

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра _____ «Прикладная математика и информатика» _____
(наименование)

02.03.03 «Математическое обеспечение и администрирование информационных систем» _____
(код и наименование направления подготовки / специальности)

Мобильные и сетевые технологии _____
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка ПО для управления отчетом по успеваемости студентов ТГУ» _____

Обучающийся

Н.А. Ставицкий

_____ (Инициалы Фамилия)

_____ (личная подпись)

Руководитель

М.А. Тренина

_____ (ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

к.п.н., доцент А.В. Егорова

_____ (ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема выпускной квалификационной работы — «Разработка ПО для управления отчетом по успеваемости студентов ТГУ».

Ключевые слова: разработка ПО, управления отчетом, успеваемость студентов, ТГУ.

Как показывает практика, необходимым условием эффективного и увлекательного процесса обучения в вузе является наличие адаптивного и гибкого механизма оценки образовательной деятельности студента, которая полезна как для студентов, так и для преподавателей.

Таким механизмом является механизм формирования аналитической отчетности успеваемости студентов, для реализации которого используется специальное программное обеспечение (ПО).

Объектом исследования бакалаврской работы является отчет по успеваемости студентов ТГУ.

Предметом исследования бакалаврской работы является ПО для управления отчетом по успеваемости студентов ТГУ.

Цель бакалаврской работы — разработка ПО для управления отчетом по успеваемости студентов ТГУ.

Практическая значимость бакалаврской работы заключается в разработке и тестировании ПО, обеспечивающего эффективное управление отчетом по успеваемости студентов ТГУ.

Бакалаврская работа состоит из 42 страниц текста, 17 рисунков, 8 таблиц и 31 источника.

Abstract

The topic of the final qualifying work is “Development of software for managing a report on the progress of TSU students.”

Key words: software development, report management, student performance, TSU.

As practice shows, a necessary condition for an effective and exciting learning process at a university is the presence of an adaptive and flexible mechanism for assessing a student’s educational activities, which is useful for both students and teachers.

Such a mechanism is the mechanism for generating analytical reporting of student performance, for the implementation of which special software is used.

The object of study of the bachelor's work is a report on the progress of TSU students.

The subject of the bachelor's work is software for managing reports on the progress of TSU students.

The purpose of the bachelor's work is to develop software for managing a report on the progress of TSU students.

The practical significance of the bachelor's work lies in the development and testing of software that provides effective management of the report on the progress of TSU students.

The bachelor's thesis consists of 42 pages of text, 17 figures, 8 tables and 31 sources.

Оглавление

| | |
|---|----|
| Введение..... | 5 |
| Глава 1 Постановка задачи на разработку ПО для управления отчетом по успеваемости студентов ТГУ | 7 |
| 1.1 Методы анализа успеваемости студентов вуза..... | 7 |
| 1.2 Методы ранжирования на основе машинного обучения | 9 |
| 1.3 Разработка требований и анализ аналогов ПО | 11 |
| Глава 2 Проектирование ПО для управления отчетом по успеваемости студентов ТГУ | 18 |
| 2.1 Логическое проектирование ПО | 18 |
| 2.2 Алгоритм управления отчетом | 23 |
| Глава 3 Реализация и тестирование ПО для управления отчетом по успеваемости студентов ТГУ | 27 |
| 3.1 Выбор средства разработки ПО..... | 27 |
| 3.2 Реализация и тестирование ПО | 30 |
| Заключение | 37 |
| Список используемой литературы и используемых источников..... | 39 |

Введение

Как показывает практика, необходимым условием эффективного и увлекательного процесса обучения в вузе является наличие адаптивного и гибкого механизма оценки образовательной деятельности студента, которая полезна как для студентов, так и для преподавателей.

Таким механизмом является механизм формирования аналитической отчетности успеваемости студентов, для реализации которого используется специальное программное обеспечение (ПО).

В настоящее время в процесс обучения вузов активно внедряются технологии дистанционного обучения и LMS (Learning management system) — современные системы управления обучением, которые обеспечивают сбор и обработку больших образовательных данных.

Интеллектуальный анализ образовательных данных Тольяттинского государственного университета (ТГУ) показал, что большой объем данных о студентах, собираемых системами электронного обучения, можно использовать для формирования аналитической отчетности студентов для всестороннего исследования их успеваемости.

Основным источником таких данных является цифровой след студента.

Таким образом, разработка ПО для управления отчетом по успеваемости студентов ТГУ является актуальной и представляет практический интерес.

Объектом исследования бакалаврской работы является отчет по успеваемости студентов ТГУ.

Предметом исследования бакалаврской работы является ПО для управления отчетом по успеваемости студентов ТГУ.

Цель бакалаврской работы — разработка ПО для управления отчетом по успеваемости студентов ТГУ.

Для достижения данной цели необходимо выполнить следующие задачи:

– выполнить постановку задачи на разработку ПО для управления

- отчетом по успеваемости студентов ТГУ;
- спроектировать ПО для управления отчетом по успеваемости студентов ТГУ;
 - реализовать и протестировать ПО для управления отчетом по успеваемости студентов ТГУ.

Методы исследования — методы анализа успеваемости студентов вуза, методы и алгоритмы машинного обучения, методы и технологии проектирования ПО.

Практическая значимость бакалаврской работы заключается в разработке и тестировании ПО, обеспечивающего эффективное управление отчетом по успеваемости студентов ТГУ.

Данная работа состоит из введения, трех глав, заключения и списка используемой литературы.

Первая глава работы посвящена постановке задачи на разработку ПО для управления отчетом по успеваемости студентов ТГУ.

Вторая глава работы посвящена проектированию ПО для управления отчетом по успеваемости студентов ТГУ.

В третьей главе рассматриваются реализация и тестирование ПО для управления отчетом по успеваемости студентов ТГУ.

В заключении описываются результаты выполнения выпускной квалификационной работы.

Бакалаврская работа состоит из 42 страниц текста, 17 рисунков, 8 таблиц и 31 источника.

Глава 1 Постановка задачи на разработку ПО для управления отчетом по успеваемости студентов ТГУ

1.1 Методы анализа успеваемости студентов вуза

Оценка качества высшего образования состоит из внешней и внутренней оценки качества образовательных услуг.

Внешняя оценка качества образовательных услуг осуществляется всеми экономическими субъектами, которые являются потребителями. Внутренняя оценка качества образовательных услуг должна проводиться самим университетом. Для этих целей университет разрабатывает систему управления качеством и организует отдел, ответственный за организацию и мониторинг качества образовательных услуг.

Внутренняя оценка качества образовательных услуг включает такие параметры, как качество подготовки студентов и качество преподавания в университете.

Для решения данной задачи в вузах разрабатывается рейтинговая оценка знаний студентов на основе данных об их успеваемости [25].

Рейтинг студента — это числовой показатель оценивания успешности освоения дисциплины (учебного курса), отражающий место студента в групповом (поточном, курсовом) рейтинге в соответствии с полученным итоговым рейтинговым баллом [19].

Алгоритм формирования (составления) рейтинга студентов определяет критерии и порядок оценивания достижений студента в соответствии с выбранным методом формирования рейтинга студентов [2].

Источником данных для формирования рейтинга студентов являются большие образовательные данные (Educational Big Data, EBD) [31].

Как показывают результаты интеллектуального анализа образовательных данных (Educational Data Mining, EDM), основным источником данных для анализа успеваемости студента является его цифровой

след [4].

Цифровой след — это запись онлайн-активности пользователя Интернет.

Цифровой след показывает, где был пользователь, данные, которыми он поделился, и информацию, которые он передал или оставил [22].

Пассивные цифровые следы – это те, которые остаются без намерения пользователя Интернета, а иногда и без ведома об этом. Пассивные цифровые следы — это скрытый процесс, о котором пользователи Интернета могут вообще не знать.

Примером набора данных о пассивных цифровых следах студентов являются данные образовательного портала ТГУ. Студенты оставляют цифровые следы в любое время, когда они находятся в сети, независимо от того, находятся ли они в вузе или за его пределами [18].

Для оценки успеваемости студентов используется метод ранжирования.

Ранжирование — это отношение между набором элементов, такое, что для любых двух элементов первый либо «ранжируется выше», либо «ранжируется ниже», либо «ранжируется равно» второму. В математике это известно, как слабый порядок или полный предварительный порядок объектов.

Уровень ранжирования используется для определения различных рейтингов, используемых институтом, на основе процента баллов по каждому предмету и на основе количества предметов.

Как показал анализ, наиболее популярным методом ранжирования студентов является метод, основанный на применении среднего балла студента.

Средний балл студента (Grade Point Average, GPA) – это число, которое показывает, насколько высокие баллы в среднем набрали студенты на своих курсах.

Используя шкалу от 1,0 до 4,0, можно с помощью среднего балла отслеживать успеваемость студента. Это число используется для оценки того, соответствует ли студент стандартам и ожиданиям, установленным

программой получения квалификации.

Формула расчета среднего балла студента имеет вид (1):

$$GPA = SO / KD, \quad (1)$$

где SO – сумма оценок студента;

KD – количество зачетных дисциплин.

После этого с помощью выбранного алгоритма производится ранжирование всех студентов одной образовательной программы по GPA.

На основе определенных уровней ранжирования могут быть сгенерированы отчеты учащегося. Для серий с системой CWA / Normal следует выставлять оценки в процентах, а для серий с системой GPA – оценку GPA [30].

