя распакован Hadoop, поэтому настройка работы фреймворка может во многих случаях отличаться.

После того как все настройки выполнены пользователь может запустить все службы Hadoop, через командную строку. Для этого нужно перейти в папку «*\sbin» и выполнить команду «start-all.cmd». О правильном запуске служб фреймворка оповещают четыре командные строки показанные на рисунке 34.

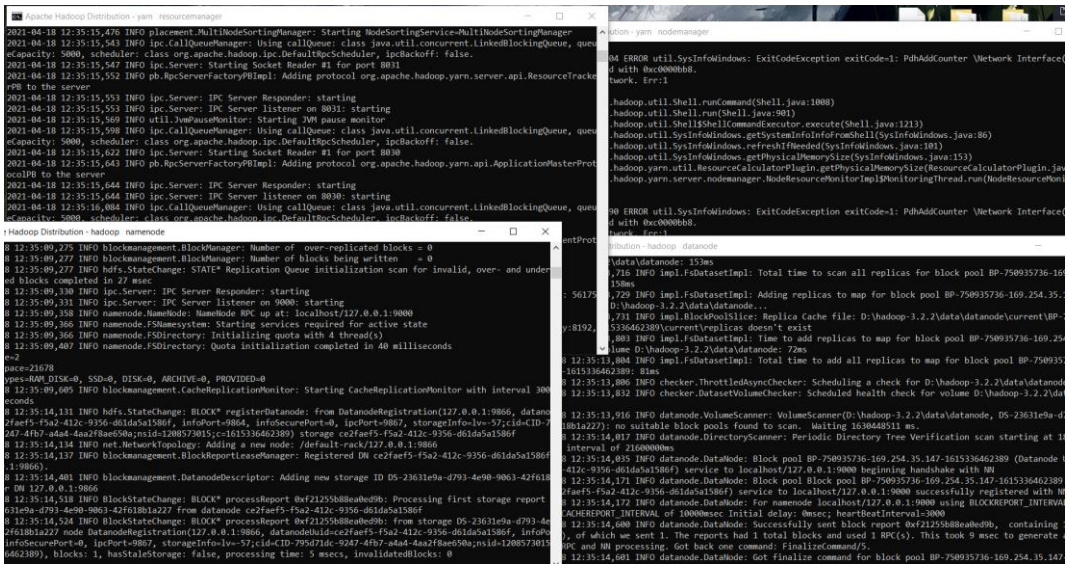


Рисунок 34 – Запуск hadoop

После успешного запуска фреймворка, можно войти в любой браузер и ввести в адресной строке – «localhost:9870», для того чтобы перейти к настройкам hdfs, или «localhost:8088», чтобы перейти к заданиям на кластере.

По адресу «localhost:9870», расположен основной объем функций, там показаны машины, которые введены в кластер, структура папок в hdfs откуда будут браться файлы, отчёты об ошибке и прочее, текущее состояние кластеров. На рисунке 35 представлена форма обзора кластера.

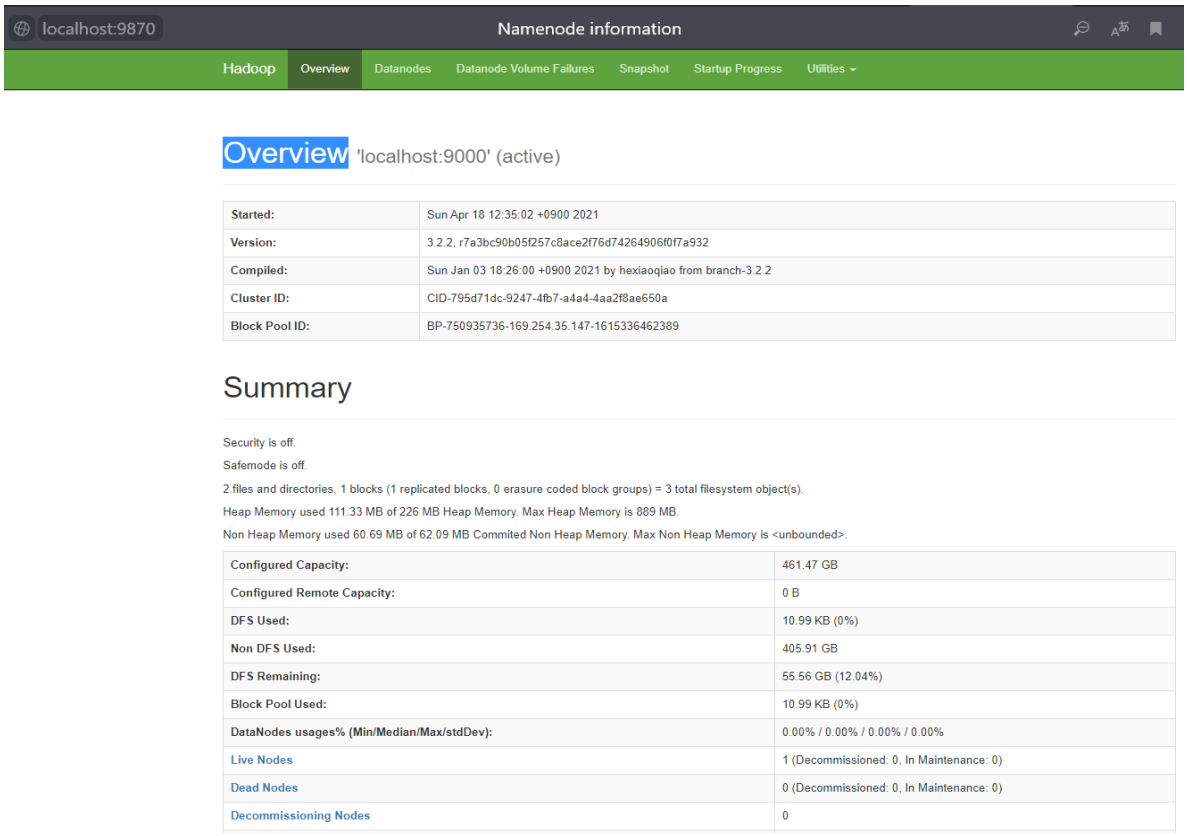


Рисунок 35 – Обзор кластера

Для того чтобы запустить программу на кластере, необходимо в командной строке (Желательно от имени администратора) выполнить команду «hadoop jar share/hadoop/tools/lib/hadoop-streaming-3.2.2.jar -files mapper.py, reducer.py \ -mapper "python mapper.py" \ -reducer "python reducer.py" \ -input /test2 \ -output /out \», в этой команде указаны папка с входными данными и выходными, и mapper и reducer.Обратите внимание что при записи в определенные папки могут потребоваться права администратора для кластера. Если кластер начал выполнять задание в консольном окне появится следующая информация (рисунок 36).

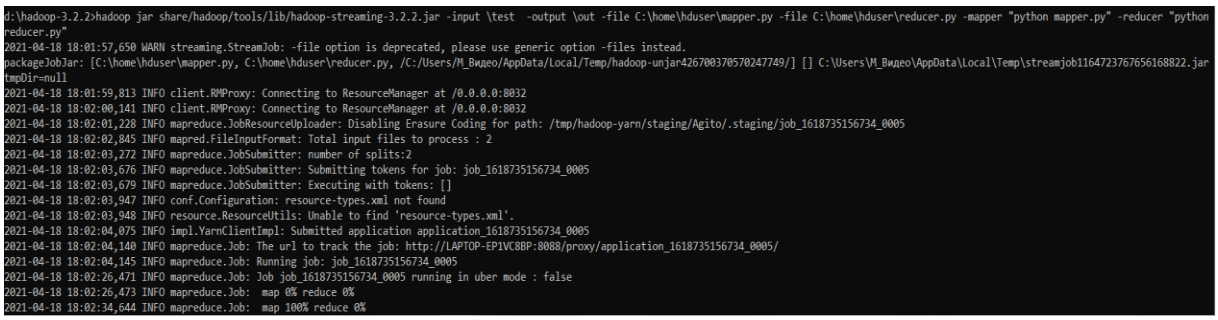


Рисунок 36 – Окно консоли при запуске задания на кластере

После успешного выполнения задания можно проверить состояние программы на кластере можно, перейдя в путь «http://localhost:8088» (рисунок 37).

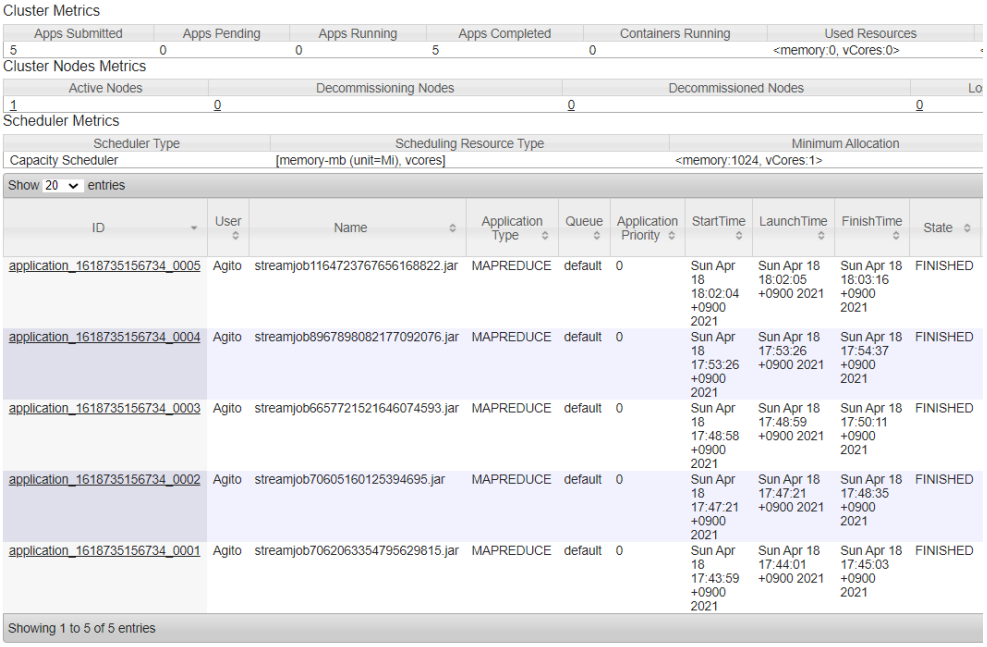


Рисунок 37 – Просмотр задания на кластере

Данные о результатах хранятся в той папке в hdfs, указанной при запуске задания.

