МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК
Кафедра программного обеспечения

РЕКОМЕНДОВАНО К ЗАЩИТЕ В ГЭК
Заведующий кафедрой
К.т.н., доцент
_____ М. С. Воробьёва
23.06. 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
магистерская диссертация

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ АНАЛИЗА РЕЗУЛЬТАТОВ ИТОГОВОЙ АТТЕСТАЦИИ УЧАЩИХСЯ ТЮМЕНСКОЙ ОБЛАСТИ

02.04.03 Математическое обеспечение и администрирование информационных систем
Магистерская программа «Разработка технологий Интернета вещей и больших данных»

Выполнил работу
студент 2 курса
очной формы обучения — Бачурин Роман Михайлович

Научный руководитель
К.п.н.,
доцент кафедры ПО — Плотоненко Юрий Анатольевич

Рецензент
Д.ф.-м.н., заместитель директора по развитию Института математики и компьютерных наук ТюмГУ — Шевляков Артём Николаевич

Тюмень
2023

ОГЛАВЛЕНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

API (Application Programming Interface) – Описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

BSON (Binary JavaScript Object Notation) – Бинарная форма представления простых структур данных и ассоциативных массивов.

CSS (Cascading Style Sheets) – Формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

DB (Database) – База данных.

ETL (Extract, Transform, Load) — Дословно «извлечение, преобразование, загрузка». Один из основных процессов в управлении хранилищами данных.

HTML (HyperText Markup Language) – Стандартизированный язык гипертекстовой разметки документов.

JSON (JavaScript Object Notation) – Текстовый формат обмена данными, основанный на JavaScript.

MVC (Model-View-Controller) – Схема разделения данных приложения и управляющей логики на три отдельных компонента: модель, представление и контроллер.

ODM (Object-Document Mapper) – Технология программирования, связывающая документно-ориентированные базы данных с классами объектно-ориентированных языков.

REST (Representational State Transfer) – Архитектурный стиль взаимодействия компонентов приложения в сети.

WEB – Интернет-пространство.

БД – База данных.

ВКР – Выпускная квалификационная работа.

ВУЗ – Высшее учебное заведение.

ЕГЭ – Единый Государственный экзамен. Форма проведения государственной итоговой аттестации по образовательным программам среднего общего образования в Российской Федерации.

ПО – Программное обеспечение.

РФ – Российская Федерация.

СУБД – Система управления базами данных.

ТОГИРРО – Тюменский областной государственный институт развития регионального образования.

ВВЕДЕНИЕ

В последние годы все больше внимания уделяется анализу результатов ЕГЭ, как важнейшего инструмента оценки знаний и компетенций выпускников средних учебных заведений. Результаты экзамена используются не только для поступления в вузы, но и для оценки эффективности образовательных программ и уровня подготовки учеников средних школ в целом. Анализ данных экзаменационных результатов позволяет выявить индивидуальные и групповые особенности овладения учебной программой, а также определить сильные и слабые стороны образовательной системы в целом. Такой анализ позволяет ТОГИРРО обратить внимание на системные проблемы, такие как низкая подготовка участников ЕГЭ, и предоставляет возможность разработки курсов повышения квалификации для преподавательского состава.

Для того чтобы оценить эффективность образовательных программ и учебных заведений среднего образования Тюменской области, возникла потребность в создании аналитической системы. Целью выпускной квалификационной работы является создание такой системы. Данная система должна включать в себя обработку, хранение, агрегацию и визуализацию данных, а также анализ результатов экзаменов. Также одним из основных требований для системы является возможность кластеризации учебных заведений по качеству выполнения отдельных заданий различных учебных предметов государственной итоговой аттестации.

Для разработки аналитической системы требуется реализовать обработку, хранение, анализ и визуализацию результатов экзаменов, и данные школ.

В ходе анализа этапов разработки аналитической системы в рамках выполнения ВКР были поставлены следующие задачи:

- Изучить предоставленные данные школ и экзаменационных работ.
- Обработать, провалидировать и преобразовать исходные данные.

- Изучить, выбрать и обосновать выбор способов визуализации данных, технологий и инструментов, необходимых для разработки аналитической системы.
- Сформировать базу данных и средства загрузки данных в систему в рамках веб-приложения.
- Реализовать пользовательский интерфейс веб-приложения, включающий в себя набор графиков и диаграмм для визуализации данных итоговой аттестации.
- Реализовать в рамках веб-приложения средства кластеризации учебных заведений по степени сходства качества выполнения тестовых заданий итоговой аттестации учениками данных учебных заведений.

ГЛАВА 1. МАТЕМАТИЧЕСКИЕ МЕТОДЫ ДЛЯ АНАЛИЗА ТЕСТОВЫХ ДАННЫХ

Для исследования различных способностей какого-либо теста используется ряд характеристик:

1. Трудность тестовых заданий.
2. Дискриминативность теста.
3. Коэффициент корреляции результатов выполнения отдельных заданий.
4. Описательные статистики: количество элементов в выборке, медиана, среднее арифметическое, среднеквадратическое отклонение, коэффициент асимметрии, коэффициент эксцесса.

Также производится визуализация значений числовых признаков.

Требуется найти учебные заведения со схожими проблемами в выполнении тестовых заданий. Для этого нужно выполнить кластеризацию учебных заведений по качеству выполнения отдельных тестовых экзаменационных заданий отдельно по каждому экзамену.

1.1. ПОКАЗАТЕЛИ КАЧЕСТВА ТЕСТОВЫХ ЗАДАНИЙ

Одним из критериев оценивания качества теста является оценка трудности тестовых заданий [22]. Трудность тестовых заданий можно вычислить только экспериментально после выполнения теста. Трудность тестового задания является разностью 1 и доли участников, верно выполнивших задание и определяется по формуле:

$$\mathrm{U}i = 1 - \frac{\mathrm{N}}{\mathrm{Na}}, \qquad (1)$$

где Ui – трудность i-го задания;

N – количество участников экзамена, справившихся с заданием;

Na – общее количество участников экзамена.

Коэффициент трудности тестового задания выше у тех заданий теста, с которыми справились меньше участников тестирования, т. е. чем выше коэффициент трудности, тем задание теста сложнее. Если коэффициент

трудности задания низкий, то с данным заданием теста справляется большинство участников и данное задание не является информативным. Трудность вычисляется для определения того, насколько тест в целом является информативным, т. е. насколько тест показывает уровень освоения образовательных тем его участниками.

Для более углубленного понимания устройства и качества экзаменационного теста требуется вычислить его дискриминативность [23]. Дискриминативность теста – дифференцирующая, различающая способность теста в целом или отдельного тестового задания, указывающая на их способность разделять отдельных испытуемых по уровню выполнения [24]. Если все испытуемые дают на тестовое задание один и тот же правильный ответ, то это означает, что данное задание обладает низкой дискриминативностью. Дискриминативность задания определяется обычно как разность между относительной численностью испытуемых, справившихся с заданием, из высокопродуктивной и низкопродуктивной групп.

Дискриминативность тестовых заданий вычисляется по следующему алгоритму: результаты тестов делятся на 4 квартиля по успешности выполнения тестовых заданий: 1-й квартиль включает учащихся, получивших низкое количество баллов, 4-й квартиль включает учащихся, получивших высокое количество баллов; далее, эти 2 группы сравниваются между собой и по каждому заданию теста проверяется какая из групп выполняет задание лучше – группа участников экзамена из 1 квартиля или из 4-го; для каждого тестового задания результатом является разность между долей участников, правильно решивших задание, из 1-го квартиля и из 4-го. В формальном виде вычисление индекса дискриминативности для тестового задания с номером i производится по формуле:

$$\mathrm{Di} = \frac{Nnmax}{Nmax} - \frac{Nnmin}{Nmin} \qquad (2)$$

где D_i – дискриминативность задания i;

$Nnmax$ – количество участников в группе лучших, верно выполнивших задание;

$Nnmin$ – количество участников в группе худших, верно выполнивших задание;

$Nmax = Nmin$ – общее количество участников в крайних группах.

1.2. КОРРЕЛЯЦИЯ РЕЗУЛЬТАТОВ ВЫПОЛНЕНИЯ ЗАДАНИЙ

Чтобы понять, есть ли похожие задания, в качестве выполнения которых могут быть четкие зависимости, необходимо вычислить попарную корреляционную зависимость между качеством выполнения заданий: отдельно для каждого задания. Если коэффициенты корреляции будут низкими, то можно принять нулевую гипотезу для конкретных заданий ЕГЭ. Нулевая гипотеза — принимаемое по умолчанию предположение о том, что не существует связи между двумя признаками. Нулевая гипотеза считается верной, пока нельзя доказать обратное.

Корреляция – статистическая взаимосвязь двух или более случайных величин (либо величин, которые можно с некоторой допустимой степенью точности считать таковыми). При этом изменения значений одной или нескольких из этих величин сопутствуют систематическому изменению значений другой или других величин. Корреляция показывает силу связи между двумя переменными и выражается числовым коэффициентом. Коэффициент корреляции – это статистическая мера, которая вычисляет силу связи между относительными движениями двух переменных. Значения коэффициента корреляции находятся в диапазоне от -1.0 до 1.0. Корреляционная матрица (матрица корреляций) – это квадратная таблица, строками и столбцами которой являются признаки, а на пересечении строк и столбцов коэффициенты корреляции для соответствующей пары признаков. Матрицы корреляции используются для демонстрации выявленных закономерностей в результате обработки данных.

Существует несколько методов вычисления коэффициентов корреляции, выбор метода зависит от вида шкалы, к которой относятся переменные.

Коэффициент ранговой корреляции Спирмена – непараметрический метод, который используется с целью статистического изучения связи между явлениями. В этом случае определяется фактическая степень параллелизма между двумя количественными рядами изучаемых признаков и дается оценка тесноты установленной связи с помощью количественно выраженного коэффициента, используется для выявления и оценки тесноты связи между двумя рядами сопоставляемых количественных показателей.