1.2 Методы ранжирования на основе машинного обучения

Принимая во внимание инновационные возможности методов ранжирования на основе машинного обучения, выбираем указанные методы для ранжирования студентов по среднему баллу [6].

Обучение ранжированию (Learning to rank) – класс задач машинного обучения, суть которых состоит в автоматизированном построении ограничений для ранжирующей модели по обучающей выборке, для их последующего применения к неизвестным объектам со сходной структурой [1].

Рассмотрим постановку задачи.

Пусть:

X – множество объектов;

$X^l = \{x_1, \dots, x_l\}$ – обучающая выборка;

$i < j$ – правильный порядок на парах $(i, j) \in \{1, \dots, l\}^2$.

Задача:

Построить ранжирующую функцию $a: X \rightarrow R$ такую, что (2):

$$i < j \Rightarrow a(x_i) < a(x_j). \quad (2)$$

Линейная модель ранжирования (3):

$$a(x; w) = \langle x, w \rangle, \quad (3)$$

где $x \mapsto (f_1(x), \dots, f_n(x)) \in R^n$ – вектор признаков объекта x .

Схема процесса ранжирования на основе машинного обучения показана на рисунке 1.

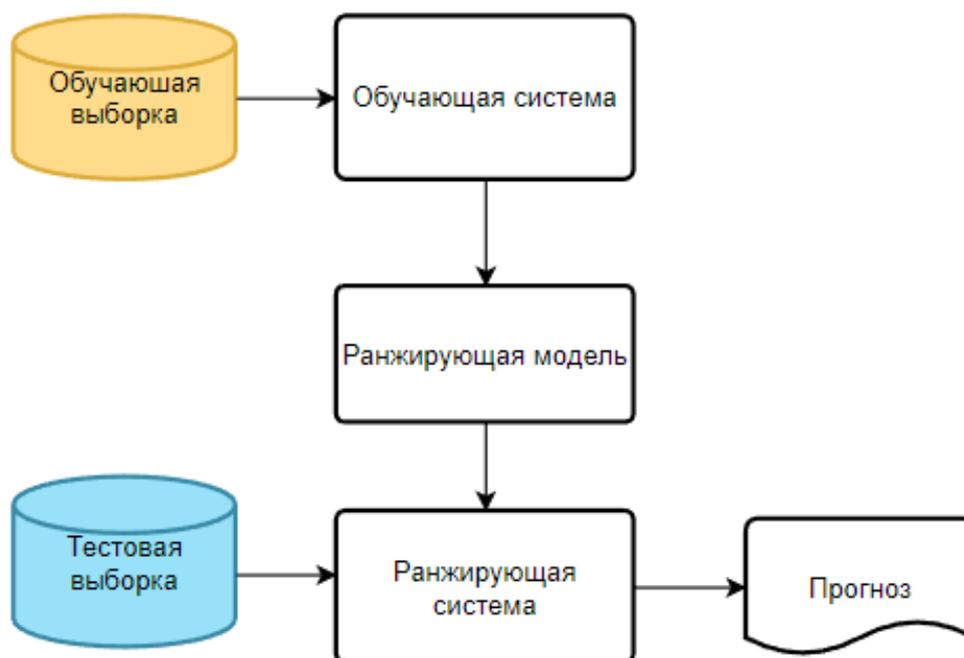


Рисунок 1 – Схема процесса ранжирования на основе машинного обучения

В качестве метода ранжирования используем аналитический иерархический процесс (Analytic Hierarchy Process, АИП) с кластеризацией данных.

На рисунке 2 представлена блок-схема алгоритма ранжирования по методу АНР с применением метода кластеризации [27].



Рисунок 2 – Блок-схема алгоритма ранжирования по методу АНР с применением метода кластеризации

Главное преимущество метода АНР заключается в простоте его реализации с помощью алгоритмов машинного обучения (МО).

1.3 Разработка требований и анализ аналогов ПО

Для разработки требований к ПО используем методологию FURPS+.

«FURPS+ — это расширенная модель классификации атрибутов качества программного обеспечения, которая помогает анализировать и специфицировать функциональные и нефункциональные требования к системе» [14].

В таблице 1 представлены требования к ПО для управления отчетом по успеваемости студентов ТГУ.

Таблица 1 – Требования к ПО для управления отчетом по успеваемости студентов ТГУ

| «Требование | Статус | Полезность | Риск | Стабильность |
|---|--------------|-------------|---------|---------------|
| Functionality — Функциональные требования | | | | |
| Управление отчетом по успеваемости студентов ТГУ | одобренное | средняя | средний | низкая |
| Формирование отчета ранжирования студентов | одобренное | средняя | средний | низкая |
| Экспорт отчета | одобренное | средняя | средний | низкая |
| Usability— Требования к удобству использования | | | | |
| Современный дизайн | одобренное | критическая | средний | низкая |
| Дружественный интуитивный интерфейс | одобренное | критическая | средний | низкая |
| Reliability— Требования к надежности | | | | |
| Допустимая частота/периодичность сбоев: 1 раз в 300 часов | одобренное | важная | средний | средняя |
| Среднее время сбоев: 1 раб. день | одобренное | важная | средний | средняя |
| Возможность восстановления системы после сбоев: 1 раб. день | одобренное | важная | средний | средняя |
| Режим работы: рабочий день | одобренное | важная | средний | средняя |
| Performance — Требования к производительности | | | | |
| Допустимое количество одновременно работающих пользователей: 10 | предложенное | важная | средний | средняя |
| Время реакции на возникновение аварийной ситуации: 10 с | предложенное | важная | средний | средняя |
| Supportability — Требования к поддержке | | | | |
| Время устранения критических проблем: в течение рабочего дня | предложенное | важная | средний | средняя» [10] |

Продолжение таблицы 1

| «Требование | Статус | Полезность | Риск | Стабильность |
|--------------------------------------|--------------|------------|---------|---------------|
| Проектные ограничения | | | | |
| Применение технологий МО | предложенное | важная | средний | средняя |
| Низкая совокупная стоимость владения | предложенное | важная | средний | средняя» [10] |

Разработанный перечень требований является основой для разработки ПО для управления отчетом по успеваемости студентов ТГУ.

С учетом вышеперечисленных требований рассмотрим и проанализируем аналоги ПО для управления отчетом по успеваемости студентов ТГУ.

Программа «Учебный учет» предназначена для учета посещаемости и успеваемости студентов в организациях среднего профессионального и высшего образования (рисунок 2) [16].

| № | ФИО студента | Сент | Октя | Нояб | Дек | Январ | Фев | Март | Апрел | Май | Июни | Итог |
|----|-------------------------|------|------|------|-----|-------|-----|------|-------|-----|------|------|
| 1 | Абрамова Елена Анатолье | 5 | 4 | | | X | X | X | X | X | X | 4 |
| 2 | Булатов Антон Игоревич | 3 | 4 | | | X | X | X | X | X | X | 4 |
| 3 | Буракова Юлия Андреевна | 4 | 4 | | | X | X | X | X | X | X | 4 |
| 4 | Валуева Евгения Павловн | 4 | 4 | | | X | X | X | X | X | X | 4 |
| 5 | Зацепин Игорь Сергеевич | 3 | 4 | | | X | X | X | X | X | X | 4 |
| 6 | Кирильшин Лев Сергеевич | 3 | 5 | | | X | X | X | X | X | X | 4 |
| 7 | Комаров Степан Владими | 4 | 4 | | | X | X | X | X | X | X | 4 |
| 8 | Копысов Михаил Игоревич | 3 | 4 | | | X | X | X | X | X | X | 4 |
| 9 | Лазаренко Степан Игорее | 4 | 4 | | | X | X | X | X | X | X | 4 |
| 10 | Макаров Антон Игоревич | 4 | 4 | | | X | X | X | X | X | X | 4 |
| 11 | Малков Сергей Валерьеви | 4 | 4 | | | X | X | X | X | X | X | 4 |
| 12 | Петренко Сергей Николае | 4 | 4 | | | X | X | X | X | X | X | 4 |
| 13 | Смирнов Александр Нико | 5 | 4 | | | X | X | X | X | X | X | 4 |
| 14 | Сысуйев Евгений Владими | 3 | | | | X | X | X | X | X | X | 4 |
| 15 | Шангин Александр Сергее | 4 | 5 | | | X | X | X | X | X | X | 5 |

По группе: Оценка 5 - 0, Оценка 3 - 0, Зачет - 0, Всего: Оценка 5 - 0, Оценка 3 - 0, Зачет - 0, Всего: Оценка 4 - 0, Оценка 2 - 0, Ср. балл - 0, Оценка 4 - 0, Оценка 2 - 0, Ср. балл - 0

Рисунок 2 – Окно отчета успеваемости обучающихся

Программа содержит опцию наглядного представления качественной успеваемости по группам (классам) и специальностям в виде графиков и диаграмм.

Программный модуль «Студенты» является частью информационной системы «УниверИС» [17].

Модуль "Студенты" предназначен для автоматизации работы со студентами на рабочем месте «Деканат»:

- работа с личными карточками студентов;
- работа с ведомостями;
- просмотр, ввод результатов выполнения студентами учебных планов;
- формирование и печать различных отчетов («Списки студентов», «Количество студентов», «Итоги успеваемости», «Показатели успеваемости», «Выход студентов на сессию» и т. д.).