Недостатком этого метода является тот факт, что не у всех пользователей есть множество компьютеров, для того чтобы выделить из них ресурсы для решения задачи, а в том случае если работа исполняется на одном компьютере, плюсом является только тот факт, что происходит разбиение задачи на подзадачи и программа выполняется параллельно, но в случае с относительно небольшим количеством обрабатываемой информации прирост производительности не особо заметен.

Второй метод устраняет недостаток первого, и этот метод — это «отечественный ответ» такому сервису как «Amazon Web Services», этот сервис называется «Yandex Cloud». Ещё несколько лет назад, этот сервис использовался просто как хранилище информации, но на сегодняшний день он разросся до огромных масштабов и предоставляет пользователю услуги по аренде вычислительных ресурсов, для обработки больших данных. Этот метод наиболее хорош своей простотой и понятность.

Воспользоваться данным сервисом с подвиг тот факт, что сейчас зарегистрировавшимся пользователям дают особую привилегию в виде ресурсов на четыре тысячи рублей, на два месяца бесплатно.

Чтобы воспользоваться этой услугой, пользователь регистрируется в сервисе Yandex Cloud (рисунок 38).

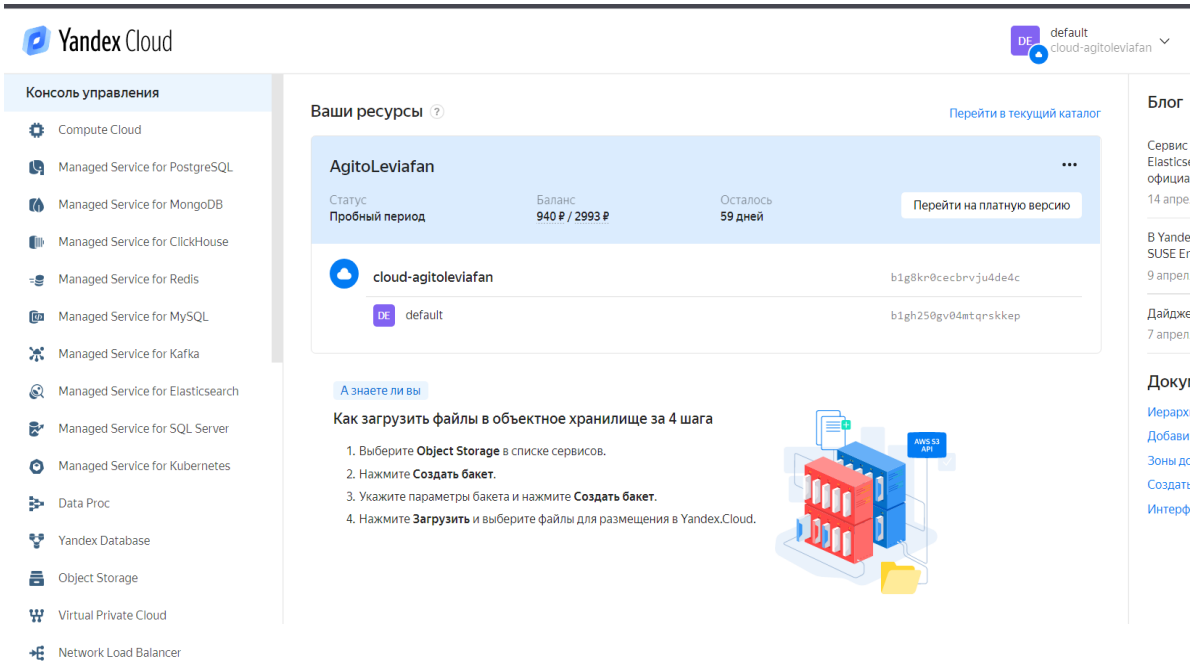


Рисунок 38 – Меню Yandex Cloud

Во время регистрации потребуется ввести SSH ключ, для того чтобы синхронизировать локальный компьютер с облаком. Для генерации SSH ключа была использована бесплатная программа «PuTTY key generator».

Первым делом после регистрации необходимо зарегистрировать сервисный аккаунт с ролями представленными на рисунке 39.

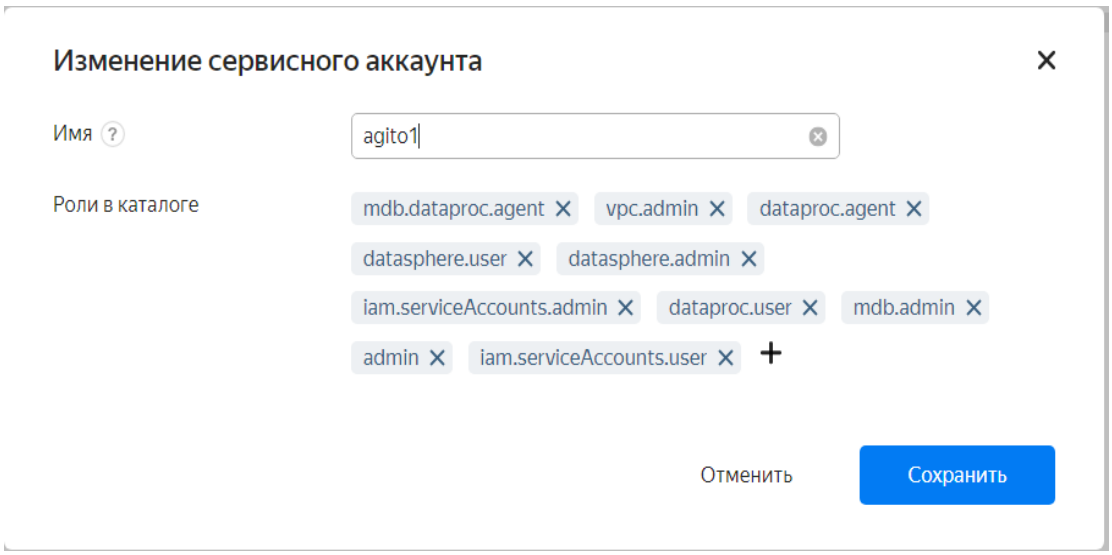


Рисунок 39 – Необходимые роли для сервисного аккаунта

После того как аккаунт создан, можно либо сразу создать кластер в меню «DataProc», либо создать проект в меню «DataSphere».

Сначала создадим новый проект, и зайдем в его настройки, укажем в нем созданный серверный аккаунт, и выделенную подсеть (Рисунок 40).

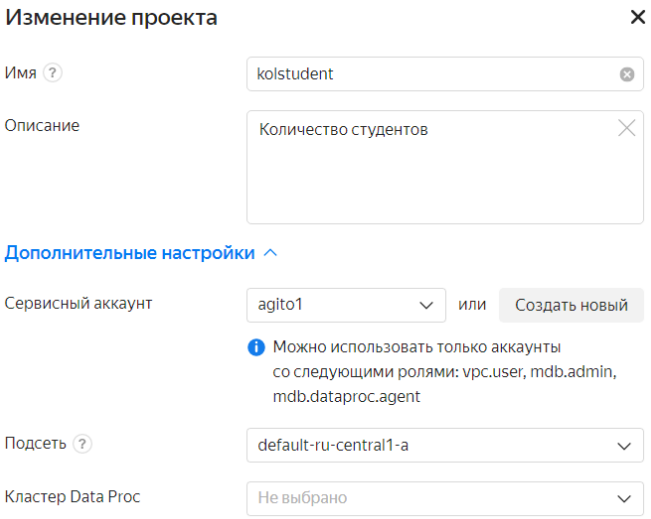


Рисунок 40 – Настройки проекта

Затем заходим в проект и загружаем в него ранее созданный файл с программой, это может быть как обычный файл с расширением «.py», так и файл созданный в Jupiter notebook с расширением «. ipynb».

Для запуска кластера необходимо сначала его создать, для этого нужно перейти по следующему пути «File-Data Proc Clusters» нажать кнопку и появиться окно (рисунок 41).