Вычисление коэффициента корреляции рангов производится по формуле:

$$r = 1 - \frac{6*\sum d^2}{n(n^2-1)} \quad (4)$$

где r – коэффициент корреляции Спирмена;

d – разности рангов X_i и Y_i;

n – количество элементов в выборке.

При использовании коэффициента ранговой корреляции условно оценивают тесноту связи между признаками, считая значения коэффициента меньше 0.3 – признаком слабой тесноты связи; значения от 0.3 до 0.7 – признаком умеренной тесноты связи; значения 0.7 и более – признаком высокой тесноты связи.

Коэффициент корреляции Кенделла — мера линейной связи между случайными величинами. Корреляция Кенделла является ранговой, то есть для оценки силы связи используются не численные значения, а соответствующие им ранги. Коэффициент инвариантен по отношению к любому монотонному преобразованию шкалы измерения. Коэффициент корреляции Кенделла используется в случае, когда переменные представлены двумя порядковыми шкалами. Этот коэффициент изменяется в пределах от -1.0 до 1.0 и рассчитывается по формуле:

$$\tau = \frac{2(\sum P - \sum Q)}{n(n-1)} \quad (5)$$

где τ — коэффициент корреляции Кендалла;

P — суммарное число наблюдений, следующих за текущими наблюдениями с большим значением рангов Y;

Q — суммарное число наблюдений, следующих за текущими наблюдениями с меньшим значением рангов Y;

n – количество элементов в выборке.

Т. е. корреляция Кенделла может использоваться для любых типов данных, даже если они не являются нормально распределенными или имеют выбросы. Она измеряет только порядковую согласованность между переменными, но не учитывает расстояния между значениями.

Коэффициент корреляции Пирсона применяется для исследования взаимосвязи двух переменных, измеренных в метрических шкалах на одной и той же выборке. Он позволяет определить, насколько пропорциональна изменчивость двух переменных. Он характеризует существование линейной связи между двумя величинами. Формула расчета коэффициента корреляции Пирсона:

$$r_{xy} = \frac{\sum(x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 * \sum(y_i - \bar{y})^2}} \quad (6)$$

где $\bar{x}, \bar{y}$ – выборочные средние.

Так как распределение результатов выполнения экзаменационных заданий не относится к нормальному, то для вычисления матрицы корреляции будет использован коэффициент Кенделла.

1.3. ОПИСАТЕЛЬНЫЕ СТАТИСТИКИ

Медиана в теории вероятностей – одна из характеристик распределения вероятностей случайной величины, иногда её называют серединным значением случайной величины. Медианой случайной величины X называется число m такое, что вероятности P {X ⩾ m} ⩾1/2 и P {X ⩽ m} ⩾ 1/2. Медиана существует всегда.

Среднее арифметическое – число, равное сумме всех чисел множества, делённой на их количество.

$$\overline{\mathrm{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{x_i} \qquad (7)$$

где $\overline{\mathrm{x}}$ – среднее арифметическое;

n – количество чисел множества;

x_i – i-е число множества.

Среднеквадратичное отклонение – квадратный корень из дисперсии случайной величины. Дисперсия – среднее арифметическое квадратов отклонений каждого значения от среднего значения. В статистике и вероятностном анализе дисперсия используется для измерения того, насколько далеко значения набора данных распределены вокруг их среднего значения. Чем больше значение дисперсии, тем больше разброс данных, а меньшая дисперсия указывает на меньший разброс.

$$S = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2} \qquad (8)$$

где X_i — i-й элемент выборки;

S — среднеквадратичное отклонение;

$\overline{\mathrm{X}}$ — среднее арифметическое выборки (выборочное среднее);

N — количество элементов в выборке.

Коэффициент асимметрии измеряет отклонение распределения данных от симметричного распределения. Симметрическое распределение – это распределение вероятностей, в котором значения случайной переменной симметрично расположены относительно своего среднего значения. Симметричные распределения характеризуются тем, что их среднее арифметическое, мода и медиана равны. Если коэффициент асимметрии равен 0, это указывает на симметричное распределение. Если коэффициент асимметрии положительный, то это указывает на то, что правый хвост распределения более тяжелый (большие значения находятся правее среднего значения). Если коэффициент асимметрии отрицательный, то левый хвост

распределения более тяжелый (маленькие значения находятся левее среднего значения). Формула расчета коэффициента асимметрии:

$$A = \frac{\sum_{i=1}^{N} (X_i - \bar{X})^3}{S_x^3 \cdot N} \quad (9)$$

где A — коэффициент асимметрии;

X_i — i-й элемент выборки;

$\bar{X}$ — среднее арифметическое выборки (выборочное среднее);

S_x — среднеквадратичное отклонение;

N — количество элементов в выборке.

Коэффициент эксцесса — числовая характеристика степени остроты пика распределения случайной величины. Формула расчёта коэффициента островершинности (коэффициента эксцесса):

$$\gamma = \frac{\sum_{i=1}^{N} (X_i - \bar{X})^4}{S_x^4 \cdot N} \quad (10)$$

где X_i — i-й элемент выборки;

γ — коэффициента эксцесса;

$\bar{X}$ — среднее арифметическое выборки (выборочное среднее);

S_x — среднеквадратичное отклонение;

N — количество элементов в выборке.

1.4. АЛГОРИТМЫ КЛАСТЕРИЗАЦИИ

Кластеризация представляет собой процесс разделения множества объектов на группы (кластеры) в соответствии с определенным критерием. Каждый кластер объединяет самые схожие друг с другом объекты.

Пусть X — множество объектов, Y — множество идентификаторов кластеров. На множестве X задана функция расстояния между объектами $\rho(x, x')$. Дана конечная обучающая выборка объектов $X^m = \{x_1, \ldots, x_m\} \subset X$. Необходимо разбить выборку на кластеры, то есть каждому объекту $x_i \in X^m$ сопоставить метку $y_i \in Y$, таким образом чтобы объекты внутри каждого

кластера были близки относительно метрики ρ, а объекты из разных кластеров значительно различались.

Алгоритмом кластеризации является такая функция a: X → Y, которая любому объекту x ∈ X ставит в соответствие идентификатор кластера y ∈ Y.

Существует ряд алгоритмов (методов) кластеризации:

- Графовые, в последнее время почти не применяющиеся на практике.
- Вероятностные, в которых каждый объект выборки относится к какому-либо из кластеров с определенной степенью вероятности.
- Иерархические, в которых создается иерархия вложенных классов.
- Алгоритм k-средних (англ. K-means), основанный на минимизации суммарного квадратичного отклонения точек кластеров от их центров.
- Распространения похожести, в котором типичные объекты каждого кластера выбираются путём распространения сообщений о похожести между каждой парой объектов.
- DBSCAN – алгоритм группировки точек в области с высокой плотностью в один кластер.

Кластерный анализ (кластеризация) применяются для решения широкого ряда задач во многих сферах человеческой жизни. Алгоритмы кластеризации используются в биологии, маркетинге, компьютерных науках, в социологии, психологии, медицине и т. д.

1.5. ОПИСАНИЕ ИСХОДНЫХ ДАННЫХ

Исходные данные представлены в виде Excel – файлов, содержащих 2 листа с таблицами. Первый лист содержит результаты экзаменов по всем общеобразовательным предметам (Таблица 1) за определенный год. Год указан в названии Excel-листа. Второй лист содержит информацию о школах, в которых сдавали экзамены в определенный год (Таблица 2).

Таблица 1

Описание основных данных в Excel-файле

Название столбца	Тип данных	Шкала	Пример данных
Код школы	int	Номинальная	201065
Класс	string	Номинальная	11А
Название предмета	string	Номинальная	И
Первичный балл	int	Отношений	19
Процент выполнения	int	Отношений	59
100 бальная шкала	int	Отношений	80
Первичный балл за часть с кратким ответом	int	Отношений	11
Оценка кратких ответов	string	Номинальная	++++++++++-
Первичный балл за часть с развернутым ответом	int	Отношений	8
Оценка развернутых ответов	string	Номинальная	2(2)0(2)2(2)1(3)2(3)0(4)1(4)
Первичный балл за устную часть	int	Отношений	19
Оценка устных ответов	string	Номинальная	1(1)5(5)3(3)2(2)2(2)3(3)2(2)2(2)

Таблица 2

Описание данных учебных заведений в Excel-файле

Название столбца	Тип данных	Шкала	Пример данных
ID Школы	Guid	Номинальная	A447EA29-0BE1-4CE7-B8C4-000B61B0ADB3
Код школы	int	Номинальная	243010

Продолжение таблицы 2

Название столбца	Тип данных	Шкала	Пример данных
Адрес школы	string	Номинальная	626157, Тюменская обл., г. Тобольск, мкр. 7, д. 54
Название школы	string	Номинальная	МАОУ «Гимназия имени Н.Д.Лицмана» г. Тобольска
Вид школы	string	Номинальная	Гимназия
Тип школы	string	Номинальная	Общеобразовательное учреждение/организация
Краткое название района (города)	string	Номинальная	Тобольск
Вид школы по юридической принадлежности	string	Номинальная	Муниципальное автономное образовательное учреждение/организация
Тип населенного пункта	string	Номинальная	Населенный пункт городского типа
Полное название района (город)	string	Номинальная	г. Тобольск
Контролирующий школу орган	string	Номинальная	Комитет по образованию администрации г. Тобольска

СУБД для хранения данных в первую очередь должна работать быстро при запросах на получение данных. Реляционная целостность данных и транзакционность не являются необходим условием при выборе СУБД. Исходя из потребностей была выбрана СУБД MongoDB. Были разработаны конвейеры обработки данных. Затем, данные были пакетно обработаны и загружены в СУБД. Данные в MongoDB хранятся в виде коллекций документов, т. к. это документно-ориентированная СУБД. Коллекции документов для разработчика представляют собой наборы JSON объектов. Для анализа и визуализации использовались данные из коллекции, пример из которой указан в Приложении 1.