Рассмотрим отчет «Показатели успеваемости по университету» (рисунок 3).

| N | Факультет | Количество | Факт. количество | процент |
|----|---------------------------------|------------|------------------|---------|
| 6 | Коммерции | 281 | 0 | 0 |
| 7 | Лингвистика | 77 | 0 | 0 |
| 8 | Международный | 108 | 0 | 0 |
| 9 | Механико-математический | 114 | 0 | 0 |
| 10 | Механико-технологический | 307 | 301 | 98 |
| 11 | Права и финансов | 313 | 0 | 0 |
| 12 | Приборостроительный | 390 | 375 | 96 |
| 13 | Психологии | 51 | 0 | 0 |
| 14 | Сервис и легкая промышленность | 193 | 0 | 0 |
| 15 | Социально-гуманитарный | 180 | 0 | 0 |
| 16 | Физико-металлургический | 231 | 0 | 0 |
| 17 | Физический | 48 | 0 | 0 |
| 18 | Экономики и предпринимательства | 171 | 0 | 0 |
| 19 | Экономики и управления | 396 | 173 | 43 |
| 20 | Энергетический | 283 | 0 | 0 |
| 21 | Юридический | 133 | 0 | 0 |
| | Итого по университету: | 4082 | 1180 | 28 |

Рисунок 3 – Окно отчета «Показатели успеваемости по университету»

Отчет содержит информацию о показателях успеваемости за выбранную сессию по всему университету, сгруппированных по факультету.

Отчет состоит из табличной и графической частей.

Модуль «Формирование отчетов об учебных результатах групп и студентов», входящий в Систему электронного обучения 3.5, обеспечивает формирование следующих видов отчетов:

- «отчет о результатах работы групп с цифровыми учебными материалами;
- отчет о результатах работы групп с оценочными средствами;
- отчет о результатах работы групп с материалами курса;
- отчет о средних результатах работы групп с цифровыми учебными материалами;
- аналитический отчет о результатах работы групп с оценочными средствами;
- отчет об активности пользователей» [12].

На рисунках 4 и 5 показаны отчет об успеваемости студента и аналитический отчет о результатах работы групп с оценочными средствами, соответственно.

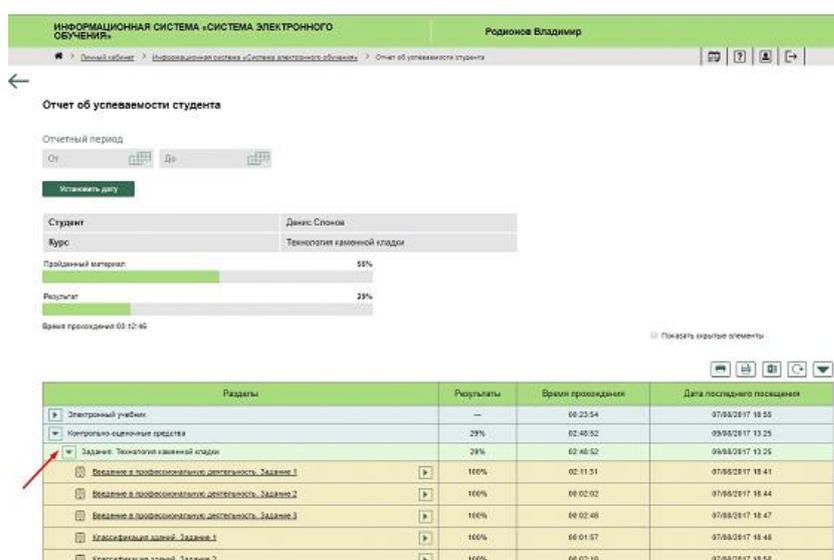


Рисунок 4 – Отчет об успеваемости студента

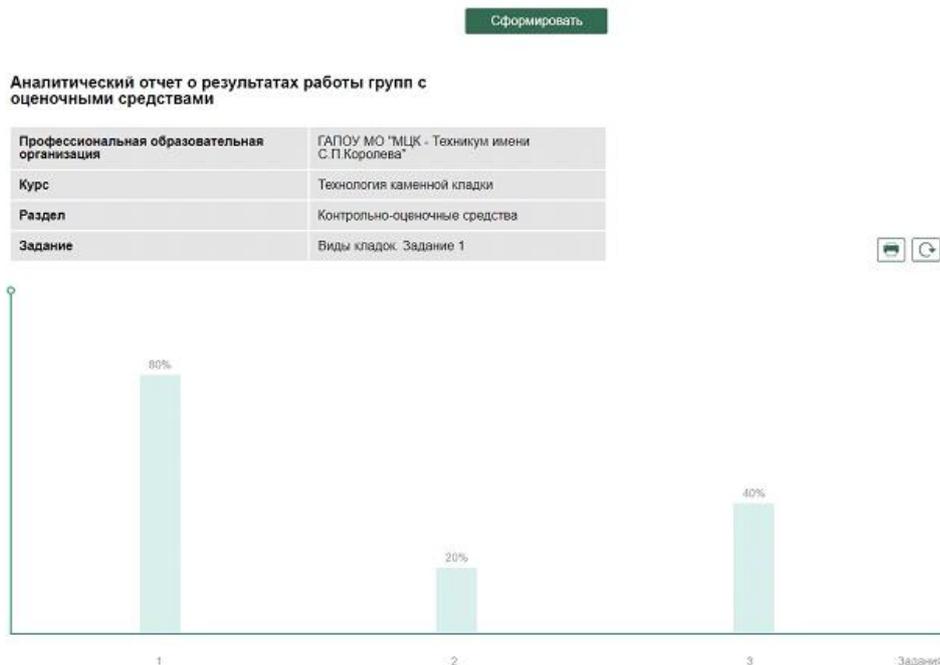


Рисунок 5 – Аналитический отчет о результатах работы групп с оценочными средствами

Для отчетов реализованы возможности печати, сохранения в формате .pdf, .xls, обновления, а также отчет может быть развернут во весь экран по всем позициям и свернут.

Для сравнения характеристик рассмотренных аналогов разработана таблица 2.

Таблица 2 – Сравнение характеристик ПО для управления отчетом по успеваемости студентов ТГУ

| Характеристика/балл (0-3) | Программа «Учебный учет» | Программный модуль «Студенты» | Модуль «Формирование отчетов об учебных результатах групп и студентов» |
|--|--------------------------|-------------------------------|--|
| Управление отчетом по успеваемости студентов ТГУ | 2 | 2 | 2 |
| Формирование отчета ранжирования студентов | 0 | 0 | 0 |

Продолжение таблицы 2

| Характеристика/балл (0-3) | Программа «Учебный учет» | Программный модуль «Студенты» | Модуль «Формирование отчетов об учебных результатах групп и студентов» |
|--------------------------------------|--------------------------|-------------------------------|--|
| Экспорт отчета | 1 | 1 | 2 |
| Использование технологий МО | 0 | 0 | 2 |
| Низкая совокупная стоимость владения | 2 | 2 | 1 |
| Итого | 3 | 3 | 5 |

Таким образом, все рассмотренные аналоги не удовлетворяют сформулированным требованиям.

Поэтому необходимо разработать новое ПО для управления отчетом по успеваемости студентов, отвечающую требованиям ТГУ.

Выводы к главе 1

Результаты выполнения главы 1 позволили сделать следующие выводы:

- как показал анализ, наиболее популярным методом ранжирования студентов является метод, основанный на применении среднего балла студента;
- главное преимущество метода АНР заключается в простоте его реализации с помощью алгоритмов машинного обучения.

Как показал анализ, все рассмотренные аналоги не удовлетворяют сформулированным требованиям.

Поэтому необходимо разработать новое ПО для управления отчетом по успеваемости студентов, отвечающую требованиям ТГУ.

Глава 2 Проектирование ПО для управления отчетом по успеваемости студентов ТГУ

2.1 Логическое проектирование ПО

В качестве методологии логического проектирования ПО используем методологию RUP [3].

RUP (Rational Unified Process) — это платформа итеративного процесса разработки ПО.

Это не единый конкретный предписывающий процесс, а, скорее, адаптируемая структура процесса, предназначенная для адаптации организаций-разработчиков и групп разработчиков программного обеспечения к их потребностям [7].

Для моделирования в RUP применяются CASE-средства, основанные на языке UML [15].

RUP представляет собой конкретную реализацию унифицированного процесса и включает в себя следующие характеристики [5]:

- на основе вариантов использования: RUP построен на использовании диаграммы вариантов использования для определения функциональных требований;
- ориентированность на архитектуру: требует разработки исполняемой архитектуры на ранних этапах проекта (диаграммы классов и/или компонентов);
- итеративность и поэтапное выполнение: разработка организована в виде серии коротких мини-проектов фиксированной длины (например, трехнедельных), называемых итерациями. Результатом каждого из них является протестированная, интегрированная и исполняемая система;
- фазы: RUP разделен на четыре фазы: создание, разработка, создание и переход. Каждая фаза имеет конкретные цели и завершается четко

определенной вехой.

Диаграмма вариантов использования UML обобщает взаимодействие между системой и ее пользователями. Варианты диаграммы использования могут быть составлены для общего сценария или нескольких явлений, которые инкапсулируют функционирование системы [11].