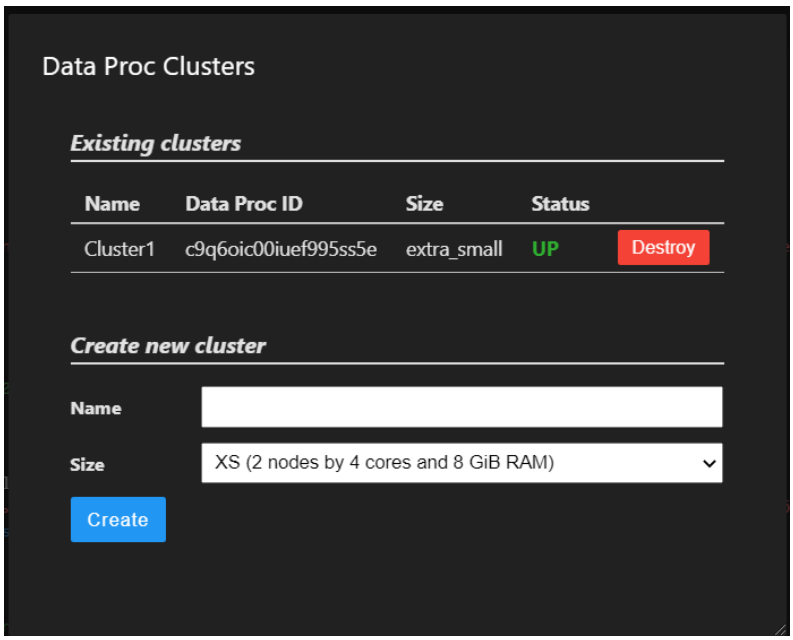


Рисунок 41 – Добавление кластера

В данном окне нужно просто ввести имя кластера и выбрать его размер в зависимости от ваших потребностей. После в коде программы необходимо добавить строчку «#! Spark --cluster <Cluster1>», где «Cluster1» это имя созданного кластера. Данная строка запускает проект на кластере.

Для функционирования созданной программы очень важно чтобы подсеть имела доступ в интернет, поскольку программа парсит данные с сайтов в режиме реального времени. Компанией Яндекс данная функция только внедряется, но если объяснить причину её использования, то можно получить Preview версию. Чтобы это сделать нужно перейти в настройки сети, выбрать там нужную подсеть, нажать на её настройки и выбрать «Включить Nat в интернет», данные настройки показаны на рисунке 42.

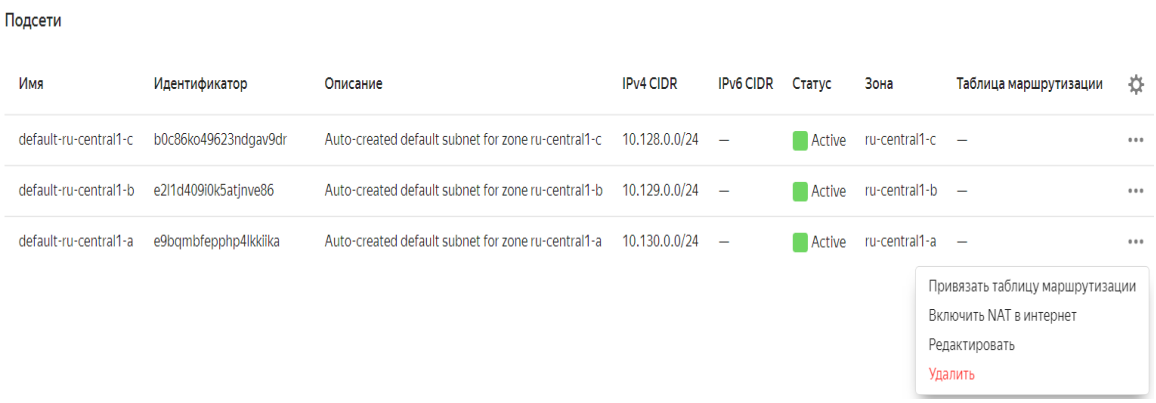


Рисунок 42 – Доступ в интернет для подсети

При первом запуске, необходимо описать причину, по которой нужен доступ в интернет и нажать кнопку «Отправить». Далее нужно ждать одобрения заявки от администрации, до тех пор при нажатии на эту кнопку будет показано сообщение (рисунок 43).

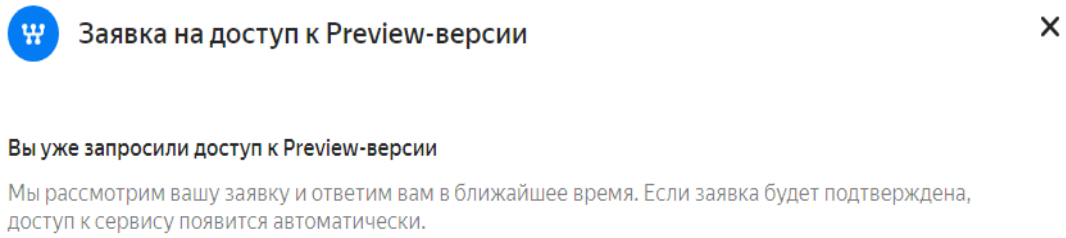


Рисунок 43 – Запрос к Preview версии

После всех настроек появляется возможность запустить работу проекта и оценить скорость работы. Система предоставляет ровно столько ресурсов пользователю, сколько ему необходимо. Существует множество вариантов конфигурации кластеров, и это не все преимущества данного решения. Зайдя в настройки кластера, можно увидеть строку сервисы, где описаны многие сервисы, которые предоставляет Yandex Cloud, можно сразу создать кластер, объединяющий возможности Apache Hadoop и Apache Spark, вся настройка занимает небольшое количество времени и все операции можно выполнять в веб интерфейсе в отличие от первого метода. Можно расширять настройки кластера по своему усмотрению, создавать подкластеры, добавлять в сеть виртуальные машины и многое другое. Окно с настройками для кластера «Cluster1» показано на рисунке 44.

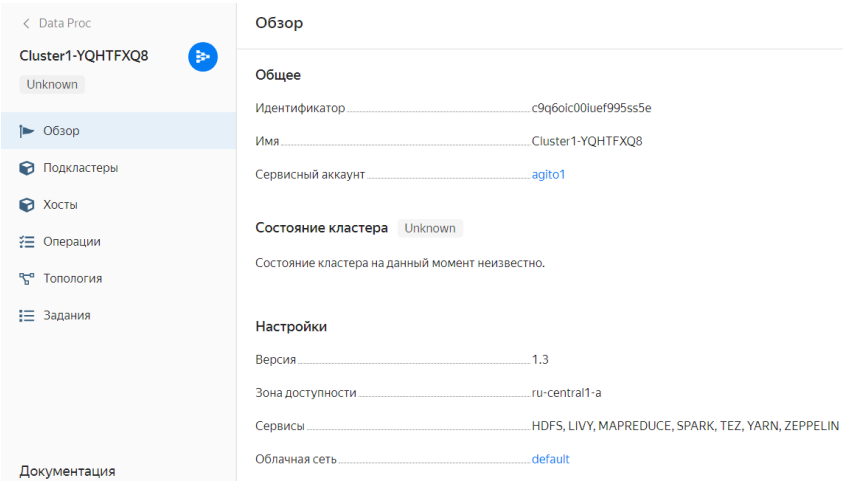
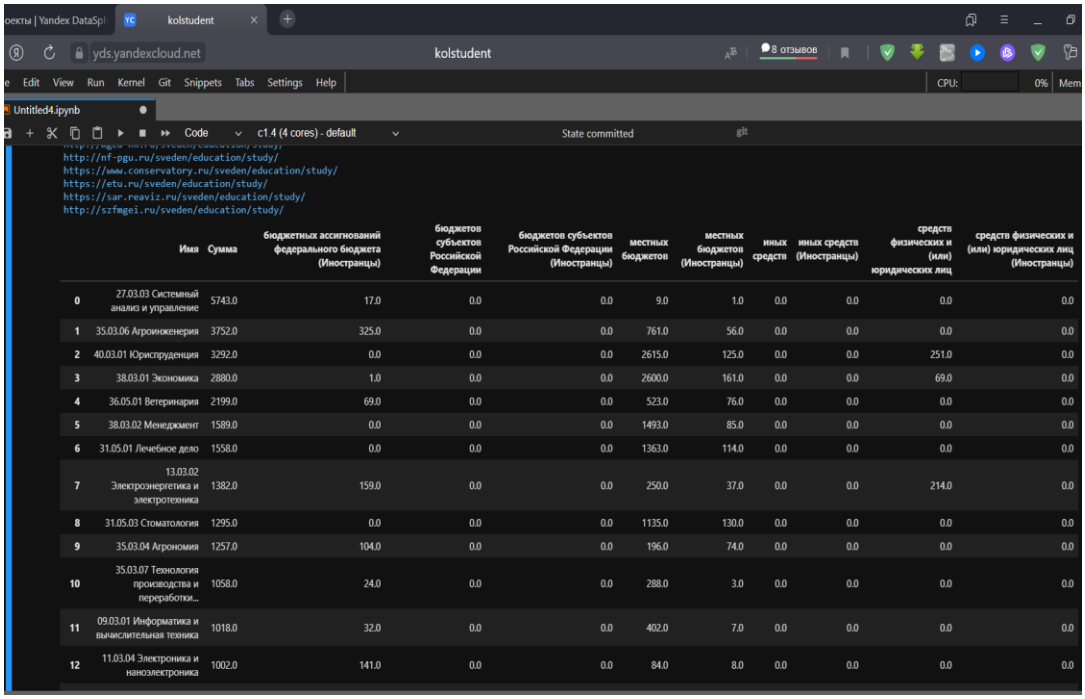


Рисунок 44 – Настройки Cluster1

После всех выполненных настроек можно запустить проект, он автоматически выполнится на выбранных кластерах и существенно ускорит работу программы. Несмотря на то, что данные обрабатываются на облаке, работа ускоряется в разы. Результат выполнения не искажается (рисунок 45).