Данные школ не содержат географических координат для дальнейшего анализа, поэтому, требуется выбрать сервис для прямого геокодирования школ, т. е. получения координат школ по их исходным адресам и реализовать геокодирование.

В рамках выполнения магистерской диссертации требуется разработать аналитическую систему, позволяющую загружать данные ЕГЭ, которые представлены в виде Excel-файлов; обрабатывать загруженные данные; сохранять данные в БД для последующих запросов и выборок; строить WEB-формы с графиками для наглядного визуального представления различных тестовых данных. Приложение должно включать возможность загрузки данных в хранилище, а, также, в нем должны отображаться графики, демонстрирующие различные распределения баллов экзаменов.

Некоторые учебные заведения в городах относились не к городским образовательным департаментам, а к районным. Административное отношение данных заведений было исправлено.

ЕГЭ в РФ предусматривает на выбор 5 иностранных языков для прохождения экзамена: английский язык, испанский язык, французский язык, немецкий язык, китайский язык. За период 2018–2022 испанский, немецкий, французский и китайский языки в качестве иностранного языка для прохождения итоговой аттестации были выбраны менее чем 1% учащихся. Количество баллов и структура заданий данного экзамена одинаковы для всех языков. Поэтому было решено объединить результаты экзаменов по всем иностранным языкам в один.

ЕГЭ по предметам русский язык и математика (на выбор базовая / профильная / оба варианта) являются основными предметами для получения аттестата. Успешное прохождение экзамена по остальным предметам не является обязательным условием для получения аттестата об окончании среднего учебного заведения. Поэтому количество участников экзамена по русскому языку и математике значительно больше количества участников

экзамена по остальным предметам. Экзамен по базовой математике не проводился в 2020 и 2021 годах.

ГЛАВА 2. ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА И ТЕХНОЛОГИИ РАЗРАБОТКИ

2.1. СПОСОБЫ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ ДАННЫХ

Визуализация данных – это процесс представления информации в графическом или визуальном формате с целью лучшего понимания данных, выявления тенденций, связей и паттернов. Она применяется в различных областях деятельности: научные исследования, бизнес-аналитика, психология, информационные дашборды, медицина, информационные технологии, финансы, социальные науки и т. д. Визуализация позволяет представить сложные и большие объемы данных в виде наглядных и понятных диаграмм, графиков, сводных таблиц и т. д. [9]

Визуализация помогает упростить сложные данные и сделать их более доступными для анализа. Графики и диаграммы легче воспринимаются и интерпретируются людьми, чем большие таблицы, даже если люди не имеют специальных знаний в области математической статистики или бизнес-аналитики. Визуализация данных позволяет легче объяснить и представить результаты исследований или аналитические выводы широкой аудитории. Графическое представление данных делает информацию более простой, понятной и запоминающейся.

Визуализация данных помогает выявить аномалии и выбросы. Графическое представление данных может помочь идентифицировать необычные значения или выбросы, которые могут быть скрыты в числовых таблицах или базах данных. Например, построение графика распределения данных позволяет быстро обнаружить наличие экстремальных значений.

Когда данные представлены в графическом формате, становится проще выявить скрытые взаимосвязи паттерны и тренды в данных, которые могут быть незаметны при простом числовом анализе. Например, диаграмма рассеяния может помочь обнаружить корреляцию между двумя переменными, которую сложно заметить в числовых таблицах.

Визуализация данных облегчает мониторинг и отслеживание. Представление данных в графическом виде позволяет легко отслеживать изменения и тренды со временем. Дашборды и графики могут обеспечить наглядное представление ключевых метрик и показателей производительности, позволяя быстро определять изменения и принимать соответствующие меры.

Визуализация данных способствует принятию обоснованных решений на основе интерпретации графической информации. Графическое представление данных позволяет точнее и полнее оценивать ситуацию и принимать осознанные решения. Визуализация позволяет сравнивать различные варианты, оценивать эффективность и результаты различных стратегий или альтернатив [19].

Графики являются одним из наиболее распространенных и эффективных инструментов визуализации данных. Они представляют собой графическое отображение данных в виде линий, столбцов, точек, кругов и других визуальных элементов. Графики позволяют наглядно представить отношения, распределения, тренды и паттерны в данных.

Самые распространенные типы графиков, используемых в визуализации данных:

- Линейный график используется для отображения изменения значений переменной. Линейный график отображает данные с помощью линий, соединяющих точки, представляющие значения переменных.
- Столбчатая диаграмма используется для сравнения значений различных категорий. Она состоит из вертикальных столбцов, где высота каждого столбца пропорциональна значению, которое он представляет. Столбчатые диаграммы могут быть горизонтальными или вертикальными.
- Гистограмма используется для показа распределения данных в числовом формате. Гистограмма разбивает данные на несколько интервалов

или столбцов и отображает количество значений, попадающих в каждый интервал.

- Визуализация географической информации на карте выполняется с помощью географической диаграммы. Географическая диаграмма – это способ представления данных на географической карте с помощью различных элементов, таких как маркеры, линии, полигоны и т. д. Она позволяет отображать пространственную информацию и связи между данными. Маркеры на карте используются для обозначения точечных объектов на карте. Маркеры могут иметь различные формы, цвета и размеры, чтобы передать дополнительную информацию о данных.
- Диаграмма размаха – это графическое представление распределения данных, которое отображает пять основных статистических характеристик: минимум, первый (нижний) квартиль, медиану, третий (верхний) квартиль и максимум. Также, диаграмма размаха отображает выбросы и "усы", которые представляют интервал, в котором ожидается, что будет находиться большинство данных.
- Визуализация матричных данных часто выполняется с помощью тепловой карты. Тепловая карта — это представление матричных данных в графической форме, которая позволяет наглядно исследовать связи и структуры внутри матрицы. Тепловая карта используется для отображения значений в матрице с помощью цветовой шкалы. Каждая ячейка матрицы получает цвет, который соответствует значению в ячейке. Тепловая карта позволяет визуализировать относительные значения внутри матрицы и выявлять паттерны или аномалии.

2.2. ТЕХНОЛОГИИ РАЗРАБОТКИ

Выбор технологий, используемых при создании хранилища, загрузке данных в БД, обработке, анализу и визуализации напрямую связан с решаемыми задачами и со скоростью разработки системы. Основными критериями при выборе технологий являлись: возможность бесплатного

использования, легкость освоения, богатая документация, популярность в сообществе разработчиков, присутствие поддержки и выпуск актуальных версий в 2022–2023 годах.

В ходе разработки аналитической системы в рамках ВКР решаются задачи извлечения, обработки, хранения, объединения, визуализации данных. Для визуализации и отображения данных конечному пользователю производятся группировки и агрегации данных в различных разрезах.

В качестве основного языка разработки был выбран язык программирования Python. Это высокоуровневый интерпретируемый язык программирования, который на апрель 2023 года является самым популярным языком среди всех языков программирования в мире [16]. Python широко используется как универсальный инструмент разработки для множества задач. Он применяется для научных вычислений, создания веб-приложений, визуализации данных, реализации алгоритмов глубокого обучения, написания скриптов и множества иных различных задач. Python имеет низкий порог вхождения и удобный синтаксис. В настоящее время Python является основным языком для анализа данных, их визуализации и машинного обучения. Большое количество библиотек для анализа и визуализации данных, классических алгоритмов машинного обучения и алгоритмов глубокого обучения написаны на Python.

Библиотека NumPy предназначена для работы с многомерными массивами [11]. Данная библиотека не используется напрямую в работе, но используется всеми остальными библиотеками, работающими с данными.

SciPy содержит математические инструменты для научных и прикладных вычислений, а, также, богатую документацию. SciPy содержит методы для проверки статистических гипотез [15] и выполнения корреляционного анализа, необходимые в данной работе.

Для обработки данных применяется библиотека Pandas, которая предлагает удобные методы для эффективной работы с наборами данных [12]. С помощью этой библиотеки можно осуществлять операции по сбору,

очистке, слиянию, нормализации, добавлению и удалению столбцов из двумерных индексированных массивов (DataFrame) [18] и широкий ряд иных операций. Она также предоставляет возможности для анализа и визуализации данных [13]. Для обеспечения высокой производительности Pandas реализована на языке C и использует диалект Cython.

Предварительный анализ и ряд преобразований и агрегаций данных производилась в облачном сервисе Google Colab, основанном на проекте Jupiter. Google Colab позволяет удаленно работать над проектами, запуская код в окне браузера. Google Colab является бесплатным инструментом от компании Google и не требует локальной установки какого-либо ПО.

В качестве базы данных была выбрана MongoDB, которая является документно-ориентированной системой управления базами данных. Одним из преимуществ MongoDB является отсутствие необходимости предварительного описания схемы таблиц, что позволяет быстро загружать документы. MongoDB является одним из наиболее популярных решений NoSQL, позволяющих хранить различные типы данных. Ее основная цель заключается в упрощении хранения данных и повышении производительности взаимодействия с ними. Данная СУБД поддерживает индексацию. Одним из главных преимуществ MongoDB является использование динамической схемы хранения данных. Данные в MongoDB представлены в формате JSON, который обеспечивает удобочитаемость при передаче данных между веб-приложениями и серверами. В MongoDB JSON реализован в формате BSON (бинарный JSON), который обеспечивает надежность и эффективность, особенно в терминах скорости и использования памяти. Важно отметить, что MongoDB не поддерживает сложные многодокументные транзакции. Однако для разработки приложений часто не требуется поддержка транзакций. MongoDB предлагает множество полезных возможностей для работы с приложениями, написанными на языке Python.