На диаграмме варианты использования представлены три типа сущностей:

- система – рассматриваемый продукт, находящийся на стадии разработки. Она представлена рамкой, в которой указаны варианты использования;
- акторы – пользователи или системы, которые взаимодействуют с основной системой. Актор может быть первичным или второстепенным. Первичные акторы — это те, кто инициирует варианты использования и взаимодействует с системой. Обычно они обращаются в левую часть системы. Вторичные акторы не взаимодействуют с системой самостоятельно. Они изображаются в правой части системы;
- варианты использования: функции, выполняемые участниками при взаимодействии с системой.

В процессе разработки диаграммы вариантов использования ПО были выделен актер Пользователь.

Описание вариантов использования ИСЭЗУП представлено в таблицах 3-6.

Таблица 3 – Описание прецедента: Выбрать датасет

| «Элемент диаграммы | Описание |
|--------------------|-------------------------|
| Прецедент | Выбрать датасет |
| ID | 1 |
| Краткое описание | Выбрать входной датасет |
| Главный актер | Пользователь» [11] |

Продолжение таблицы 3

| «Элемент диаграммы | Описание |
|-----------------------|--|
| Второстепенный актер | Нет |
| Предусловие | Нет |
| Основной поток | Пользователь открывает диалоговое окно и выбирает входной датасет показателей успеваемости студентов |
| Постусловие | Нет |
| Альтернативные потоки | Нет» [11] |

Таблица 4 – Описание прецедента: Сформировать группы ранжирования студентов

| «Элемент диаграммы | Описание |
|-----------------------|---|
| Прецедент | Сформировать группы ранжирования студентов |
| ID | 2 |
| Краткое описание | Сформировать группы ранжирования студентов по среднему балу |
| Главный актер | Пользователь |
| Второстепенный актер | Нет |
| Предусловие: | Нет |
| Основной поток | Пользователь формирует группы ранжирования студентов по среднему балу |
| Постусловие | Нет |
| Альтернативные потоки | нет» [11] |

Таблица 5 – Описание прецедента: Определить рейтинг студента

| «Элемент диаграммы | Описание |
|-----------------------|--|
| Прецедент | Определить рейтинг студента |
| ID | 3 |
| Краткое описание | Определение рейтинга студента |
| Главный актер | Пользователь |
| Второстепенный актер | Нет |
| Предусловие | Создание группы ранжирования студентов по среднему балу |
| Основной поток | Пользователь вводит данные цифрового следа студента и определяет его рейтинг |
| Постусловие | Нет |
| Альтернативные потоки | Нет» [11] |

Таблица 6 – Описание прецедента: Экспорт отчета

| «Элемент диаграммы | Описание |
|-----------------------|---|
| Прецедент | Экспорт отчета |
| ID | 4 |
| Краткое описание | Экспорт отчета |
| Главный актер | Пользователь |
| Второстепенный актер | Нет |
| Предусловие | Нет |
| Основной поток | Пользователь создает отчет в формате HTML |
| Постусловие: | Нет |
| Альтернативные потоки | Нет» [11] |

Разработанная диаграмма вариантов использования программы показана на рисунке 5.

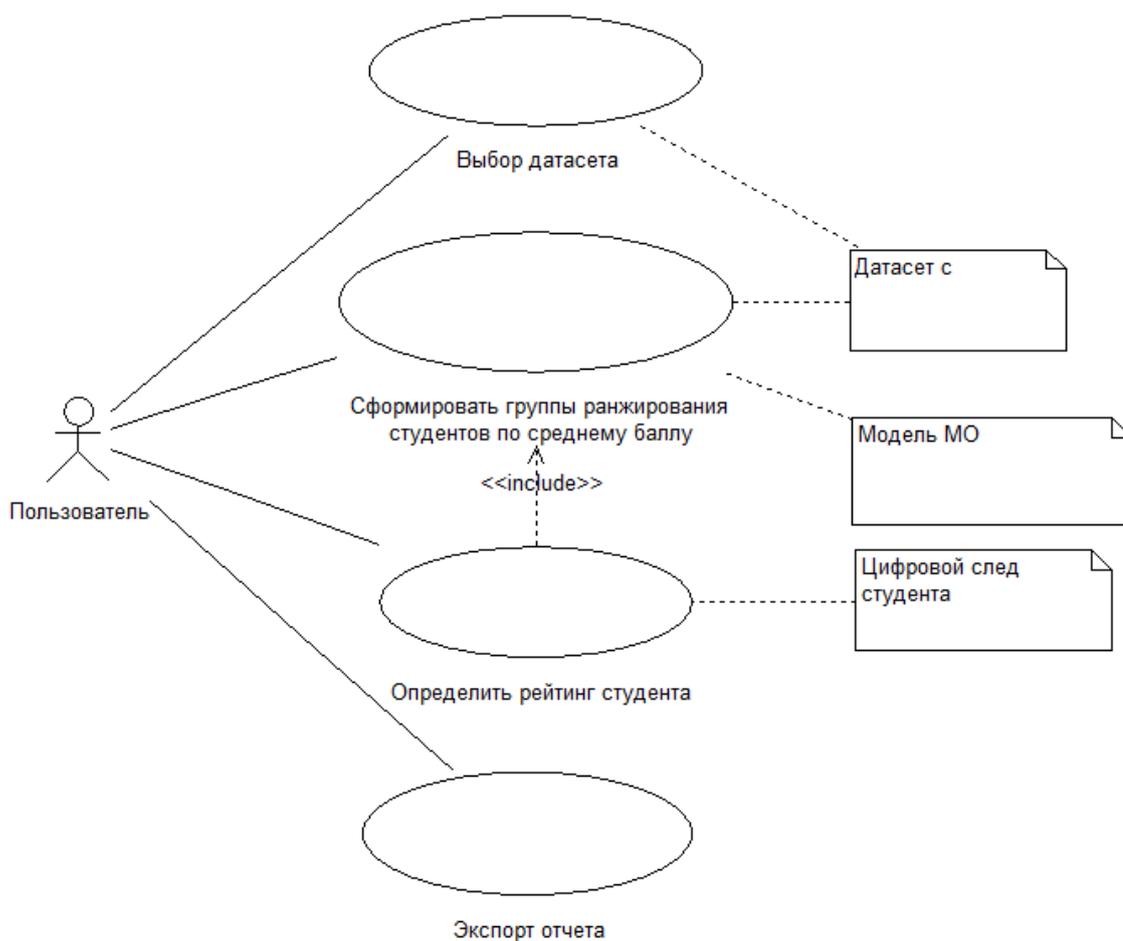


Рисунок 5 – Диаграмма вариантов использования программы для управления отчетом по успеваемости студентов ТГУ

Диаграмма вариантов использования отражает функционал разработанной программы.

В программной инженерии диаграмма классов — это тип диаграмм UML, которые показывают структуру ПО с точки зрения ее классов, атрибутов и методов. Это также показывает разработчику отношения, существующие между классами.

В первую очередь программа классов была создана для того, чтобы разработчики могли понять требования к ПО и описать детальный проект.

Например, на этапах анализа и проектирования цикла разработки разработчик может создавать диаграммы классов, чтобы отобразить структуру классов и определить отношения между классами и классификаторами.

Диаграмма классов программы показана на рисунке 6.

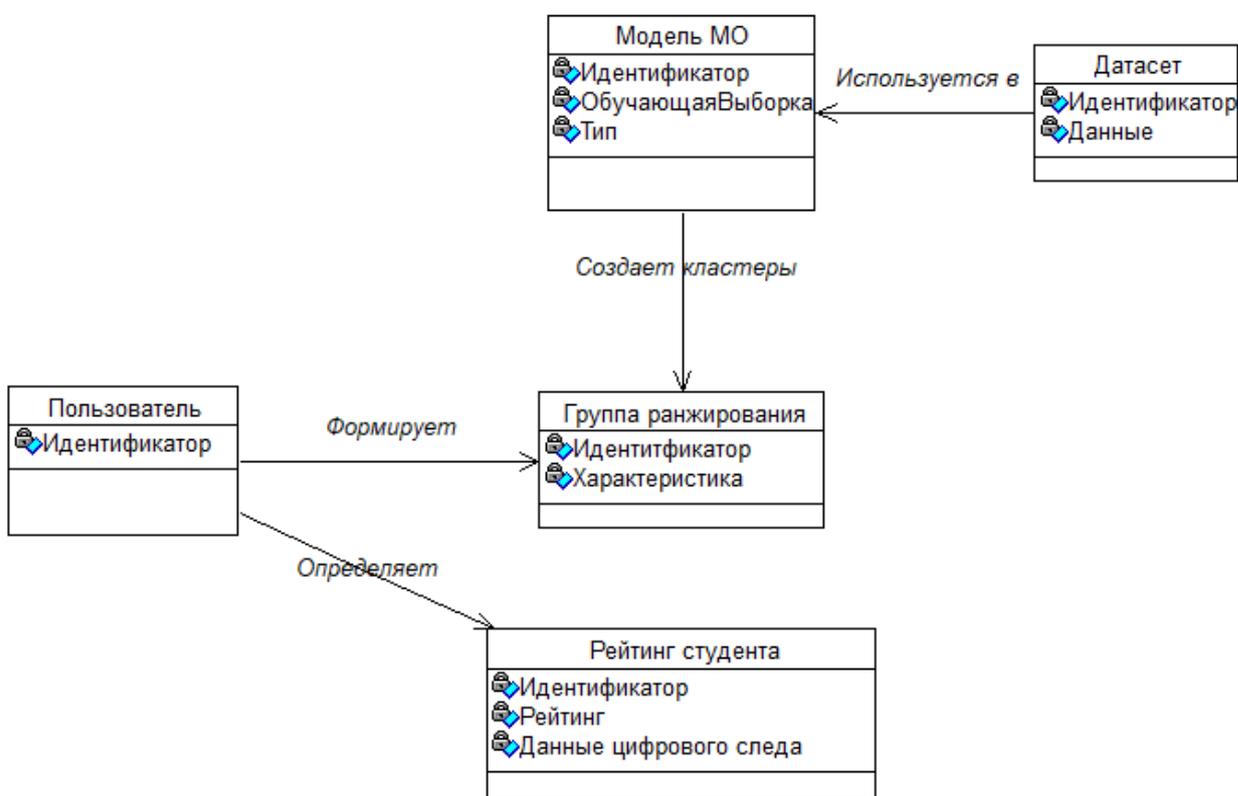


Рисунок 5 – Диаграмма классов программы для управления отчетом по успеваемости студентов ТГУ

Спецификация диаграммы классов представлена в таблице 7.