	Имя	Сумма	бюджетных ассигнований федерального бюджета (Иностранцы)	бюджетов субъектов Российской Федерации	бюджетов субъектов Российской Федерации (Иностранцы)	местных бюджетов	местных бюджетов (Иностранцы)	иных средств	иных средств (Иностранцы)	средств физических и (или) юридических лиц	средств физических и (или) юридических лиц (Иностранцы)
0	27.03.03 Системный анализ и управление	5743.0	17.0	0.0	0.0	9.0	1.0	0.0	0.0	0.0	0.0
1	35.03.06 Агроинженерия	3752.0	325.0	0.0	0.0	761.0	56.0	0.0	0.0	0.0	0.0
2	40.03.01 Юриспруденция	3292.0	0.0	0.0	0.0	2615.0	125.0	0.0	0.0	251.0	0.0
3	38.03.01 Экономика	2880.0	1.0	0.0	0.0	2600.0	161.0	0.0	0.0	69.0	0.0
4	36.05.01 Ветеринария	2199.0	69.0	0.0	0.0	523.0	76.0	0.0	0.0	0.0	0.0
5	38.03.02 Менеджмент	1589.0	0.0	0.0	0.0	1493.0	85.0	0.0	0.0	0.0	0.0
6	31.05.01 Лечебное дело	1558.0	0.0	0.0	0.0	1363.0	114.0	0.0	0.0	0.0	0.0
7	13.03.02 Электроэнергетика и электротехника	1382.0	159.0	0.0	0.0	250.0	37.0	0.0	0.0	214.0	0.0
8	31.05.03 Стоматология	1295.0	0.0	0.0	0.0	1135.0	130.0	0.0	0.0	0.0	0.0
9	35.03.04 Агрономия	1257.0	104.0	0.0	0.0	196.0	74.0	0.0	0.0	0.0	0.0
10	35.03.07 Технология производства и переработки...	1058.0	24.0	0.0	0.0	288.0	3.0	0.0	0.0	0.0	0.0
11	09.03.01 Информатика и вычислительная техника	1018.0	32.0	0.0	0.0	402.0	7.0	0.0	0.0	0.0	0.0
12	11.03.04 Электроника и наноэлектроника	1002.0	141.0	0.0	0.0	84.0	8.0	0.0	0.0	0.0	0.0

Рисунок 45 – Результат выполнения работы программы на кластере

Таким образом один из основных недостатков при работе с большими данными, был решен при помощи решения Yandex Cloud.

3.6 Кластеризация и визуализация

Результаты исследования должны быть представлены более наглядно, было решено использовать методы визуализации язык python. Для построения графиков используются библиотеки «matplotlib.pyplot» и «seaborn».

Библиотека «matplotlib», это «основа» построения графиков для языка python. Чтобы скрыть огромное количество ненужного кода, поверх «matplotlib» используется библиотека seaborn, её особенность — это заложенный в неё механизм предварительной обработки данных, который взаимодействуя с библиотекой pandas, это дает возможность получать и передавать данные сразу в dataframe.

Перед тем как строить графики, при помощи функции set, задаем размер фигуры. Для построения столбчатой диаграммы используется функция «barplot». Наиболее удобно построить диаграмму, где по оси «Y» будет располагаться столбец «Сумма», а по «X» имя направления подготовки/специальности. При построении данного графика берутся значения, сгруппированные по столбцу «Имя», т.е не берутся в учет столбцы содержащие информацию о форме обучения и уровне образования. Данный график представлен на рисунке 46.

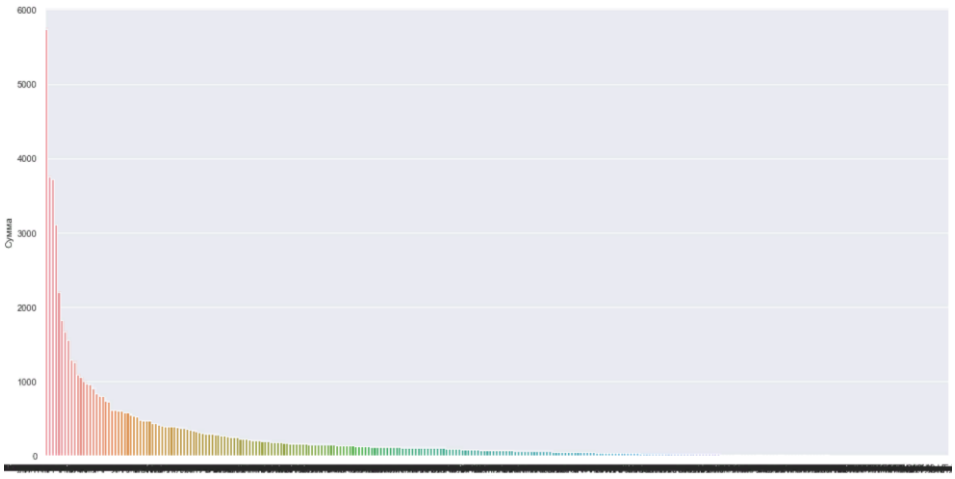


Рисунок 46 – График зависимости суммы от имени

На рисунке 46 видно черную полосу под графиком, это большое количество имен столбцов, наложенных друг на друга. При обработке больших данных это частое явление. Осуществим выборку десяти наиболее популярных направлений подготовки/специальностей, и сделаем вывод диаграммы, основанной на отфильтрованных данных. График примет более читаемый вид (Рисунок 47).

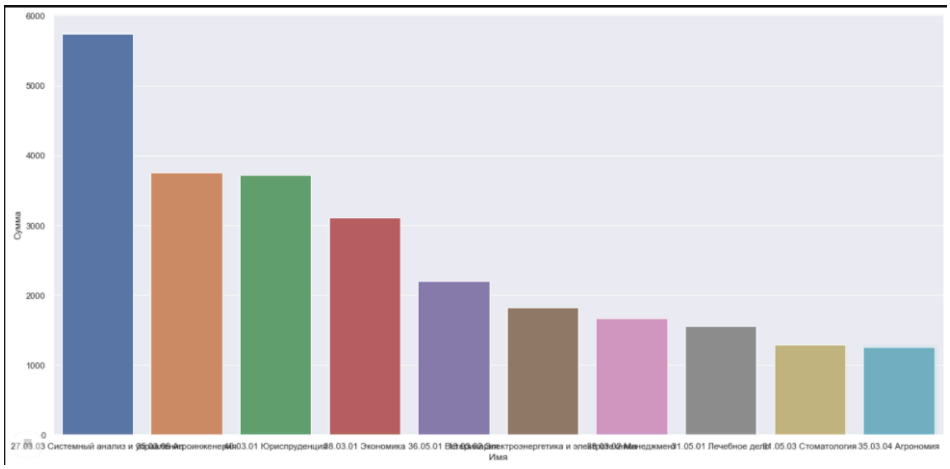


Рисунок 47 – Отфильтрованных график топа специальностей/ направлений подготовки

На рисунке 47, наименования столбцов всё равно сливаются, поменяем оси местами для создания более читаемого графика, и увеличим размер шрифта (рисунок 48).

Рисунок 48 – График топа специальностей/направлений подготовки со сменой осей

Такие действия можно сделать при помощи библиотеки seaborn, и это занимает всего несколько секунд времени пользователя. Если вынести код создания диаграммы и количества наиболее популярных направлений подготовки/специальностей в отдельный блок «In», то не нужно пере вызывать функцию по созданию таблицы постоянно, это очень удобно и позволяет достаточно быстро изменять графики к виду необходимому пользователю. Следует уточнить, что можно вызывать так же топ самых непопулярных направлений подготовки/специальностей, и сделать такие же диаграммы.

В программе можно создать круговую диаграмму. Вид такой диаграммы представлен на рисунке 49.

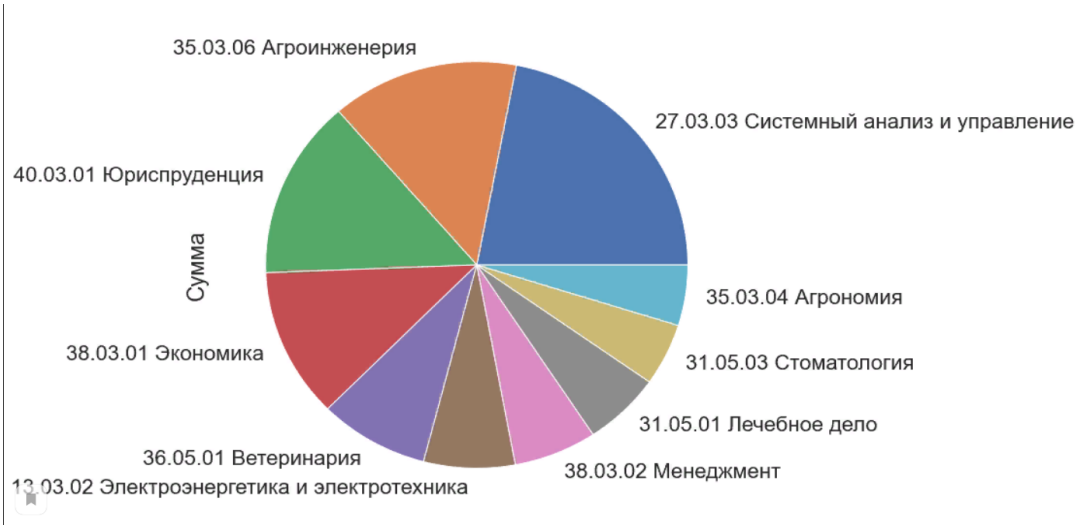


Рисунок 49 – Круговая диаграмма

Стандартный шрифт для рисунка 49 был увеличен, для наилучшей видимости, если пользователю нужно уменьшить шрифт он может это сделать.