Для взаимодействия с базой данных MongoDB существуют несколько библиотек. Одними из самых популярных являются: PyMongo и MongoEngine.

PyMongo является официальным драйвером MongoDB для языка Python и предоставляет полный функционал API запросов для работы с MongoDB [8]. С помощью этой библиотеки можно подключаться к базе данных MongoDB и выполнять запросы данных с использованием MongoDB Query API. При получении данных с помощью PyMongo, результатом будут словари, к которым можно обращаться по ключам.

MongoEngine — это ODM библиотека, которая позволяет подключаться к базе данных MongoDB и использовать документы, как если бы они были Python объектами. Она предоставляет более высокоуровневый интерфейс и абстракции для работы с базой данных, что делает код более удобочитаемым и позволяет использовать модели объектов для работы с данными в MongoDB [10]. MongoEngine построен поверх PyMongo и использует PyMongo внутри себя для управления соединениями с базой данных. MongoEngine позволяет определять схему с помощью Python классов для данных. Затем он сопоставляет документы с этими классами и позволяет манипулировать ими. Для валидации и загрузки данных в БД была выбрана библиотека MongoEngine, т. к. в отличие от PyMongo, MongoEngine инкапсулирует подключение и отключение от базы данных, создание и очистку коллекций, не требует написания дополнительного маппинга документов из Mongo коллекций в Python классы, а также, позволяет валидировать данные с помощью удобных синтаксических конструкций.

Для анализа и визуализации данных традиционно используются индексированные двумерные массивы библиотеки pandas, имеющие тип DataFrame. Для получения данных из БД и их последующей записи в объекты DataFrame были рассмотрены несколько непопулярных библиотек, позволяющих значительно ускорять получение данных. Большая часть таких библиотек не поддерживаются на протяжении нескольких лет (по состоянию на 2023 год). Одной из таких библиотек, которая имеет поддержу, является PyMongoArrow [14]. Данная библиотека является расширением PyMongo. Она

содержит набор инструментов для быстрой загрузки документов из MongoDB в таблицы DataFrame библиотеки pandas или в массивы NumPy.

Для веб-разработки на языке Python разработчики обычно выбирают один из следующих фреймворков: Django, Flask, FastAPI, Pyramid, web2py и CherryPy. Фреймворки web2py и CherryPy имеют меньшую популярность и отсутствие активного сообщества разработчиков. Flask и FastAPI являются легковесными фреймворками, которые хорошо подходят для быстрого создания REST API. Однако, они не имеют встроенной поддержки веб-форм, что делает их менее подходящими для создания аналитических систем и дашбордов. Наиболее распространенным и популярным фреймворком является Django. Django представляет собой полноценный веб-фреймворк, основанный на шаблоне проектирования MVC, и широко используется для разработки веб-приложений на Python. Он обладает богатой документацией и активным сообществом разработчиков. Django позволяет быстро разрабатывать веб-приложения и обладает высокой гибкостью, позволяющей масштабировать прототипы до крупных проектов и систем. Фреймворки web2py и Pyramid имеют немного более низкую производительность по сравнению с Django. Исходя из всех вышеперечисленных достоинств и недостатков, в качестве веб-фреймворка для разработки системы был выбран Django.

Bootstrap — это фронтенд-фреймворк, который обеспечивает простую и быструю разработку веб-приложений. Он содержит предварительно созданные шаблоны дизайна на основе HTML и CSS для различных элементов, таких как формы, кнопки, таблицы, навигация, карусели изображений и многое другое. Использование Bootstrap имеет множество преимуществ. Например, он автоматически адаптирует размер страницы к устройству, на котором отображается приложение. Кроме того, Bootstrap обеспечивает удобное расположение элементов на сайте с помощью встроенной сетки [21].

Для визуализации данных в веб-приложении были рассмотрены библиотеки Seaborn, Matplotlib и Plotly. Для предварительного анализа и

визуализации данных при решении различных задач, связанных с анализом данных и машинным обучением в связке с библиотекой pandas по-умолчанию используется библиотека matplotlib. В более продвинутых случаях используется библиотека Seaborn, предоставляющая более богатые возможности для визуализации данных. Обе вышеописанные библиотеки не предоставляют интерактивность при взаимодействии с выстраиваемыми графиками. Plotly – интерактивная библиотека для языка Python с открытым исходным кодом, предназначенная для визуализации данных. Данная библиотека поддерживает более 40 уникальных типов графиков и диаграмм, охватывающих широкий спектр статистических, финансовых, географических, научных и трехмерных сценариев использования. Главной особенностью Plotly является возможность создавать интерактивные веб-визуализации, которые можно использовать как часть веб-приложений, созданных исключительно на Python. Также, Plotly позволяет сохранять построенные графики в отдельные файлы. Для визуализации данных была выбрана библиотека Plotly.

Dash – фреймворк с открытым исходным кодом, используемый для создания аналитических веб-приложений на Python [5]. Dash хорошо интегрирован с Django средствами библиотеки Django-plotly-dash [6] и имеет хорошую документацию. Он упрощает разработку приложений для обработки данных, позволяет создавать информационные панели в браузере. Из фреймворка Dash были взяты шаблоны форм для разработки дашборда. Приложения Dash написаны на языке Python, поэтому необходимость в дополнительном HTML или JavaScript практически не возникает.

Для управления версиями разработки был выбран Git, который является наиболее популярным инструментом и фактическим стандартом в индустрии разработки программного обеспечения в 2023 году [2].

PyCharm – интегрированная среда разработки для языка Python. Предоставляет средства для анализа кода, отладчик и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе

IntelliJ IDEA. PyCharm является платным коммерческим продуктом, но, компания JetBrains предоставляет бесплатные лицензии для разработки в образовательных и академических целях.

Для просмотра документов, содержащихся в коллекциях баз данных MongoDB в рамках создания аналитической системы было использовано ПО MongoDBCompass. MongoDBCompass позволяет подключаться к БД MongoDB и предоставляет удобный, простой и интуитивно понятный интерфейс. MongoDBCompass позволяет подключаться к серверам и базам данных MongoDB, просматривать документы коллекций, создавать индексы, фильтровать запросы, добавлять, удалять и изменять документы коллекций [3].

Для ускорения разработки было принято решение контейнеризовать разрабатываемое приложение. Контейнеры представляют собой стандартизированный способ упаковки программного обеспечения, в котором приложение и все его зависимости объединяются в одну единицу, обеспечивая простой перенос и запуск на различных платформах без необходимости установки или настройки дополнительных компонентов. Для реализации контейнеризации был выбран Docker [17]. Docker предоставляет разработчикам возможность создавать приложения в изолированных контейнерах, что обеспечивает более эффективное функционирование, повышенную портативность и упрощенное управление инфраструктурой. Docker-контейнера на 2023 год, фактически, являются стандартным средством контейнеризации приложений.

Для прямого геокодирования координат школ по их заданным адресам были рассмотрены следующие сервисы геокодирования:

1. 2GIS Geocoder [1].
2. Сервис Geocoding Google [7].
3. geotree.ru/geocoder Геокодирование адреса, координаты по адресу, API maps, карты [20].
4. Dadata.ru Геокодирование (координаты по адресу) [4].

5. Яндекс Геокодер [25].

Сравнение различных сервисов геокодирования приведено в Таблице 3.

Таблица 3

Сравнение сервисов геокодирования

Сервис	Точность	Наличие богатой документации	Возможность бесплатного использования	Скорость начала использования
2GIS Geocoder	+	+	+	-
Geocoding Google	-	+	-	-
GeoTree	-	-	-	-
DaData	+	+	-	-
Яндекс Геокодер	**+**	**+**	**+**	**+**

Из всех вышеперечисленных сервисов самым точным для прямого геокодирования школ в Тюменской области РФ был Яндекс Геокодер. Точность геокодирования остальных сервисов была ниже, т. е. часть адресов не была распознана. Также Яндекс Геокодер имеет удобное API, быструю обратную связь, возможность бесплатного использования и богатую документацию [26]. Предоставление ключа доступа к API от сервиса Яндекс Геокодер было произведено в течение 10 часов после запроса. В качестве сервиса геокодирования был выбран Яндекс Геокодер.

Для разработки автоматизированной системы анализа данных использованы следующие библиотеки, средства и технологии:

- Python – основной язык программирования для разработки системы анализа и визуализации данных.
- NumPy – библиотека для языка Python, содержащая функции для работы с многомерными массивами.

- SciPy – библиотека для языка Python, предназначенная для выполнения научных расчётов.
- pandas – библиотека для языка Python, содержащая инструменты для обработки и подготовки данных.
- Google Colab – бесплатный облачный веб-сервис, предоставляющий инструменты для работы с кодом на языке Python в браузере.
- MongoDB – документно-ориентированная СУБД.
- PyMongo – официальная Python-библиотека для работы с СУБД MongoDB.
- MongoEngine – ODM библиотека, которая позволяет подключаться к базе данных MongoDB и использовать документы, как если бы они были Python объектами.
- MongoDB Compass – бесплатное ПО для манипуляций с базами данных MongoDB.
- PyMongoArrow – настроенная над PyMongo библиотека, позволяющая быстро загружать документы из коллекций MongoDB в объекты DataFrame библиотеки pandas.
- Django – свободный веб-фреймворк для разработки MVC веб-приложений на языке Python.
- Bootstrap – веб-фреймворк для создания шаблонов, веб-форм и веб-приложений.
- Plotly – библиотека для создания интерактивных графиков.
- Dash – фреймворк для построения визуальных приложений на языке Python.
- Django-plotly-dash – библиотека для использования Dash-приложений внутри форм фреймворка Django.
- Git – система управления версиями разработки ПО.
- Docker – самое распространенное бесплатное средство для контейнеризации разработанного ПО.