Таблица 7 – Спецификация диаграммы классов программы для управления отчетом по успеваемости студентов ТГУ

| Класс | Описание |
|---------------------|---|
| «Пользователь | Класс объектов-пользователей ПО |
| Модель МО | Класс объектов-моделей машинного обучения |
| Датасет | Класс объектов-входных наборов данных |
| Группа ранжирования | Класс объектов-групп ранжирования студентов по среднему баллу |
| Рейтинг студента | Класс объектов-рейтингов студентов» [11] |

Представленные диаграммы разработаны с помощью CASE-средства Rational Rose.

2.2 Алгоритм управления отчетом

Для представления алгоритма управления отчетом по успеваемости студентов ТГУ используем диаграмму деятельности UML.

Диаграмма деятельности по сути является расширением диаграммы рабочего процесса. Она описывает динамические аспекты системы и считается одной из наиболее важных особенностей UML. Диаграммы деятельности представляют собой поток одного действия из другого.

Деятельность описывается как рабочий процесс системы. Таким образом, диаграмма деятельности представляет собой блок-схему действий.

Основные состояния диаграммы деятельности:

- стартовое состояние. Это состояние представляет собой начало диаграммы активности;
- активность. Она описывается как рабочий процесс системы. Действия начинаются в начальном узле и заканчиваются в каком-то конечном узле;
- узел принятия решения. Рамка в форме ромба, используемая для принятия решения в любом узле активности. У него есть два

результата: да или нет;

- окончательное состояние. Он используется для иллюстрации окончания процесса/когда процесс должен быть завершен;
- разветвление и слияние.

На рисунке 6 показан диаграмма деятельности управления отчетом по успеваемости студентов ТГУ.

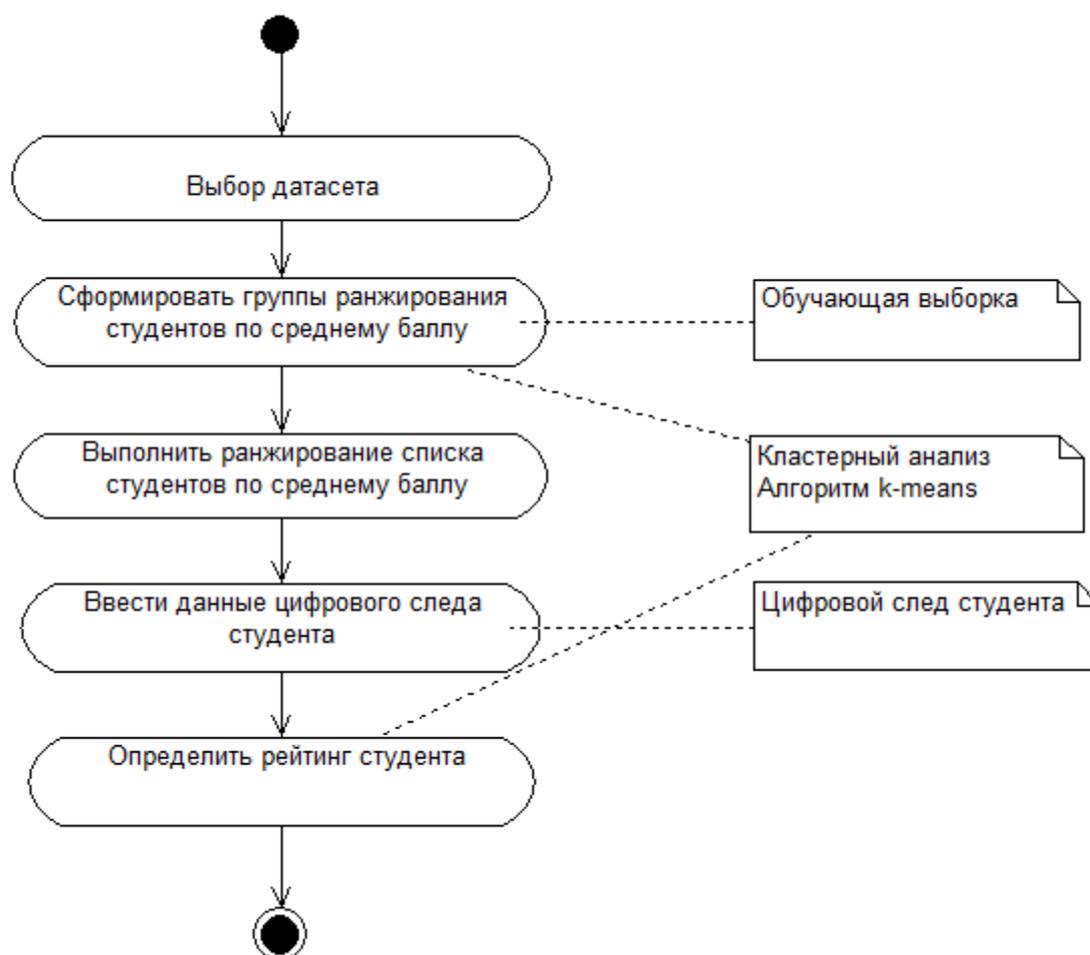


Рисунок 6 – Диаграмма деятельности управления отчетом по успеваемости студентов ТГУ

Как следует из диаграмм, для формирования групп ранжирования используется метод кластеризации с помощью алгоритма *k-means*.

«Алгоритм *k-means* — это итеративный алгоритм, который пытается разделить набор данных на заранее определенные *k*-отдельных

неперекрывающихся групп-кластеров, где каждая точка данных принадлежит только одной группе» [9].

Блок-схема алгоритма k-means показана на рисунке 7.

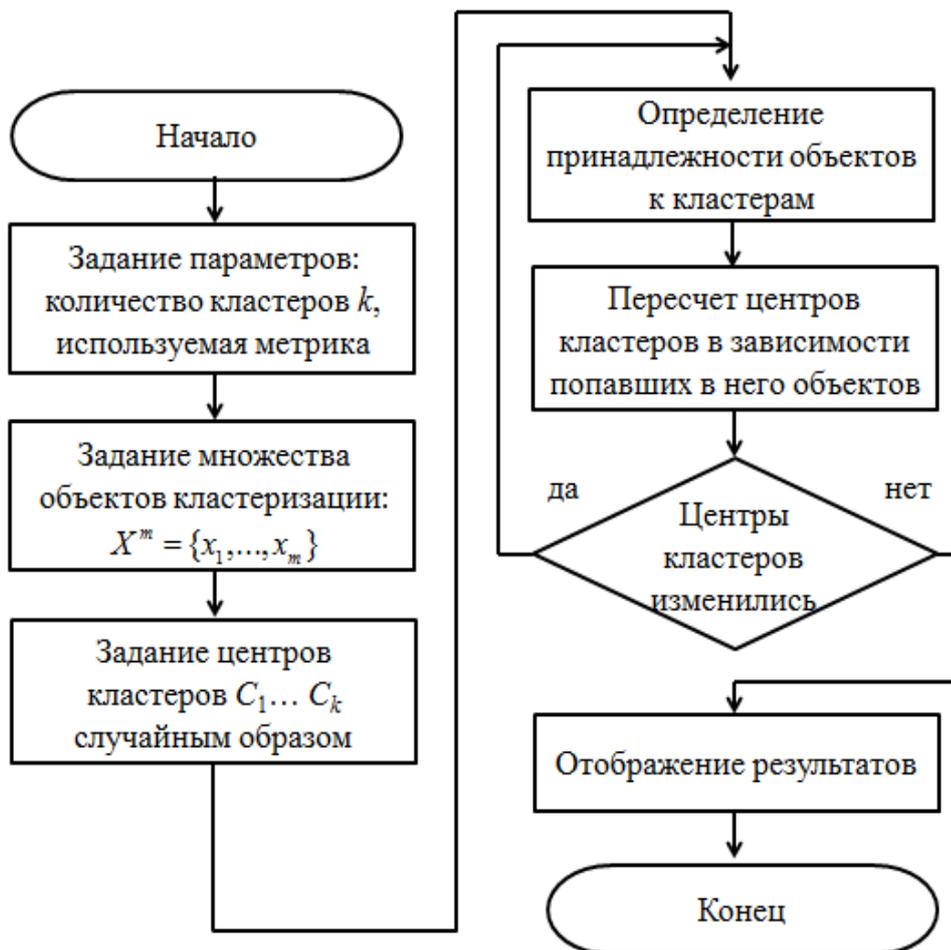


Рисунок 7 – Блок-схема алгоритма k-means

Алгоритм k-means пытается сделать точки данных внутри кластера как можно более похожими, но при этом сохраняя кластеры как можно более разными.