Seaborn так же дает возможность создать графики корреляции (рисунок 50)

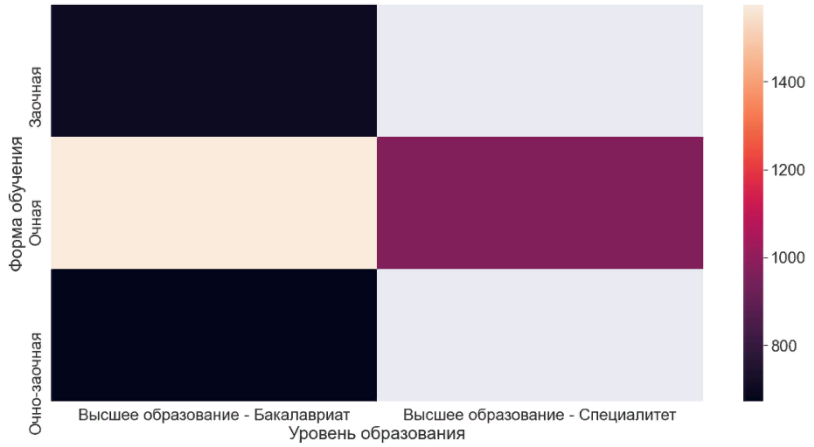


Рисунок 50 – Диаграмма корреляции для столбцов «Уровень образования» и «Форма обучения»

Диаграмма, представленная на рисунке 50, показывает, что наибольшее количество студентов, из десяти отобранных направлений/специальностей, обучается на очной форме обучения, и имеет уровень образования «Высшее образование – Бакалавриат». Такие графики можно построить по абсолютно разным столбцам, чтобы увидеть наглядно их взаимодействие.

Ещё одним примером построенной диаграммы корреляции – диаграмма показывающая зависимость названия специальности/направления подготовки от формы обучения (рисунок 51).

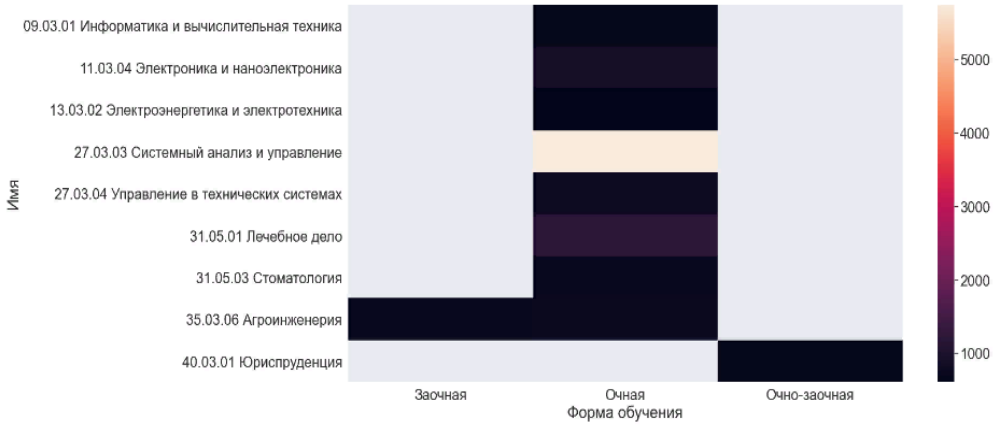


Рисунок 51 – Диаграмма корреляции для столбцов «Имя» и «Форма обучения»

На рисунке 51 можно заметить, что только два последних направления подготовки, дают возможность обучаться заочно и очно – заочно.

Последний график корреляции показывает зависимость уровня образования от формы обучения (рисунок 52).

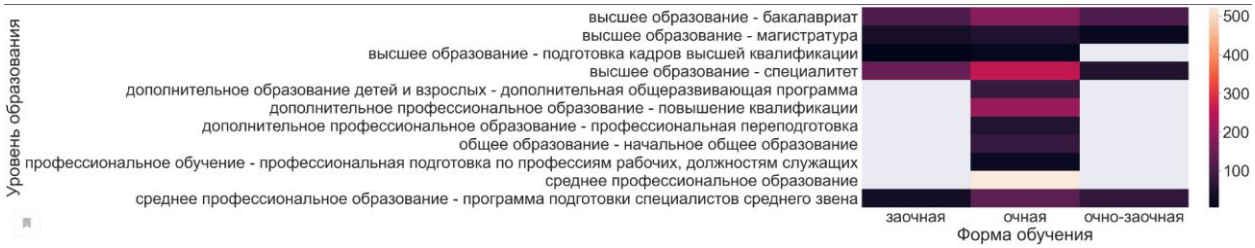


Рисунок 52 – Диаграмма корреляции для столбцов «Уровень образования» и «Форма обучения»

Из диаграммы, представленной на рисунке 52 видно, что наибольшее количество студентов получает среднее профессиональное образование.

Ещё одним положительным аспектом библиотеки seaborn, является то, что, при тесной интеграции она может выполнять кластеризацию объектов, без усложнения работы математическими функциями. Поскольку кластеризация представляет собой деление на группы по какому-либо признаку, осуществим кластеризацию по имени и сумме, а в качестве цвета укажем столбец форма обучения (рисунок 53).

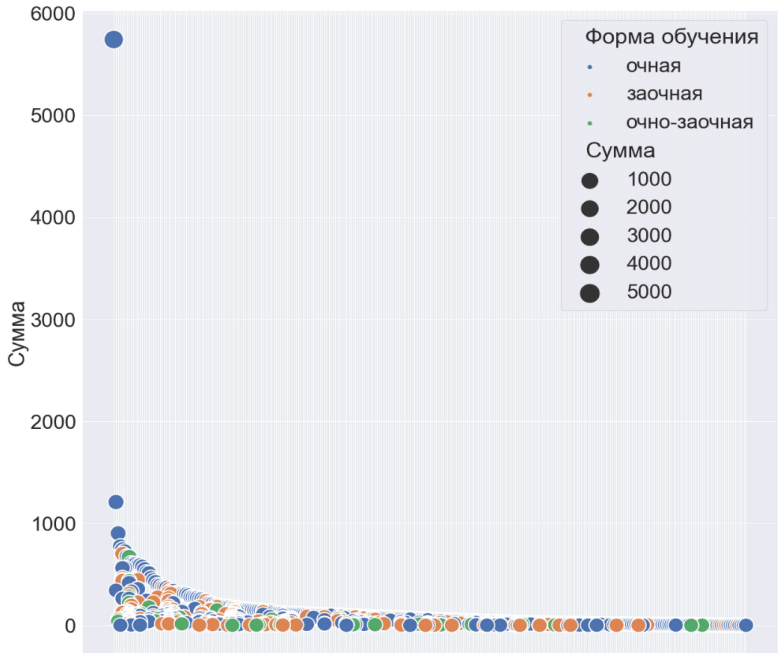


Рисунок 53 – Кластеризация

Из рисунка 53 видно большое количество точек, каждая из которых включает себя три параметра: форма обучения, сумма и имя. Таким образом на этом графике может быть представлено, например направление «35.03.05 Агроинженерия» очная, и «35.03.06 Агроинженерия» заочное направление. Даже без группировки по имени, лидером среди направлений подготовки остается «27.03.03 Системный анализ и управление». Размер точки зависит от суммы, чем больше сумма, тем больше точка. Соответственно выполнить кластеризацию можно не только по этим параметрам, но и по любому из столбцов таблицы. Поскольку, по результатам других диаграмм видно, что самое популярное направление подготовки имеет весомый отрыв от других, был увеличен минимальный размер элемента.

Чтобы подробно рассмотреть пример кластеризации, выведем на диаграмме только десять самых популярных направлений подготовки/специальностей, без группировки по имени (рисунок 54).

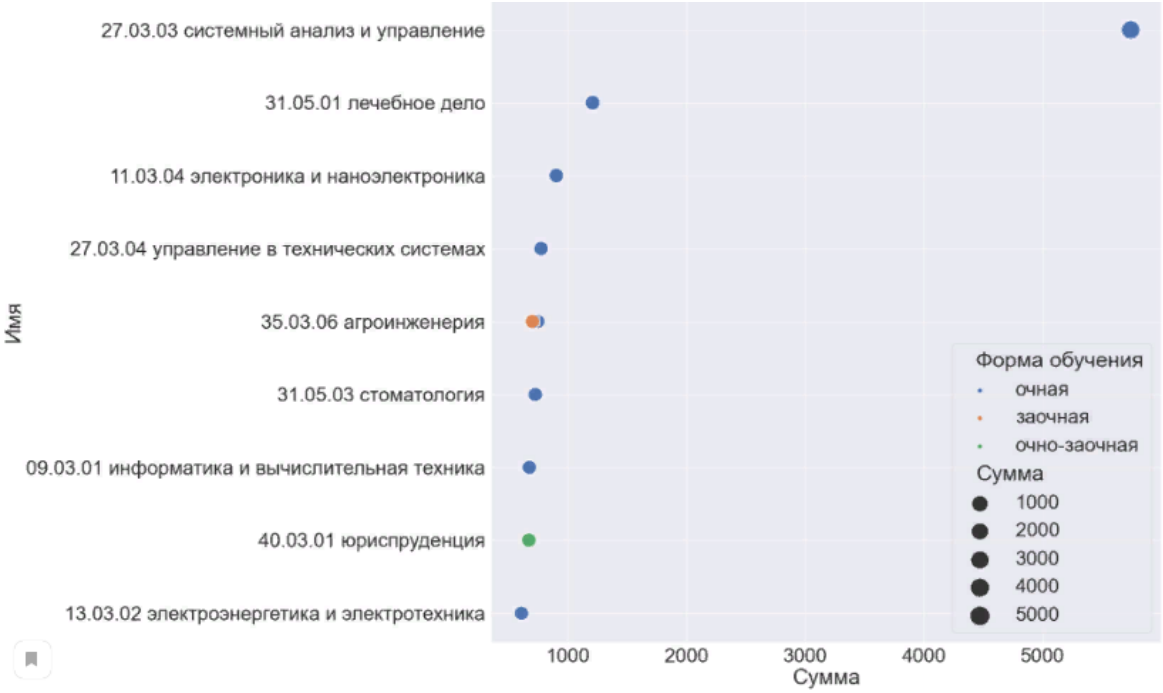


Рисунок 54 – Кластеризация десяти самых популярных направлений/специальностей

На рисунке 54 можно увидеть, у каких направлений подготовки/специальностей, есть альтернативные формы обучения, которые суммируются при группировке по имени, это очень удобно, если делать выборку.