- Яндекс Геокодер – сервис геокодирования.
- Visual Studio Code – редактор кода, включающий богатый набор плагинов.
- PyCharm – интегрированная среда для разработки на языке Python.

Выбранные технологии позволяют решить поставленные в Главе 1 задачи.

ГЛАВА 3. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

В результате выполнения ВКР была разработана аналитическая система для анализа и визуализации результатов Единого Государственного Экзамена по различным учебным предметам за различные годы. Разработанная система представляет собой WEB – приложение, написанное на языке программирования Python версии 3.9 с использованием фреймворка Django версии 3.2 и технологий, перечисленных в главе 2 данной работе.

В хранилище загружаются “сырые” данные, содержащие результаты экзаменов, т. е. данные в том виде, в котором они хранятся в исходных файлах (Таблица 1). Также, загружаются данные учебных заведений (Таблица 2). При загрузке данных производится прямое геокодирование для получения координат учебных заведений. При запросах пользователя к системе производится построение графических элементов на визуальных WEB-формах; кластеризация учебных заведений по степени похожести выполнения отдельных заданий итоговой аттестации; подсчет различных числовых статистических метрик, характеризующих тестовые данные ЕГЭ: медиан, средних, количества участников, коэффициентов асимметрии и эксцесса.

3.1. ОБРАБОТКА ДАННЫХ

Исходные данные хранятся в Excel–документах. Данные загружаются в систему, валидируются с помощью правил, предоставляемых сущностями, унаследованными от типов Document и BaseField из библиотеки MongoEngine [10], подключаемой как Python-модуль. Результаты экзаменов представлены в строковом виде. Для их анализа необходимо представить данные в числовом виде. Например, развернутые результаты простых заданий представлены строкой вида: -+-2+-++7++-, где “+” – правильно выполненное задание, “-” – неправильно выполненное, целое число – количество баллов за задание. После преобразования данные принимают вид: 010210117110, где, соответственно, “1” – правильно выполненное задание, “0” – неправильно выполненное, количество баллов за задание не преобразуется.

Результаты заданий, требующих развернутый ответ представлены строкой вида: 2(2)2(2)2(2)1(3)3(3)0(4)1(4), где число без скобок – балл за выполнение задания, число в скобках – максимально возможный балл за задание.

Результаты заданий, требующих устный ответ представлены строкой вида: 2(2)2(2)2(2)1(3)3(3)0(4)1(4), где число без скобок – балл за выполнение задания, число в скобках – максимально возможный балл за задание.

Результаты заданий экзаменов, не относящихся к дихотомической шкале измерения и требующих развернутый ответ, записываются в отдельные поля Python-объектов, унаследованных от типа DynamicDocument библиотеки MongoEngine. Преимущество СУБД MongoDB состоит в том, что она предоставляет разработчику хранить различные BSON-документы без изменения логики приложения и моделей при разработке в паттерне MVC. BSON-документы могут иметь различную структуру. MongoDB позволяет сохранять и запрашивать денормализованные данные без дополнительных запросов данных из иных физических или логических источников: коллекций, файлов или таблиц. MongoEngine позволяет описывать документы в не строгом виде, используя модели, унаследованные от DynamicDocument. Объекты, унаследованные от данного класса, позволяют добавлять в БД документы с переменным числом полей и их типов.

Данные школ записываются в отдельные коллекции MongoDB. Для того, чтобы получить требуемую для реализации системы географическую информацию об учебных заведениях, необходимо произвести прямое геокодирование учебных заведений среднего образования по их адресам, представленным строкой (Приложение 1). Данные учебных заведений хранятся в том же Excel-файле, что и данные результатов экзаменов. Геокодирование производится с помощью вызовов API сервиса “Яндекс Геокодер”. После этого, в Python-объекты, содержащие информацию о школах, записываются координаты школ.

После извлечения и обработки данных школ и экзаменов итоговые наборы данные сливаются в единые Python объекты (Приложение 2), унаследованные от DynamicDocument и пакетно записываются в БД.

3.2. ЗАГРУЗКА И ИЗВЛЕЧЕНИЕ ДАННЫХ ИЗ БД

При обработке данных в БД строки с записями в Excel-Файлах о результатах экзаменов были преобразованы в массивы экземпляров классов Document с валидирующими ограничениями библиотеки MongoEngine.

После всех вышеперечисленных операций в подглаве 3.1 данные загружаются в хранилище MongoDB. База данных содержит несколько коллекций:

1. Необработанные данные с результатами экзаменов.
2. Необработанные данные с информацией о школах.
3. Обработанные данные школ с их координатами.
4. Полные данные ЕГЭ с информацией о школах (Приложение 3).

На Рисунке 1 приведены потоки данных в разработанной системе.

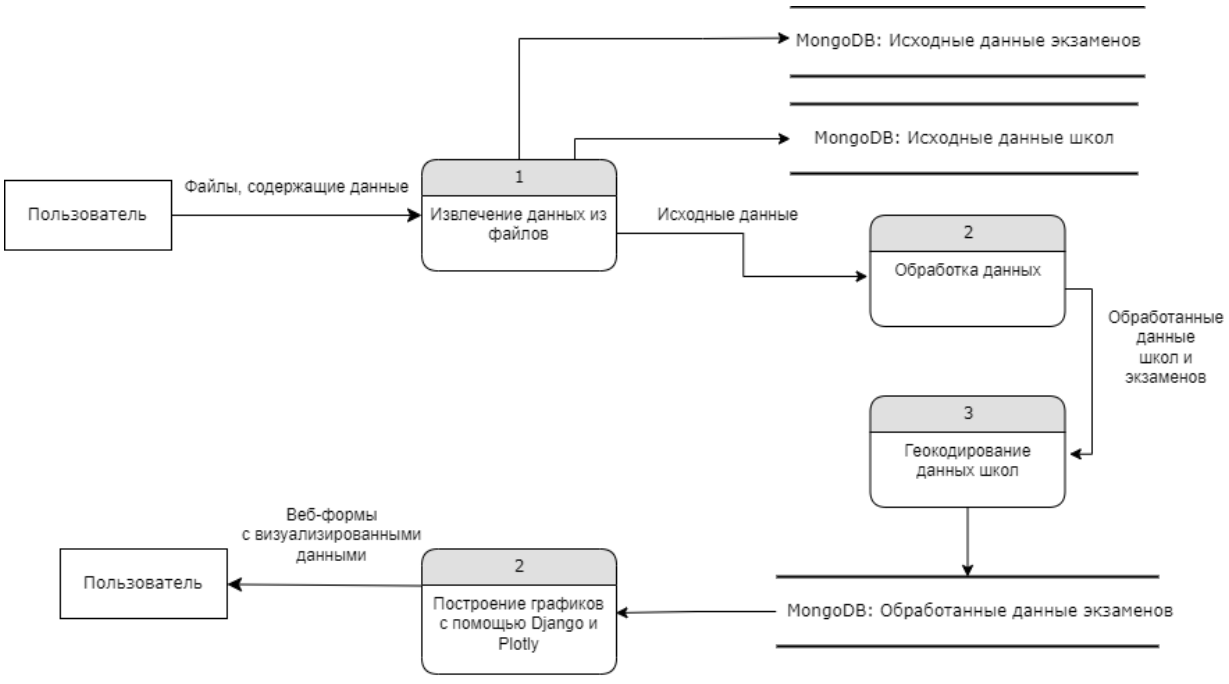

Рис. 1. Потоки данных в системе.

После обработки и загрузки данных в коллекции базы данных MongoDB итоговые документы коллекций имеют поля, представленные в Таблицах 4–7.

Таблица 4 включает документы коллекции exam_result_raw_info, содержащей результаты экзаменов из Excel таблиц с указанием года сдачи экзамена.

Таблица 4

Описание полей в коллекции с MongoDB exam_result_raw_info

Название	Тип данных	Описание	Пример данных
_id	ObjectId	Уникальный идентификатор документа	6410f0068bd63270ded41bdc
school_code	Int32	Код школы	201200
class_code	String	Код класса	“11А”
subject	Int32	Код предмета	1
subject_name	String	Название предмета	“Русский язык”
primary_score	Int32	Первичный балл	33
completion_percent	Int32	Процент выполнения	56
exam_score_100	Int32	Балл по 100-бальной шкале	57
simple_tasks_score	Int32	Балл за задания в части с кратким ответом	23
simple_tasks_result	String	Развернутые баллы за часть с краткими ответами	"1+-++-3---++++1-+++++-++4"
extra_tasks_score	Int32	Балл за задания в части с развернутым ответом	10
extra_tasks_result	String	Развернутые баллы за часть с развернутым ответом	"1(1)2(3)0(1)0(3)1(2)1(2)2(3)0(3)0(2)1(2)1(1)1(1)"
oral_tasks_score	Int32	Балл за задания в части с устным ответом	0

Продолжение таблицы 4

Название	Тип данных	Описание	Пример данных
oral_tasks_result	String	Развернутые баллы за часть с устным ответом	“”
exam_year	Int32	Год экзамена	2018

Таблица 5 включает документы коллекции school_raw_info, содержащей информацию об учебных заведениях из Excel таблиц с указанием года актуальности этой информации.