Он назначает точки данных кластеру таким образом, чтобы сумма квадратов расстояния между точками данных и центроидом кластера (среднем арифметическим всех точек данных, принадлежащих этому кластеру) была минимальной.

Чем меньше вариаций внутри кластеров, тем более однородные (похожие) точки данных находятся в одном кластере.

Подход, который использует k-means для решения проблемы, называется EM (Expectation-Maximization)-алгоритмом.

Главным преимуществом алгоритма k-means является высокая производительность и простота реализации.

Выводы к главе 2

Результаты выполнения главы 2 позволили сделать следующие выводы:

- RUP представляет собой конкретную реализацию унифицированного процесса проектирования;
- диаграмма вариантов использования UML обобщает взаимодействие между системой и ее пользователями;
- диаграмма деятельности описывает динамические аспекты системы и считается одной из наиболее важных особенностей UML.

Главным преимуществом алгоритма k-means является высокая производительность и простота реализации.

Глава 3 Реализация и тестирование ПО для управления отчетом по успеваемости студентов ТГУ

3.1 Выбор средства разработки ПО

Для реализации алгоритма машинного обучения выбран язык Python.

Для выбора среды разработки сравним характеристики двух сред разработки: Jupyter Notebook и Python IDLE.

Рассмотрим возможности среды Jupyter Notebook [23].

Первоначально Jupyter Notebook был разработан как инструмент для анализа данных и научных вычислений. Он приобрел популярность среди ученых и исследователей данных благодаря своей интерактивной природе и возможности сочетать код, текст и визуализацию в одном документе.

Jupyter Notebook имеет несколько функций, которые делают его популярным инструментом для специалистов по данным и исследователей:

- интерактивный пользовательский интерфейс;
- выполнение кода в режиме онлайн;
- интеграция с несколькими языками программирования;
- легкое сотрудничество и обмен данными;
- визуализация данных и результатов;
- поддержка языка разметки и др.

Jupyter Notebook имеет ряд преимуществ и недостатков по сравнению с другими IDE, в том числе:

- интерактивное и исследовательское кодирование;
- визуализация и анализ данных в реальном времени;
- зависимость от веб-браузера;
- ресурсоемкий и медленный для больших наборов данных;
- ограниченные инструменты отладки;
- трудности с контролем версий;
- проблемы безопасности при запуске кода из ненадежных источников

и др.

Рассмотрим возможности среды Python IDLE [24].

Python IDLE – интегрированная среда разработки с языком программирования Python и базовым интерфейсом, который позволяет пользователям писать, выполнять и отлаживать код Python.

Python IDLE имеет несколько функций, которые делают его популярным среди начинающих и опытных разработчиков Python:

- базовый пользовательский интерфейс;
- инструменты отладки;
- подсветка синтаксиса;
- автозаполнение и др.

Преимущества и недостатки среды Python IDLE:

- простота и скорость;
- интегрированные инструменты отладки;
- ограниченные возможности и функциональность;
- нет поддержки совместной работы;
- ограниченная поддержка визуализации данных;
- нет поддержки языка разметки;
- нет интеграции с другими языками программирования.

Для сравнения рассмотренных сред программирования используем таблицу 8.

Таблица 8 – Сравнение характеристик среда разработки на языке Python

| Характеристика | Jupyter Notebook | Python IDLE |
|-------------------|--|--|
| Интерфейс | Веб-интерфейс | Десктоп-приложение |
| Вывод | отображается в строке | в отдельном окне консоли |
| Совместная работа | Встроенная поддержка совместного использования и совместной работы | Нет встроенной функции совместной работы |
| Организация кода | Код организован в ячейках | Код, записан в одном файле |
| Отладка | Доступна ограниченная поддержка | Доступна поддержка отладки |

Продолжение таблицы 8

| Характеристика | Jupyter Notebook | Python IDLE |
|------------------------|---|---|
| Визуализация | Отличная поддержка | Ограниченная поддержка |
| Кривая обучения | Крутая кривая обучения | Легко изучается |
| Варианты использования | Интерактивный анализ данных, создание прототипов и визуализация | Написание небольших скриптов и изучение синтаксиса Python |

По результатам сравнения и с учетом предпочтений разработчика выбираем среду разработки Jupyter Notebook.

Для поддержки машинного обучения используем библиотеку Scikit-learn [13].

Scikit-learn — это библиотека анализа данных с открытым исходным кодом и золотой стандарт машинного обучения в экосистеме Python [26].

Ключевые концепции и функции включают в себя алгоритмические методы принятия решений, в том числе:

- классификация: идентификация и категоризация данных на основе закономерностей;
- регрессия: прогнозирование или прогнозирование значений данных на основе среднего значения существующих и запланированных данных;
- кластеризация: автоматическая группировка похожих данных в наборы данных;
- взаимодействие с библиотеками numPy, pandas и matplotlib и др.

Scikit-learn включает алгоритмы, поддерживающие прогнозный анализ, которые варьируются от простой линейной регрессии до распознавания образов нейронной сети.

Для проведения тестирования разработанного ПО использован метод функционального тестирования [21].

3.2 Реализация и тестирование ПО

В качестве входного датасета использованы данные средних баллов успеваемости студентов в файле CSV, для выбора которого используется диалоговое окно (рисунки 8,9).

```
1 from sklearn.cluster import KMeans
2 import pandas as pd
3 import numpy as np
4 import pickle
5 import pathlib

1 from sklearn.cluster import KMeans
2 from tkinter import filedialog as fd
3 from pathlib import Path
4 selected_file = Path(fd.askopenfile(mode='r', filetypes=[('Datasets', '*.csv')]).name)
5 print(type(selected_file), selected_file)
6 p=selected_file
```

Рисунок 8 – Код диалога для выбора входного датасета

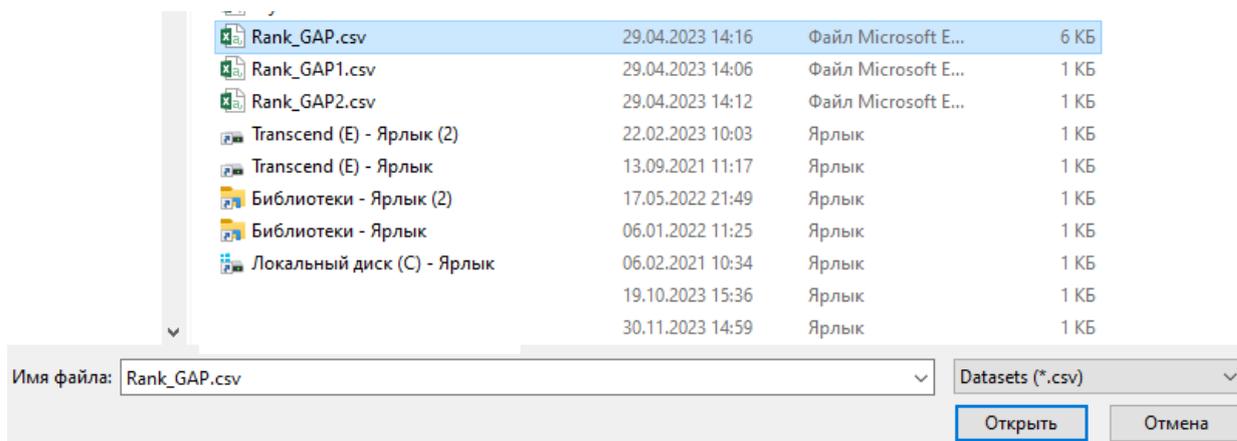


Рисунок 9 – Окно выбора входного датасета

Выбранный датасет будет использован для создания обучающей выборки для модели обучения кластеризации.

На рисунке 10 показаны основные показатели выбранного датасета.

```
Ввод [3]: 1 indata = pd.read_csv(p)
          2 indata.describe()
```

Out[3]:

| | x | y |
|-------|------------|------------|
| count | 860.000000 | 860.000000 |
| mean | 3.579535 | 3.434884 |
| std | 0.969812 | 1.163749 |
| min | 2.000000 | 2.000000 |
| 25% | 2.600000 | 2.000000 |
| 50% | 3.800000 | 4.000000 |
| 75% | 4.400000 | 4.000000 |
| max | 5.000000 | 5.000000 |

Рисунок 10 – Основные показатели выбранного входного датасета

Построена тепловая карта входного датасета.

Тепловые карты визуализируют данные посредством вариаций цвета.

При применении к табличному формату тепловые карты полезны для перекрестного анализа многомерных данных путем размещения переменных в строках и столбцах и раскрашивания ячеек в таблице [20].