3.7 Тестирование программного продукта

Начать тестирование программы следует с теста времени работы программы в трех вариантах, первый вариант, это работа программы на локальном компьютере (рисунок 55).

```
http://nf-pgu.ru/sveden/education/study/
https://www.conservatory.ru/sveden/education/study/
https://etu.ru/sveden/education/study/
https://sar.reaviz.ru/sveden/education/study/
http://szfmgei.ru/sveden/education/study/
Время, затраченное на выполнение данного кода =  35.25927495956421
```

Рисунок 55 – Время выполнения программы на локальном компьютере

Время работы на локальном компьютере достаточно большое. Локальный компьютер обладает следующими характеристиками: 12 ГБ ОЗУ, видеокарта Nvidia MX 150, процессор Intel(R) Core(TM) i5-8250U.

Оценим время работы программы на кластере, развернутом с помощью фреймворка Hadoop, кластер состоит из двух машин (рисунок 56).

```
Время, затраченное на выполнение данного кода = 23,28573465184736
```

Рисунок 56 – Время выполнения программы на кластере Hadoop

Время работы на кластере Hadoop уменьшилось, за счет того, что в кластер были включены ресурсы другого компьютера.

Третий вариант работы, на облачном сервисе Yandex Cloud, с выделенным кластером (рисунок 57).

```
https://sar.reaviz.ru/sveden/education/study/
http://szfmgei.ru/sveden/education/study/
Время, затраченное на выполнение данного кода =  8.81182312965393
```

Рисунок 57 – Время выполнения программы на облаке «Yandex Cloud»

Третий вариант работы включает только вычислительные мощности, предоставляемые компанией «Яндекс». Ресурсы локального компьютера используются только для подключения к веб-интерфейсу.

Тесты производились при парсинге информации с двадцати четырех сайтов, при увеличении объема собираемой информации увеличиться и время выполнения. Но из рисунков 55,56,57 видно, что кластер, развернутый на облаке, значительно ускоряет время выполнения работы программы. Это самый лучший

вариант работы, в том случае если пользователь не хочет настраивать кластер локально.

При первичном тесте программы наиболее частой проблемой являлись предупреждения о версии pandas, при парсинге таблиц с семью столбцами (рисунок 58).

```
<ipython-input-1-c4cf10a7747b>:72: FutureWarning: Starting with Pandas version 2.0 all arguments of read_html except for the ar
gument 'io' will be keyword-only
  df = pd.read_html(i,'Численно')[0].fillna(method='ffill')
```

Рисунок 58 – «Предупреждение о версии pandas»

Для отключения предупреждения, используется библиотека warnings, и производиться фильтрация и игнорирование предупреждение. После включения фильтра предупреждение не выводиться.

Следующая ошибка связана с тем, что при парсинге информации некоторые сайты могут быть недоступны, в этом случае процесс обработки прекращался и результат не выводился.

Решением этой проблемы является библиотека requests, используемая для составления HTTP-запросов в Python. В ней указывается параметр «timeout», который устанавливает время ожидания ответа от указываемого параметра. В том случае если время ожидания истекло, то автоматически анализируется следующая ссылка.

При парсинге иногда возникает ошибка, связанная с сертификатом сайта, для того чтобы обработать эту ошибку, была использована библиотека ssl, которая автоматически создает сертификат, для прохождения верификации. Было добавлено стандартное имя пользователя для доступа к некоторым сайтам.

Для того чтобы не перерабатывать информацию несколько раз, для получения новых графиков, было решено составить составление графиков в отдельных блоках «In».

Таким образом можно получить данные один раз, а после вызывать функции для составления графиков, которые необходимы пользователям.

После исправления этих проблем программа работает.

3.8 Анализ достоверности и практической значимости результатов

После того, как была протестирована работоспособность программы, необходимо осуществить анализ достоверность результатов. Было решено выбрать несколько направлений подготовки, при помощи программы Excel создать таблицу и автоматизировано посчитать количество обучающихся. Сделать это необходимо поскольку не существует универсального программного продукта, который смог бы осуществить выборочно парсинг данных с сайтов образовательных организаций. Тестирование проводиться при парсинге для двадцати пяти сайтов.

Первое направление подготовки «40.03.01 юриспруденция», в пункте 3.4 была создана сводная таблица, которая показывала, что на этом направлении подготовки обучается 3719 человек.

Осуществим переход по каждому сайту, выберем там нужное направление подготовки и занесем в таблицу excel. Полученная таблица представлена на рисунке 59.

Сайт	Направление подготовки	Очная(С ин)	Заочная(С ин)	Очно-заочная(С ин)	Сумма
https://mgeu-kirov.ru/sveden/education/study/	40.03.01 Юриспруденция	35	58	47	140
http://www.amursu.ru/sveden/education/study/	40.03.01 Юриспруденция	264	98	65	427
https://rpa-mu.ru/sveden/education	40.03.01 Юриспруденция	417	107	28	552
https://biti.mephi.ru/sveden/education/	40.03.01 Юриспруденция	нет	нет	нет	0
http://www.iile.ru/sveden/education/study/	40.03.01 Юриспруденция	121	32	436	589
https://www.imsit.ru/sveden/education/study/	40.03.01 Юриспруденция	нет	нет	нет	0
http://www.bsaa.edu.ru/sveden/education/study,	40.03.01 Юриспруденция	нет	нет	нет	0
http://vfmgei.ru/sveden/education/study/	40.03.01 Юриспруденция	68	46	224	338
https://vgsha.info/sveden/education/study/	40.03.01 Юриспруденция	нет	нет	нет	0
http://iai.sursau.ru/sveden/education/study/	40.03.01 Юриспруденция	нет	нет	нет	0
http://insagro.sursau.ru/sveden/education/study	40.03.01 Юриспруденция	нет	нет	нет	0
http://www.isi-vuz.ru/sveden/education/study/	40.03.01 Юриспруденция	нет	нет	нет	0
https://www.kf-rmat.ru/sveden/education/study,	40.03.01 Юриспруденция	нет	нет	нет	0
https://kgsxa.ru/sveden/education/study/	40.03.01 Юриспруденция	нет	нет	нет	0
http://lfkai.ru/sveden/education/study/	40.03.01 Юриспруденция	нет	нет	нет	0
https://reaviz.ru/sveden/education/study/	40.03.01 Юриспруденция	нет	нет	нет	0
https://www.miu-sochi.ru/sveden/education/stu	40.03.01 Юриспруденция	89	30	672	791
https://www.miu-edu.ru/sveden/education/study	40.03.01 Юриспруденция	60	6	285	351
http://mgeu-nk.ru/sveden/education/study/	40.03.01 Юриспруденция	нет	нет	нет	0
http://nf-pgu.ru/sveden/education/study/	40.03.01 Юриспруденция	нет	нет	нет	0
https://www.conservatory.ru/sveden/education/	40.03.01 Юриспруденция	нет	нет	нет	0
https://etu.ru/sveden/education/study/	40.03.01 Юриспруденция	нет	нет	нет	0
https://sar.reaviz.ru/sveden/education/study/	40.03.01 Юриспруденция	нет	нет	нет	0
http://szfmgei.ru/sveden/education/study/	40.03.01 Юриспруденция	130	312	89	531
Общее количество:					3719

Рисунок 59 – Таблица для направления подготовки «40.03.01 юриспруденция»

Общее количество обучающихся полученное при парсинге, и при подсчете автоматизировано совпадает, это говорит о правильности работы программы.

Для второго примера выберем менее популярное направление подготовки «09.03.01 информатика и вычислительная техника» (рисунок 60).