Таблица 5

Описание полей в коллекции MongoDB school_raw_info

Название поля	Тип данных	Описание	Пример данных
_id	ObjectId	Уникальный идентификатор документа	6410f0068bd63270ded41bdc
school_id	GUID	Уникальный идентификатор школы	A447EA29-0BE1-4CE7-B8C4-000B61B0ADB3
school_code	Int32	Код школы	201200
law_address	String	Адрес школы	626157, Тюменская обл., г. Тобольск, мкр. 7, д. 54
short_name	String	Краткое название школы	МАОУ "Гимназия имени Н.Д.Лицмана" г.Тобольска
school_kind_code	Int32	Код вида школы	103
school_kind_name	String	Вид школы	Гимназия
school_type_code	Int32	Код типа школы	1
school_type_name	String	Тип школы	Общеобразовательное учреждение/организация
township_name	String	Краткое название района (города)	Тобольск

Продолжение таблицы 5

Название поля	Тип данных	Описание	Пример данных
school_property_code	Int32	Код вида школы по юридической принадлежности	8
school_property_name	String	Вид школы по юридической принадлежности	Муниципальное автономное образовательное учреждение/организация
towntype_code	Int32	Код типа населенного пункта	2
towntype_name	String	Тип населенного пункта	Населенный пункт городского типа
area_code	Int32	Код района	143
area_name	String	Название района	г.Тобольск
government_code	Int32	Код контролирующего школу органа	243
government_name	String	Контролирующий школу орган	Комитет по образованию администрации г.Тобольска
year	Int32	Год, в который данные школы актуальны	2018

Таблица 6 включает документы коллекции school_info, содержащей обработанную информацию об учебных заведениях из Excel таблиц с указанием года актуальности этой информации. Также, данная коллекция включает в себя географические координаты учебных заведений, полученные прямым геокодированием по адресам (Приложение 1).

Таблица 6

Описание полей в коллекции MongoDB school_info

Название поля	Тип данных	Описание	Пример данных
_id	ObjectId	Уникальный идентификатор документа	6410f0068bd63270ded41bdc
school_code	Int32	Код школы	201200
law_address	String	Адрес школы	626157, Тюменская обл., г. Тобольск, мкр. 7, д. 54
short_name	String	Краткое название школы	МАОУ "Гимназия имени Н.Д.Лицмана" г.Тобольска
school_kind_name	String	Вид школы	Гимназия
school_type_name	String	Тип школы	Общеобразовательное учреждение/организация
township_name	String	Краткое название района (города)	Тобольск
school_property_name	String	Вид школы по юридической принадлежности	Муниципальное автономное образовательное учреждение/организация
towntype_name	String	Тип населенного пункта	Населенный пункт городского типа
area_name	String	Название района	г.Тобольск
government_name	String	Контролирующий школу орган	Комитет по образованию администрации г.Тобольска
school_coords.lon	Double	Долгота, на которой располагается школа	68.274109
school_coords.lat	Double	Широта, на которой располагается школа	58.231988
year	Int32	Год, в который данные школы актуальны	2018

Таблица 7 включает документы коллекции exam_result, содержащей обработанную информацию об учебных заведениях и результатах экзаменов из Excel таблиц с указанием года актуальности этой информации. Также, данная коллекция включает в себя географические координаты учебных заведений. При формировании данной коллекции были выделены новые признаки, содержащие нормированные доли выполнения заданий ЕГЭ из частей с кратким ответом, развернутым и устным. Данные, содержащиеся в данной коллекции, являются конечным результатом обработки исходных данных. Данная коллекция MongoDB используется для анализа и визуализации в разработанной системе.

Таблица 7

Описание полей в коллекции MongoDB exam_result

Название поля	Тип данных	Описание	Пример данных
_id	ObjectId	Уникальный идентификатор документа	6410f0068bd63270ded41bdc
school_code	Int32	Код школы	201200
law_address	String	Адрес школы	626157, Тюменская обл., г. Тобольск, мкр. 7, д. 54
short_name	String	Краткое название школы	МАОУ "Гимназия имени Н.Д.Лицмана" г.Тобольска
school_kind_name	String	Вид школы	Гимназия
school_type_name	String	Тип школы	Общеобразовательное учреждение/организация
township_name	String	Краткое название района (города)	Тобольск
school_property_name	String	Вид школы по юридической принадлежности	Муниципальное автономное образовательное учреждение/организация
towntype_name	String	Тип населенного пункта	Населенный пункт городского типа
area_name	String	Название района	г.Тобольск

Продолжение таблицы 7

Название поля	Тип данных	Описание	Пример данных
government_name	String	Контролирующий школу орган	Комитет по образованию администрации г.Тобольска
school_coords.lon	Double	Долгота, на которой располагается школа	68.274109
school_coords.lat	Double	Широта, на которой располагается школа	58.231988
class_code	String	Код класса	“11А”
subject	Int32	Код предмета	1
subject_name	String	Название предмета	“Английский язык”
primary_score	Int32	Первичный балл	33
completion_percent	Int32	Процент выполнения	56
exam_score_100	Int32	Балл по 100-бальной шкале	57
simple_tasks_score	Int32	Балл за задания в части с кратким ответом	23
extra_tasks_score	Int32	Балл за задания в части с развернутым ответом	10
oral_tasks_score	Int32	Балл за задания в части с устным ответом	11
exam_year	Int32	Год экзамена	2018
simple_task_1	Double	Доля полученных баллов за 1-е задание с кратким ответом	1
simple_task_n	Double	Доля полученных баллов за n-ое задание с кратким ответом	0.25

Продолжение таблицы 7

Название поля	Тип данных	Описание	Пример данных
extra_task_1	Double	Доля полученных баллов за 1-е задание с развернутым ответом	0.5
extra_task_n	Double	Доля полученных баллов за n-ое задание с развернутым ответом	1
oral_task_1	Double	Доля полученных баллов за 1-е задание с устным ответом	0.5
oral _task_n	Double	Доля полученных баллов за n-ое задание с устным ответом	0.6

Для извлечения данных из БД используется библиотека PyMongoArrow, преимущества использования которой указаны в Главе 2 данной работы. Обработанные данные из коллекции exam_result MongoDB, содержащей индексы по названию предмета и году выполнения экзамена, извлекаются из БД и записываются в объекты DataFrame библиотеки pandas (Приложение 4). Фильтрация для запроса на получение данных выполняется по названию учебного предмета и году выполнения экзамена.

3.3. ВЫЧИСЛЕНИЕ СТАТИСТИЧЕСКИХ ХАРАКТЕРИСТИК

После получения данных из БД выполняются вычисления различных статистических характеристик тестовых данных. Смещение среднего значения влево или вправо может указывать на ситуацию, когда задания теста слишком трудные или, наоборот, слишком легкие. Если результаты испытуемых имеют небольшую вариацию, это может указывать на низкое качество самого теста. Асимметрия распределения является положительной, если большая часть

индивидуальных баллов находится справа от среднего значения. Это обычно характерно для излишне легких тестов. Асимметрия распределения баллов будет отрицательной, если большинство учеников получили оценки ниже среднего балла. Отрицательная асимметрия часто встречается в излишне трудных тестах, которые не были правильно сбалансированы по трудности при составлении заданий для теста [22].

3.4. ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

Для отображения экзаменационной аналитики были разработаны 4 приложения Dash, интегрированные в проект Django средствами библиотеки Django-plotly-dash [6]:

1. Экзаменационную аналитику по выбранному предмету экзамена и году экзамена;
2. Экзаменационную аналитику всех результатов экзаменов за все годы, хранящиеся в БД;
3. Экзаменационную аналитику тестовых заданий;
4. Экзаменационную аналитику участников экзамена с низкими баллами.

На главной странице веб-приложения (Рисунок 2) присутствует возможность выбора года экзамена и предмета с помощью элементов Dropdown. После соответствующего выбора формируется географическая карта, включающая учебные заведения, отмеченные метками (Приложение 5). Также на форме выводятся количество участников экзамена и статистические характеристики баллов участников экзамена: медианные и средние баллы, стандартное отклонение баллов, коэффициенты асимметрии и эксцесса баллов.

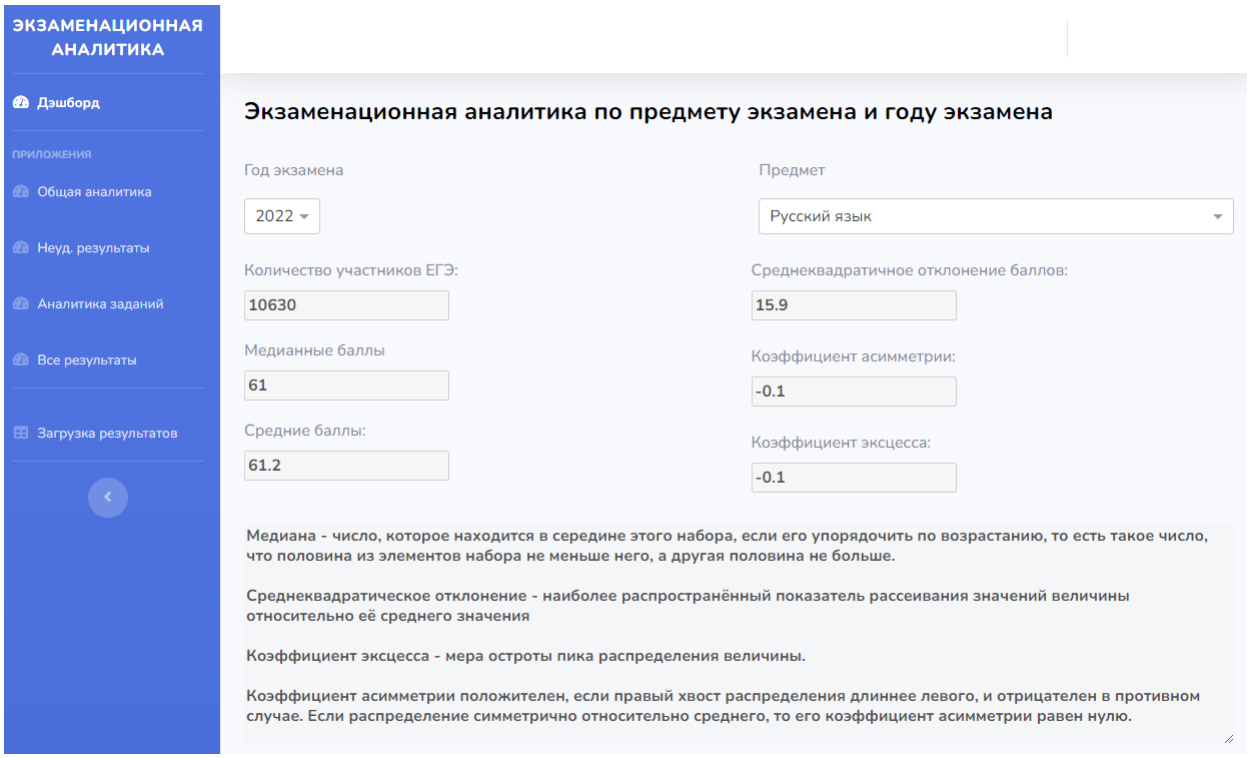

Рис. 2. Приложение Django Dash с аналитикой результатов экзамена по году и предмету.