Тепловые карты хороши для отображения различий между несколькими переменными, выявления любых закономерностей, отображения того, похожи ли какие-либо переменные друг на друга, а также для обнаружения наличия каких-либо корреляций.

Обычно все строки тепловой карты относятся к одной категории (метки отображаются слева или справа), а все столбцы относятся к другой категории (метки отображаются сверху или внизу). Отдельные строки и столбцы разделены на подкатегории, которые совпадают друг с другом в матрице. Ячейки представляют собой пересечения строк и столбцов, которые могут содержать категориальные данные или числовые данные.

На рисунке 11 показана тепловая карта признаков набора данных.



Рисунок 11 – Тепловая карта признаков набора данных

Реализован и протестирован алгоритм k-means для кластеризации данных (рисунок 12).

```
kmeans = KMeans(4)
kmeans.fit(indata.values)
print(kmeans.cluster_centers_)
```

Рисунок 12 – Код реализации алгоритма k-means

Для визуализации результатов кластеризации разработан код, показанный на рисунке 13.

```

2 import matplotlib.pyplot as plt
3 from matplotlib.colors import ListedColormap
4 %matplotlib inline
5 customcmap = ListedColormap(["crimson", "mediumblue", "darkmagenta"])
6 fig, ax = plt.subplots(figsize=(10, 8))
7 plt.scatter(x=indata['x'], y=indata['y'], s=200,
8             c=kmeans.labels_,
9             cmap=customcmap)
10 ax.set_xlabel(r'x', fontsize=14)
11 ax.set_ylabel(r'y', fontsize=14)
12 plt.xticks(fontsize=12)
13 plt.yticks(fontsize=12)
14 plt.show()

```

Рисунок 13 – Код реализации алгоритма k-means

Полученный график показан на рисунке 14.

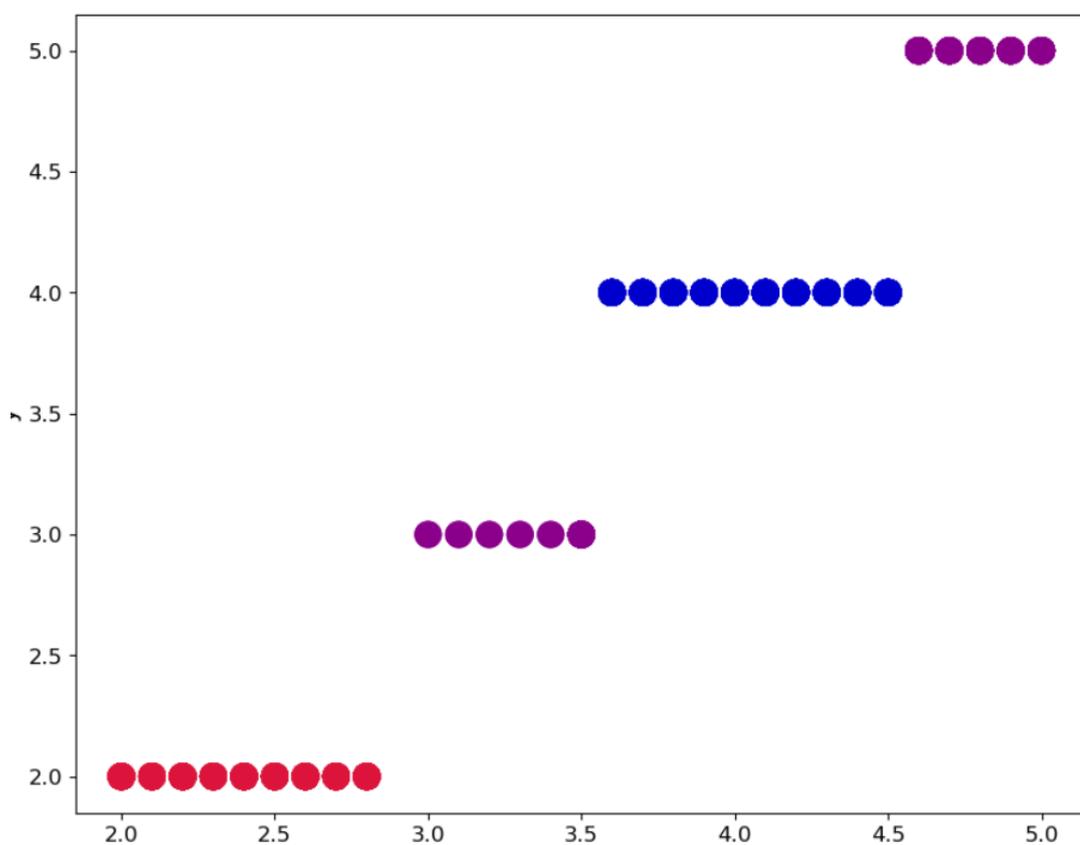


Рисунок 14 – График результатов кластеризации

Как следует из графика, из обучающей выборки были выделены 4 группы ранжирования студентов:

- отличники (средний балл = 5);
- хорошисты (средний балл = 4);
- троечники (средний балл = 3);
- неуспевающие (средний балл = 2).

Важной особенностью алгоритма k-means является наличие метода `predict()` для предсказания вхождения объекта в определенный кластер и определения его рейтинга [8].

На рисунке 15 показан код и результат определения рейтинга студента по данным его цифрового следа.

```
6 print(sf)
7 predicted_class = kmeans.predict([sf])
8 print(predicted_class)

[[2.4      2.      ]
 [4.05     4.      ]
 [4.8      5.      ]
 [3.43181818 3.      ]]
Введите показатели студента:
3
5
[3.0, 5.0]
[1]
```

Рисунок 15 – Результат определения рейтинга студента

Для построения отчета по успеваемости студентов использована библиотека `Sweetviz`.

`Sweetviz` — это библиотека Python с открытым исходным кодом, которая генерирует красивые визуализации высокой плотности для запуска EDA (исследовательский анализ данных) с помощью всего двух строк кода. Результатом является полностью автономное HTML-приложение [28].

Метод построен на быстрой визуализации целевых значений и сравнении наборов данных.

Ее цель — помочь быстро проанализировать целевые характеристики, данные обучения и тестирования и другие подобные задачи по описанию характеристик данных.

На рисунке 16 показан код формирования отчета в виде файла HTML.

```
1 !pip install sweetviz
2 import sweetviz as sv
3 hd=pd.read_csv(p)
4 report = sv.analyze(hd)
5 report.show_html("Perform_rep.html")
```

Рисунок 16 – Код формирования отчета в виде файла HTML

Фрагмент скриншота страницы отчет показан на рисунке 17.

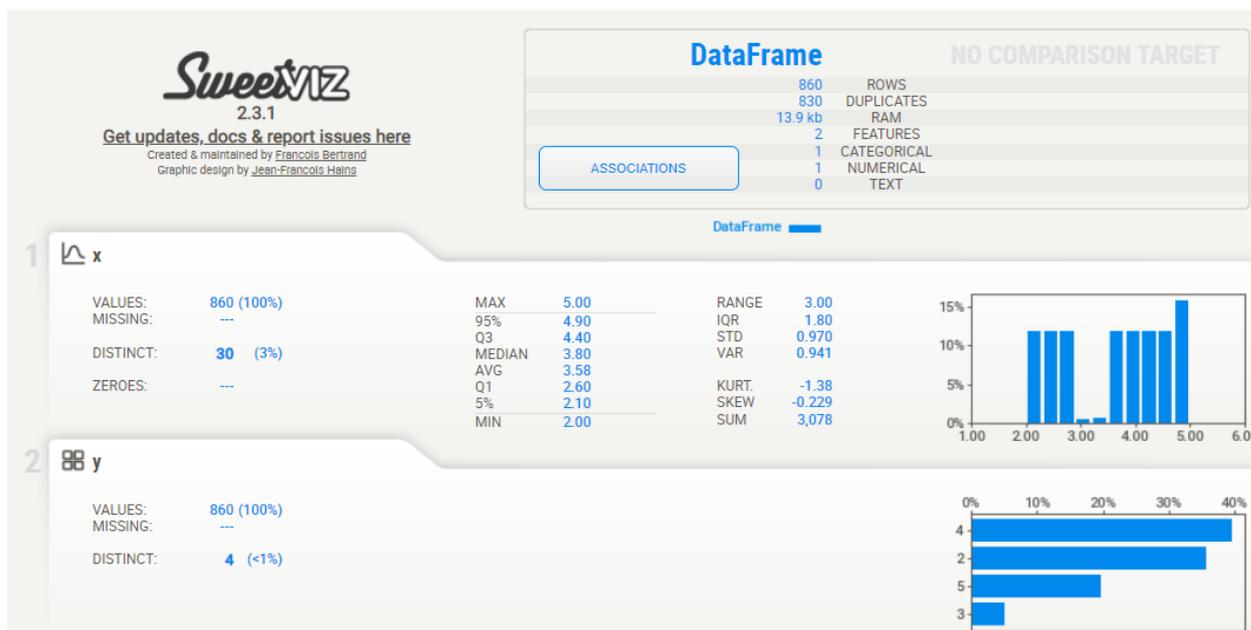


Рисунок 17 – Код формирования отчета в виде файла HTML

Таким образом, проведенное функциональное тестирование подтвердило работоспособность разработанного ПО для управления отчетом по успеваемости студентов ТГУ.