Сайт	Направление подготовки	Очная(С ин)	Заочная(С ин)	Очно-заочная(С ин)	Сумма
https://mgeu-kirov.ru/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
http://www.amursu.ru/sveden/education/study/	09.03.01 информатика и вь	76	нет	нет	76
https://rpa-mu.ru/sveden/education	09.03.01 информатика и вь	нет	нет	нет	0
https://biti.mephi.ru/sveden/education/	09.03.01 информатика и вь	нет	нет	нет	0
http://www.iile.ru/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
https://www.imsit.ru/sveden/education/study/	09.03.01 информатика и вь	15	78	нет	93
ttp://www.bsaa.edu.ru/sveden/education/study,	09.03.01 информатика и вь	нет	нет	нет	0
http://vfmgei.ru/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
https://vgsha.info/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
http://iai.sursau.ru/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
http://insagro.sursau.ru/sveden/education/study	09.03.01 информатика и вь	нет	нет	нет	0
http://www.isi-vuz.ru/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
https://www.kf-rmat.ru/sveden/education/study,	09.03.01 информатика и вь	нет	нет	нет	0
https://kgsxa.ru/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
http://lfkai.ru/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
https://reaviz.ru/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
https://www.miu-sochi.ru/sveden/education/stu	09.03.01 информатика и вь	нет	нет	нет	0
https://www.miu-edu.ru/sveden/education/stud	09.03.01 информатика и вь	нет	нет	нет	0
http://mgeu-nk.ru/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
http://nf-pgu.ru/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
https://www.conservatory.ru/sveden/education/	09.03.01 информатика и вь	нет	нет	нет	0
https://etu.ru/sveden/education/study/	09.03.01 информатика и вь	676	110	139	925
https://sar.reaviz.ru/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
http://szfmgei.ru/sveden/education/study/	09.03.01 информатика и вь	нет	нет	нет	0
Общее количество:					1094

Рисунок 60 – Таблица для направления подготовки «09.03.01 информатика и вычислительная техника»

Общее количество так же совпадает, с тем количеством, которое получается при парсинге. Приведенные два примера совпали с результатами автоматизированного расчета, который выполнялся в пункте 3.4. Это подтверждает корректность расчетов.

При анализе достоверности возникли проблемы при рассмотрении построенных диаграмм, возникало дублирование формы обучения и уровня образования из-за того, что они не были приведены к стандартизированному виду.

Это отразилось при построении диаграмм, вид подобной диаграммы представлен на рисунке 61.

Рисунок 61 – Ошибка в графике

Как видно на рисунке 61, происходит дублирование форм обучения, из-за того, что некоторые люди вносят информацию с заглавной буквы, а другие нет, этот недостаток был устранен, переводом столбцов «Форма обучения», «Уровень образования», и «Имя» в нижний регистр. Помимо этого, в столбце «Уровень образования», иногда используют тире, а иногда среднее тире, этот недостаток был решен заменой всех средних тире на тире в столбце. Такая ошибка не влияла на подсчеты, сгруппированные по имени, но, если пользователь осуществлял группировку по другим признакам, некоторые данные были бы сгруппированы неправильно. Эта ошибка, влияющая на достоверность информации была устранена.

Практические результаты исследования применимы для образовательных организаций, они позволяют оценить популярность направлений подготовки/специальностей, не только в общем, но и по определенным регионам.

На текущий момент времени, из-за нестабильной обстановки в стране, многие образовательные организации переходят на дистанционное обучение, а обучающиеся меняют свой взгляд на образование в целом. В связи с этим, происходит изменение популярности направлений подготовки/специальности.

Программный продукт позволит организации узнать наиболее популярное направление подготовки/специальность, либо вести статистику изменения популярности, и на её основе делать выводы.

ЗАКЛЮЧЕНИЕ

Разработанный программный продукт имеет узконаправленную специализацию, но это не делает его менее востребованным. Ведь из-за нестабильной ситуации в мире в связи с пандемией коронавирусной инфекции, многие обучающиеся изменили свой взгляд на образование. Вопрос выбора направления подготовки/специальности сейчас, как некогда играет важную роль, а определить то, какие направления подготовки/специальности наиболее актуальны позволяет созданный программный продукт.

Программу можно использовать как локально (в том случае если анализируется информация с относительно небольшого количества сайтов), так и на кластере (когда большой объем информации не может анализировать на одной машине). Этот функционал был протестирован в нескольких возможных вариациях.

Программный продукт не претендует на то, чтобы вуз определял какое направление подготовки/специальность открывать только по результатам работы программы, но он может быть одним из факторов этого выбора.

Существуют ситуации, когда после определенного периода времени меняются тенденции развития образования и многие специальности/направления подготовки становятся не востребованными. Но если осуществлять парсинг достаточный период времени, и хранить результаты хотя бы десяти наиболее популярных направлений подготовки/специальностей, то можно проследить динамику.

Результаты исследования позволили выяснить наиболее популярные направления подготовки/специальности среди двадцати четырех вузов страны, результаты достоверны и могут быть использованы для формирования статистической информации и принятия решений на их основе.

Помимо этого, программа предоставляет возможности сортировки информации по различным критериями, и достаточно широкий диапазон методов визуализации информации.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

1 Молодежь XXI века: шаг в будущее: матер. XXI регион. науч.- практ. конф. (Благовещенск, 20 мая 2020 г.). В 4 т. – Благовещенск: Изд-во Дальне-вост. гос. аграр. ун-та, 2020. –Т. 4: Сельскохозяйственные науки. Физико-математические науки. Химические науки. Информационные технологии. Технические науки. – 235 с.

2 Иванов, П. Д. Технологии big data и их применение на современном промышленном предприятии / П. Д. Иванов, В. Ж. Вампилова // Инженерный журнал: наука и инновации. – 2014. – № 8(32). – С. 3.

3 Агаев Ф.Т. Применение методов анализа больших данных в электронном образовании / Ф.Т. Агаев, Г.А. Мамедова, Л.А. Зейналова, Р.Т. Меликова// Инновации в образовании. – 2018. – № 9 – С. 83-95.

4 Там же. С. 84.

5 Там же. С. 87.

6 Ковшик А.А. Выбор методов и средств для обработки больших данных в сфере электронного обучения / А.А. Ковшик, С.Г. Самохвалова// Вестник Амурского государственного университета. Серия: Естественные и экономические науки. – 2020. – № 89 – С. 32-35.

7 Локтев, Е. С. Принципы работы с большими данными / Е.С. Локтев, Н.С. Бутенко, В.В. Данилова// Вестник современных исследований. – 2018. – № 8.1(23) – С. 271-273.

8 Бухарин, С. В. Математические методы экспертизы в экономике : учебное пособие / С. В. Бухарин, А. В. Мельников. – Воронеж : Воронежский государственный университет инженерных технологий, 2012. – 329 с.

9 Там же.

10 Кошкин, Д. Е. Анализ и сравнение алгоритмов кластеризации данных применительно к кластеризации текстового контента / Д. Е. Кошкин, Н. В. Багдасарова // Информатизация образования и науки. – 2018. – № 4(40). – С. 116-128.

11 Там же.

12 Гращенко, Л. А. Snowforce: быстрый стеммер для русского языка / Л. А. Гращенко, В. А. Муравлев // Новые информационные технологии в автоматизированных системах. – 2018. – № 21. – С. 194-200.

13 Жердева, М. В. Стемминг и лемматизация в lucene.net / М. В. Жердева, В. М. Артюшенко // Вестник московского государственного университета леса - лесной вестник. – 2016. – № 3. – С. 131-134.

14 Там же.

15 Ilya Segalovich A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine [Электронный ресурс] // yandex.ru : офиц. сайт. – 2001. Режим доступа: https://cache-khabrt02.cdn.yandex.net/download.yandex.ru/company/iseg-las-vegas.pdf – 07.06.2020

16 Там же.

17 Иванова, Н. С. Распределенная обработка данных в информационных системах с использованием hadoop/ Н. С. Иванова // Информационные технологии : сб. науч. тр. / Рязан. гос. ун-т. – Рязань, 2019. – С. 107– 110.

18 Как устроен apache storm: архитектура и принцип работы [Электронный ресурс] : офиц. сайт. – 02.10.2019. Режим доступа: https://www.bigdataschool.ru/wiki/storm – 02.12.2020

19 Там же.

20 Революция Big Data: Как извлечь необходимую информацию из «Больших Данных»? [Электронный ресурс] : офиц. сайт. – 7.10.2015. Режим доступа: http://statsoft.ru/products/Enterprise/big-data.php – 01.02.2020

21 Тюрин, А. Г. Кластерный анализ, методы и алгоритмы кластеризации / А. Г. Тюрин, И. О. Зуев // Вестник МГТУ МИРЭА. – 2014. – № 2(3). – С. 86-97.

22 Кластеризация текста с помощью K-means и TF-IDF [Электронный ресурс]: Лямбда. – 13.04.2020. Режим доступа: https://lambda-it.ru/post/klasterizatsiia-teksta-s-pomoshchiu-k-means-i-tf-i –11.08.2020

23 Черепков, Е. А. Технологии для обработки и анализа больших данных / Е. А. Черепков, С. В. Рыбкин // Электронный журнал: наука, техника и образование. – 2016. – № 4(9). – С. 120-127.

24 Назаренко, Ю. Л. Обзор технологии "Большие данные" (big data) и программно-аппаратных средств, применяемых для их анализа и обработки / Ю.Л. Назаренко // European science. – 2017. – № 9(31). – С. 25-30.

25 Там же.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Агаев Ф.Т. Применение методов анализа больших данных в электронном образовании / Ф.Т. Агаев, Г.А. Мамедова, Л.А. Зейналова, Р.Т. Меликова// Инновации в образовании. – 2018. – № 9 – С. 83-95.

2 Безверхий, О.А. Кластеризация большого объема текстовых поисковых запросов / О.А. Безверхий, С.Г. Самохвалова // Ученые заметки ТОГУ. – 2016. – № 3-1. – С. 104-110.

3 Благирев, А. Big data простым языком. / А. Благирев. – Москва : Издательство АСТ, 2019. – 256 с.: ил. – (Бизнес-бук).