При наведении курсора на учебное заведение отображается окно, содержащее статистические характеристики результатов выполнения заданий итоговой аттестации учащихся данного учебного заведения, а также информация об учебном заведении: название, код, название выбранного предмета, количество участников экзамена от данного учебного заведения, их медианные и средние баллы, стандартное отклонение баллов, коэффициенты асимметрии и эксцесса баллов (Рисунок 3). Также при наведении курсора на метку учебного заведения выводится номер кластера, к которому оно было отнесено после выполнения кластеризации и список проблемных заданий. Проблемными заданиями считаются те задания, с которыми справились менее половины участников экзамена.

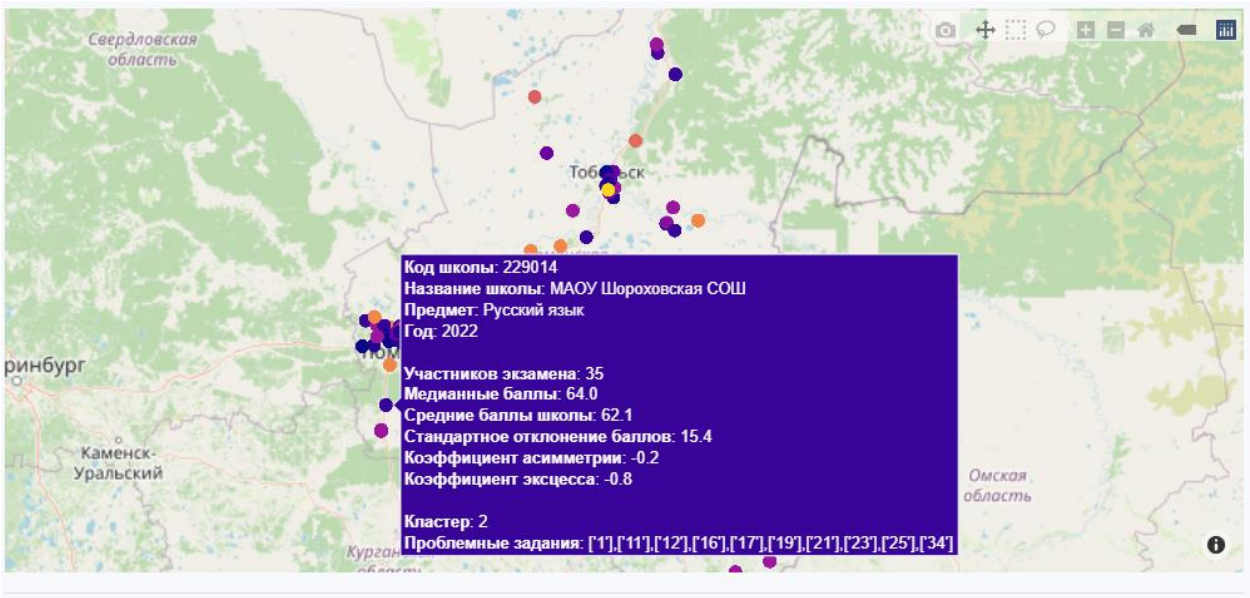

Рис. 3. Географическая визуализация учебных заведений.

Графики для просмотра распределения баллов итоговой аттестации учащихся и процента их выполнения (Рисунки 4–5).

Рис. 4. Визуализация распределения итоговых тестовых баллов ГИА по 100-бальной шкале.

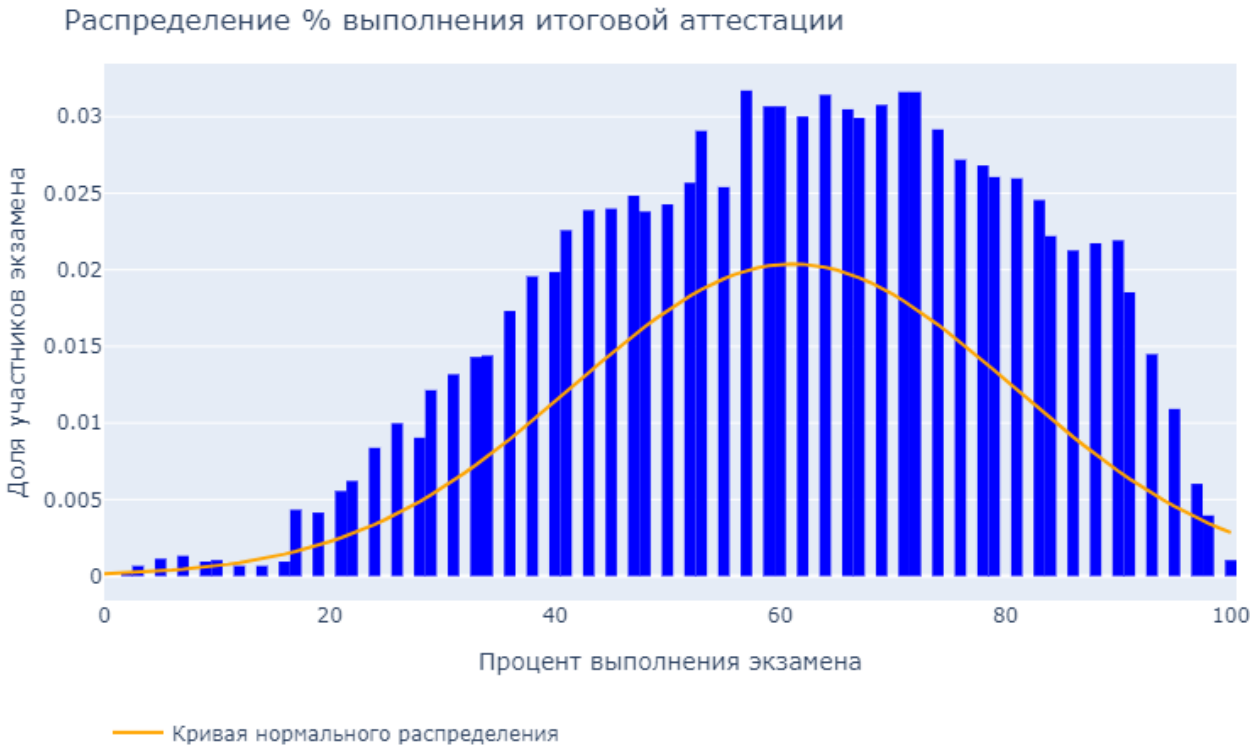

Рис. 5. Визуализация распределения процента выполнения итоговой аттестации.

При получении данных из хранилища для отображения их на соответствующих формах в виде наглядных графиков производится агрегация данных с группировкой по школам, районам Тюменской области (Рисунки 6–7), видам и типам школ, населенным пунктам.

Рис. 6. Распределение баллов по районам Тюменской области.

Рис. 7. Гистограмма количества участников по районам Тюменской области.

Отдельно вынесены распределения баллов участников по городам (Рисунки 8–9). Это сделано для возможности визуального ранжирования районов, т. к. при помещении на один и тот же график районов и городов, города масштабируют все графики.

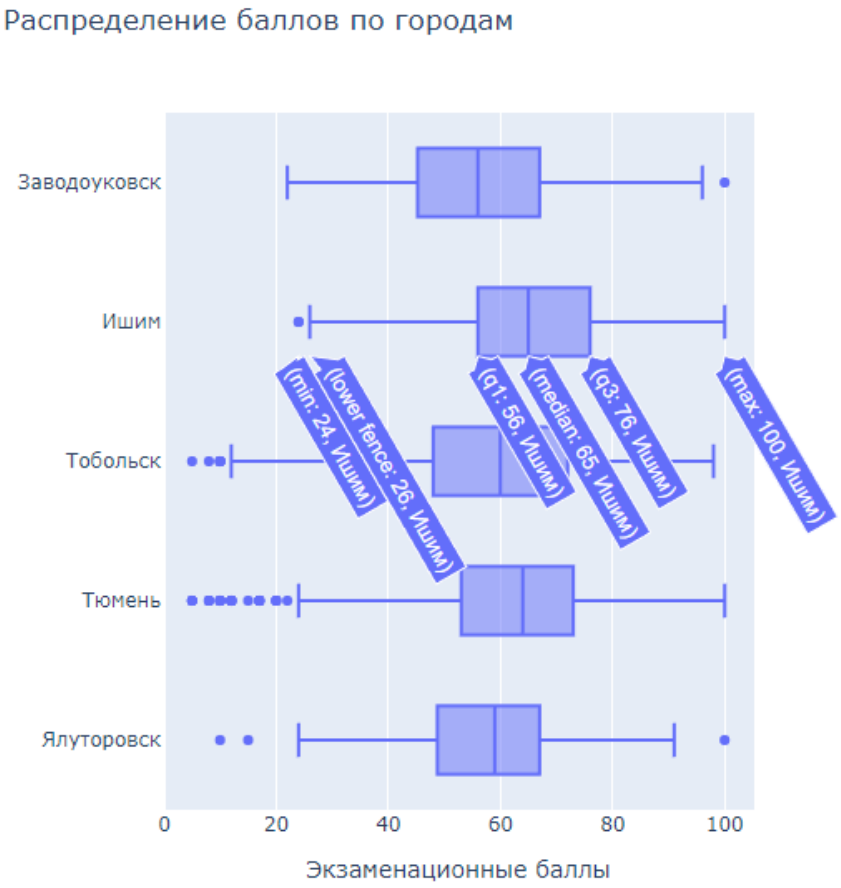

Рис. 8. Распределение баллов по городам Тюменской области.