Выводы к главе 3

Результаты выполнения главы 3 позволили сделать следующие выводы:

- по результатам сравнения и с учетом предпочтений разработчика для разработки ПО выбрана среда разработки Jupyter Notebook. Для поддержки машинного обучения использована библиотека Scikit-learn;
- для проведения тестирования разработанного ПО использован метод функционального тестирования.

Проведенное функциональное тестирование подтвердило работоспособность разработанного ПО для управления отчетом по успеваемости студентов ТГУ.

Заключение

Бакалаврская работа посвящена актуальной проблеме разработки ПО для управления отчетом по успеваемости студентов ТГУ.

Выполненные в рамках бакалаврской работы задачи представлены следующими основными результатами:

- произведена постановка задачи на разработку и сформулированы требования к ПО для управления отчетом по успеваемости студентов ТГУ. Как показал анализ, наиболее популярным методом ранжирования студентов является метод, основанный на применении среднего балла студента. Главное преимущество метода АНР заключается в простоте его реализации с помощью алгоритмов машинного обучения. Как показал анализ, все рассмотренные аналоги не удовлетворяют сформулированным требованиям. Поэтому необходимо разработать новое ПО для управления отчетом по успеваемости студентов, отвечающую требованиям ТГУ;
- спроектировано ПО для управления отчетом по успеваемости студентов ТГУ. Как показал анализ, методология RUP представляет собой конкретную реализацию унифицированного процесса проектирования. Диаграмма вариантов использования UML обобщает взаимодействие между системой и ее пользователями. Диаграмма деятельности описывает динамические аспекты системы и считается одной из наиболее важных особенностей UML. Главным преимуществом алгоритма k-means является высокая производительность и простота реализации;
- выполнены реализация и тестирование ПО для управления отчетом по успеваемости студентов ТГУ. По результатам сравнения и с учетом предпочтений разработчика для разработки ПО выбрана среда разработки Jupyter Notebook. Для поддержки машинного обучения использована библиотека Scikit-learn. Для тестирования

разработанного ПО использован метод функционального тестирования. Для построения отчета по успеваемости студентов использована библиотека Sweetviz. Это библиотека Python с открытым исходным кодом, которая генерирует красивые визуализации высокой плотности для запуска EDA (исследовательский анализ данных) с помощью всего двух строк кода. Результатом является полностью автономное HTML-приложение.

Проведенное функциональное тестирование подтвердило работоспособность разработанного ПО для управления отчетом по успеваемости студентов ТГУ.

Результаты бакалаврской работы могут представлять интерес для разработчиков программного обеспечения на языке Python, использующих среду разработки Jupyter Notebook.

Список используемой литературы и используемых источников

1. Активное обучение ранжированию [Электронный ресурс]: Машинное обучение. Ранжирование URL: <https://habr.com/ru/companies/vk/articles/461927/> (дата обращения: 25.03.2024).
2. Алгоритм составления рейтинга студентов для назначения государственных академических стипендий в повышенном размере за учебную, научно-исследовательскую, общественную, культурно-творческую и спортивную деятельность [Электронный ресурс]: Описание алгоритма. URL: https://news.sfu-kras.ru/files/algorithm_s_ispravleniyami.pdf (дата обращения: 25.03.2024).
3. Белик А. Г., Цыганенко В. Н. Проектирование и архитектура программных систем : учеб. пособие. Омск : Изд-во ОмГТУ, 2016. 96 с.
4. Большие данные: доказательное развитие образования [Электронный ресурс]: Цифровая трансформация школы. URL: <https://rffi.1sept.ru/article/337> (дата обращения: 25.03.2024).
5. Виноградова М. В., Белоусова В.И. Унифицированный процесс разработки программного обеспечения : учебное пособие. М: МГТУ им. Н.Э. Баумана, 2015. 82 с. URL: <https://www.iprbookshop.ru/134990.html> (дата обращения: 10.04.2024).
6. Горбаченко В. И., Савенков К.Е., Малахов М.А. Машинное обучение : учебное пособие. М.: Ай Пи Ар Медиа, 2023. 217 с. URL: <https://www.iprbookshop.ru/125886.html> (дата обращения: 02.04.2024).
7. Грекул В.И., Денищенко Г. Н., Коровкина Н.Л. Проектирование информационных систем : учебное пособие. М. : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. URL: <https://www.iprbookshop.ru/97577.html> (дата обращения: 25.03.2024).
8. Демидова Л. А. Кластерный анализ. Python : учебное пособие. М. : РТУ МИРЭА, 2022. 103 с. URL: <https://e.lanbook.com/book/240092> (дата обращения: 19.02.2024).

9. Кластеризация: алгоритмы k-means и c-means [Электронный ресурс <https://habr.com/ru/articles/67078/>] (дата обращения: 19.02.2024).

10. Коцюба И.Ю., Чунаев А.В., Шиков А.Н. Методы оценки и измерения характеристик информационных систем. Учебное пособие. СПб: Университет ИТМО, 2015. 264 с.

11. Леоненков А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose : учебное пособие. М. : ИНТУИТ, Ай Пи Ар Медиа, 2020. 317 с. URL: <https://www.iprbookshop.ru/97554.html> (дата обращения: 15.02.2024).

Методы кластеризации. URL: https://ai-news.ru/2018/07/klasterizaciya_algoritmy_k_means_i_c_means.html (дата обращения: 20.03.2024).

12. Модуль «Формирование отчетов об учебных результатах групп и студентов» [Электронный ресурс]: Описание модуля. URL: https://elearning.academia-moscow.ru/seo/help/module_5_3_7.php (дата обращения: 25.03.2024).

13. Накарякова Н.Н., Русаков С.В., Русакова О.Л. Прогнозирование группы риска (по успеваемости) среди студентов первого курса с помощью дерева решений // Прикладная математика и вопросы управления. 2020. № 4. С. 121-136.

14. Нефункциональные требования к программному обеспечению [Электронный ресурс]. URL: <https://habr.com/ru/articles/231961/> (дата обращения: 15.03.2024).

15. Носова Л. С. Case-технологии и язык UML : учебно-методическое пособие. Челябинск, Саратов : Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. 67 с. URL: <https://www.iprbookshop.ru/81479.html> (дата обращения: 11.04.2024).

16. Программа «Учебный учет» [Электронный ресурс]: Описание программы. URL: <https://pbprog.ru/catalog/all/411> (дата обращения: 25.03.2024).

17. Программный модуль «Студенты» [Электронный ресурс]: Описание модуля. URL: <https://univerisupdate.susu.ru/files/help/Student/naznach.htm> (дата обращения: 25.03.2024).

18. Сайт ТГУ. [Электронный ресурс]: Сайт Тольяттинского государственного университета. URL: <https://www.tltsu.ru/> (дата обращения: 25.03.2024).

19. Стариченко Б. Е. Балльно-рейтинговая система оценивания учебной деятельности студентов: вопросы моделирования // Педагогическое образование в России. 2017. № 6. С. 205-215.

20. Тепловая карта [Электронный ресурс]: Анализ данных. URL: <https://drozd.red/glossary/teplovaya-karta/ru/> (дата обращения: 20.03.2024).

21. Функциональное тестирование ПО [Электронный ресурс]: Задачи, виды, методы проведения. URL: <https://gb.ru/blog/funktsionalnoe-testirovanie-po/> (дата обращения: 25.03.2024).

22. Цифровой след студентов IT-направлений [Электронный ресурс]: РФФИ. URL: https://www.rfbr.ru/rffi/ru/science_news/o_2133692 (дата обращения: 25.03.2024).

23. Project Jupyter [Электронный ресурс]: Описание среды. URL: <https://jupyter.org/> (дата обращения: 20.03.2024).

24. Python IDLE [Электронный ресурс]: Документация. URL: <https://docs.python.org/3/library/idle.html> (дата обращения: 20.03.2024).

25. Rating Assessment of Students' Knowledge vis-a-vis Rating Evaluation of University Teachers. URL: <https://www.igi-global.com/chapter/rating-assessment-of-students-knowledge-vis-a-vis-rating-evaluation-of-university-teachers/236035> (дата обращения: 20.03.2024).

26. Scikit-learn 1.4.1.post1 [Электронный ресурс]: Описание проекта. URL: <https://pypi.org/project/scikit-learn/> (дата обращения: 20.03.2024).

27. Suhailan Safei et al. Targeted Ranking-Based Clustering Using ANP K-Means // Int. J. Advance Soft Compu. Appl, 2015. Vol. 7, No. 3. P. 100-113.

28. Sweetviz 2.3.1 [Электронный ресурс]: Описание проекта. URL:

<https://pupi.org/project/sweetviz/> (дата обращения: 20.03.2024).

29. What Is a GPA and Why Is It So Important? [Электронный ресурс]: Средний балл студента. URL: <https://www.mastersportal.com/articles/2126/what-is-a-gpa-and-why-is-it-so-important.html> (дата обращения: 25.03.2024).

30. What is Ranking Level? [Электронный ресурс]: Ранжирование. URL: <https://support.fedena.com/support/solutions/articles/53900-what-is-ranking-level-> (дата обращения: 25.03.2024).

31. Xiaomei Bai, Fuli Zhang, Jinzhou Li, Teng Guo, Abdul Aziz, Aijing Jin, Feng Xia “Educational Big Data: Predictions, Applications and Challenges”, Big Data Research, Vol. 26, 2021, 100270.