4 Бояров, А. Введение в кластеризацию [Электронный ресурс] / А. Бояров, А. Сенов // spbu.ru : офиц. сайт. – 10.03.2014. Режим доступа: https://math.spbu.ru/user/gran/courses/LECTURE_2_1.pdf. – 11.04.2021.

5 Брауде, Э. Технология разработки программного обеспечения. / Э. Брауде – СПб: Питер, 2004. – 655 с.

6 Бухарин, С. В. Математические методы экспертизы в экономике : учебное пособие / С. В. Бухарин, А. В. Мельников. – Воронеж : Воронежский государственный университет инженерных технологий, 2012. – 329 с.

7 Вайгенд, А. Big Data. Вся технология в одной книге. / А. Вайгенд; [перевод с английского С. Богданова]. – Москва : Эксмо, Бомбора, 2018 – 384 с.

8 Головицына, М.В. Информационные технологии в экономике : курс лекций / Головицына М.В. — Москва : Интуит НОУ, 2016. — 589 с.

9 Горбачев, С. В. Нейро-нечеткие методы в интеллектуальных системах обработки и анализа многомерной информации : монография / С. В. Горбачев, В. И. Сырямкин ; под редакцией В. И. Сырямкина. — Томск : ТГУ, 2014. — 442 с.

10 Гращенко, Л. А. Snowforce: быстрый стеммер для русского языка / Л. А. Гращенко, В. А. Муравлев // Новые информационные технологии в автоматизированных системах. – 2018. – № 21. – С. 194-200.

11 Жердева, М. В. Стемминг и лемматизация в lucene.net / М. В. Жердева, В. М. Артюшенко // Вестник московского государственного университета леса - лесной вестник. – 2016. – № 3. – С. 131-134.

12 Иванов, П. Д. Технологии big data и их применение на современном промышленном предприятии / П. Д. Иванов, В. Ж. Вампилова // Инженерный журнал: наука и инновации. – 2014. – № 8(32). – С. 3.

13 Иванова, Н. С. Распределенная обработка данных в информационных системах с использованием hadoop/ Н. С. Иванова // Информационные технологии : сб. науч. тр. / Рязан. гос. ун-т. – Рязань, 2019. – С. 107– 110.

14 Изофатов, К. А. Кластерный и интеллектуальный анализ текстовой информации. Основные понятия и проблемы [Электронный ресурс] // rae.ru : офиц. сайт. – 01.10.2005. Режим доступа: http://econf.rae.ru/pdf/2011/03/120.pdf – 16.01.2021

15 Как устроен apache storm: архитектура и принцип работы [Электронный ресурс] : офиц. сайт. – 02.10.2019. Режим доступа: https://www.big-dataschool.ru/wiki/storm – 02.12.2020

16 Кластеризация текста с помощью K-means и TF-IDF [Электронный ресурс]: Лямбда. – 13.04.2020. Режим доступа: https://lambda-it.ru/post/klasterizatsiia-teksta-s-pomoshchiu-k-means-i-tf-i –11.08.2020

17 Ковшик А.А. Выбор методов и средств для обработки больших данных в сфере электронного обучения / А.А. Ковшик, С.Г. Самохвалова// Вестник Амурского государственного университета. Серия: Естественные и экономические науки. – 2020. – № 89 – С. 32-35.

18 Котов, А. Кластеризация данных [Электронный ресурс] / А. Котов, Н. Красильников // docplayer.ru : офиц. сайт. – 2.10.2006. Режим доступа: https://docplayer.ru/26368283-Klasterizaciya-dannyh.html – 04.05.2021

19 Кошкин, Д. Е. Анализ и сравнение алгоритмов кластеризации данных применительно к кластеризации текстового контента / Д. Е. Кошкин, Н. В. Багдасарова // Информатизация образования и науки. – 2018. – № 4(40). – С. 116-128.

20 Кравченко, В.О «Большие данные» - практические аспекты и особенности / В. О. Кравченко, А. А. Крюкова // Academy. – 2016. – № 6(9). – С. 65-67.

21 Локтев, Е. С. Принципы работы с большими данными / Е.С. Локтев, Н.С. Бутенко, В.В. Данилова// Вестник современных исследований. – 2018. – № 8.1(23) – С. 271-273.

22 Майер-Шенбергер, В. Большие данные. Революция, которая изменит то, как мы живем, работаем и мыслим / Виктор Майер-Шенбергер, Кеннет Кукьер; пер с англ. Инны Гайдюк. – М.: Манн, Иванов и Фербер, 2014. – 240 с.

23 Материалы XIV международной научной конференции «Системный анализ в меди-цине» (САМ 2020) / под общ. ред. В.П. Колосова. Благовещенск, 2020. 192 с.

24 Михеев, Д. С. Построение онтологических связей в области знаний на основании поиска и анализа текстовых ссылок / Д. С. Михеев // International journal of open information technologies. – 2018. – № 11(6). – С. 50-53.

25 Молодежь XXI века: шаг в будущее: матер. XXI регион. науч.- практ. конф. (Благовещенск, 20 мая 2020 г.). В 4 т. – Благовещенск: Изд-во Дальневост. гос. аграр. ун-та, 2020. –Т. 4: Сельскохозяйственные науки. Физико-математические науки. Химические науки. Информационные технологии. Технические науки. – 235 с.

26 Назаренко, Ю. Л. Обзор технологии "Большие данные" (big data) и программно-аппаратных средств, применяемых для их анализа и обработки / Ю.Л. Назаренко // European science. – 2017. – № 9(31). – С. 25-30.

27 Нечипорук, Д. В. Особенности технологии Datamining / Д. В. Нечипорук // Молодой исследователь Дона. – 2017. – № 1(4). – С. 1-4.

28 Обзор алгоритмов кластеризации данных [Электронный ресурс] : Сообщество IT-специалистов. – 11.10.2010. Режим доступа: https://habr.com/ru/post/101338/ – 10.06.2020

29 Распределенные вычисления с помощью Linux и Hadoop [Электронный ресурс] : офиц. сайт. – 02.04.2009. Режим доступа: https://www.ibm.com/developerworks/ru/library/l-hadoop/ – 03.05.2021

30 Революция Big Data: Как извлечь необходимую информацию из «Больших Данных»? [Электронный ресурс] : офиц. сайт. – 7.10.2015. Режим доступа: http://statsoft.ru/products/Enterprise/big-data.php – 01.02.2020

31 Свидетельство о государственной регистрации программы для ЭВМ 2021610970 Российская Федерация. Программа преобразования текстовой информации к виду, готовому для обработки / А. А. Ковшик, С. Г. Самохвалова ; заявитель и правообладатель Федеральное государственное бюджетное образовательное учреждение высшего образования «Амурский государственный университет». – № 2021610321 ; заявл. 13.12.2021 ; опубл 20.12.2021. – 1 с.

32 Современные подходы к исследованию и моделированию в экономике, финансах и бизнесе: Материалы конференции Европейского университета в Санкт-Петербурге и Санкт-Петербургского экономико-математического института РАН. – СПб.: Изд-во Европ. ун-та в С.- Петербурге, 2007. — 177 с.

33 Технология разработки программного обеспечения [Электронный ресурс] : учеб. – метод. пособие / АмГУ, ФМиИ ; сост. Т. А. Галаган. – Благовещенск : Изд-во Амур. гос. ун-та, 2015. – 49 с. Режим доступа: http://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/6799.pdf – 24.11.2020.

34 Технология разработки программного обеспечения [Электронный ресурс] : сб. учеб.-метод. материалов для направления подготовки 09.04.04 "Программная инженерия" / АмГУ, ФМиИ ; сост. Т. А. Галаган. - Благовещенск : Изд-во Амур. гос. ун-та, 2017. - 51 с. Режим доступа: http://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/10382.pdf – 06.02.2021.

35 Тюрин, А. Г. Кластерный анализ, методы и алгоритмы кластеризации / А. Г. Тюрин, И. О. Зуев // Вестник МГТУ МИРЭА. – 2014. – № 2(3). – С. 86-97.

36 Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» (ред. от 30.04.2021) // Собр. законодательства Российской Федерации. – 2012. – № 1. – ст. 404.

37 Черепков, Е. А. Технологии для обработки и анализа больших данных / Е. А. Черепков, С. В. Рыбкин // Электронный журнал: наука, техника и образование. – 2016. – № 4(9). – С. 120-127.

38 Чехарин, Е. Е. Большие данные – Большие проблемы / Е. Е. Чехарин // Перспективы науки и образования. – 2016. – № 3(21). – С. 7-11.

39 Чубукова, И.А. Data Mining : учебное пособие / Чубукова И.А. — Москва : Интуит НОУ, 2016. — 470 с.

40 Шлюйкова, Д. П. Большие данные – современные подходы к хранению и обработке / Д. П. Шлюйкова // Наука, техника и образование. – 2016. – № 1(19). – С. 75-11.

41 Hadoop: введение в системы больших данных [Электронный ресурс] : офиц. сайт. – 23.11.2018. Режим доступа: https://www.8host.com/blog/hadoop-vvedenie-v-sistemy-bolshix-dannyx/ – 18.12.2020

42 Hadoop: что, где и зачем [Электронный ресурс] : Сообщество IT-специалистов. – 16.10.2014. Режим доступа: https://habr.com/ru/post/240405/ – 21.05.2021

43 Ilya Segalovich A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine [Электронный ресурс] // yandex.ru : офиц. сайт. – 2001. Режим доступа: https://cache-khabrt02.cdn.yandex.net/download.yandex.ru/company/iseg-las-vegas.pdf – 07.06.2020