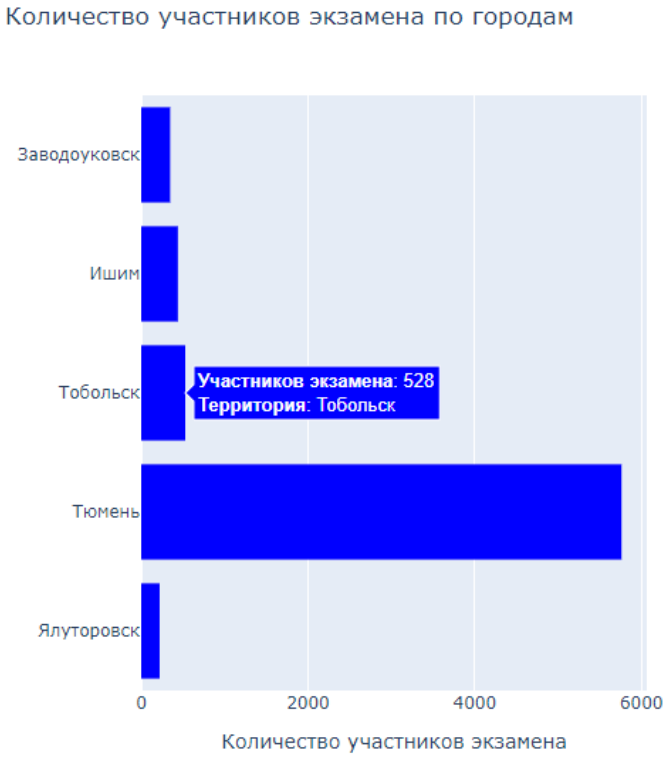

Рис. 9. Распределение баллов по городам Тюменской области.

Приложение позволяет увидеть распределение баллов (Рисунок 10) и количество участников (Рисунок 11) по видам учебных заведений.

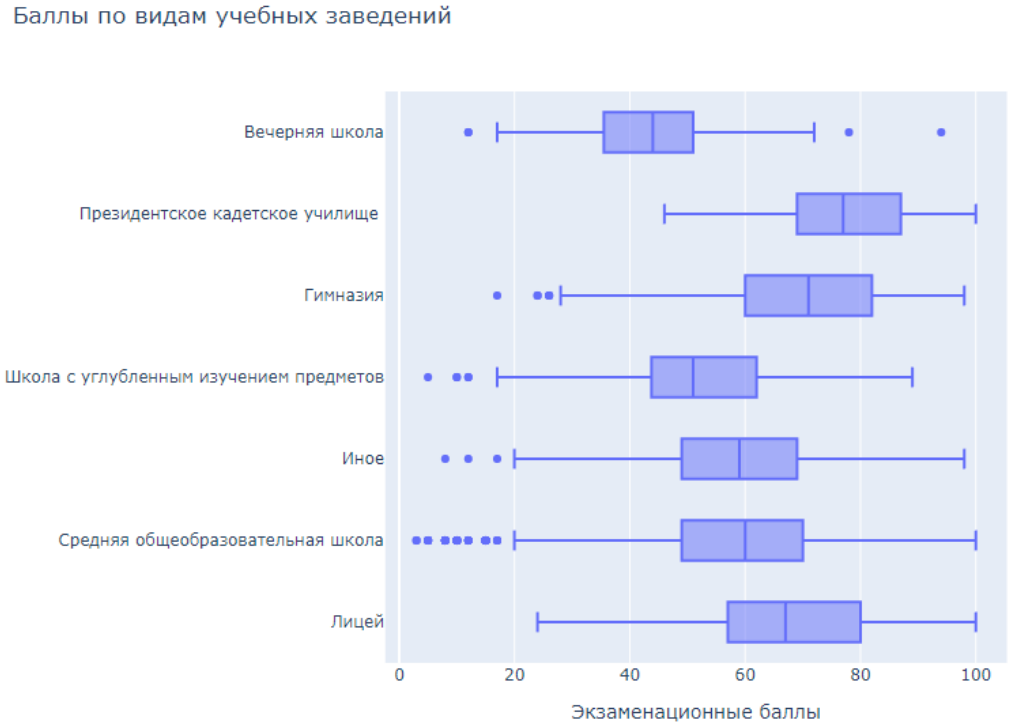

Рис. 10. Распределение итоговых баллов ГИА по видам учебных заведений.

Рис. 11. Диаграмма визуализации количества участников ГИА по видам учебных заведений.

Другое Dash-приложение формирует визуальные гистограммы (Приложение 6), предоставляющие информацию о популярности выбора предмета экзамена по различным годам (Рисунок 12) и количество участников обязательных экзаменов (Рисунок 13).

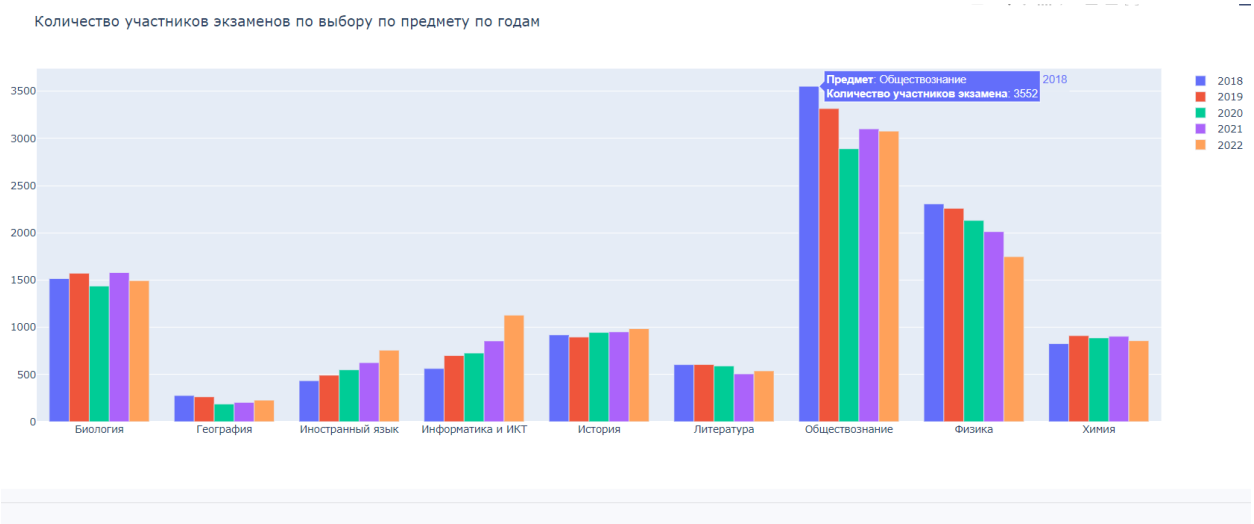

Рис. 12. Гистограмма популярности выбора предмета по годам в приложении с аналитикой всех результатов экзаменов.

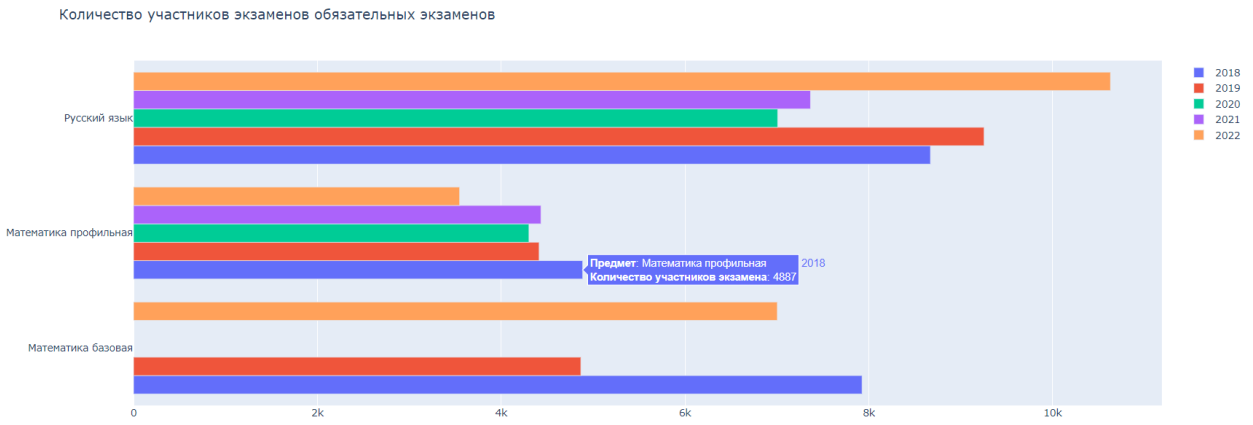

Рис. 13. Гистограмма с количеством участников обязательных экзаменов по годам.

Dash-приложение с аналитикой тестовых заданий и их характеристиками включает в себя таблицу, матрицу корреляции, справочную информацию и элементы управления для выбора предмета и года экзамена. Таблица содержит 3 столбца: номер задания, рассчитанный коэффициент дискриминативности задания, рассчитанный коэффициент трудности задания

(Рисунок 15). Матрица корреляции содержит все задания экзамена (Рисунок 14). При наведении курсора на соответствующие столбец и строку данной матрицы пользователь видит коэффициент корреляции результатов двух выбранных заданий и их номера. Справочная информация для пользователей изложена неформальным языком (Рисунок 15).

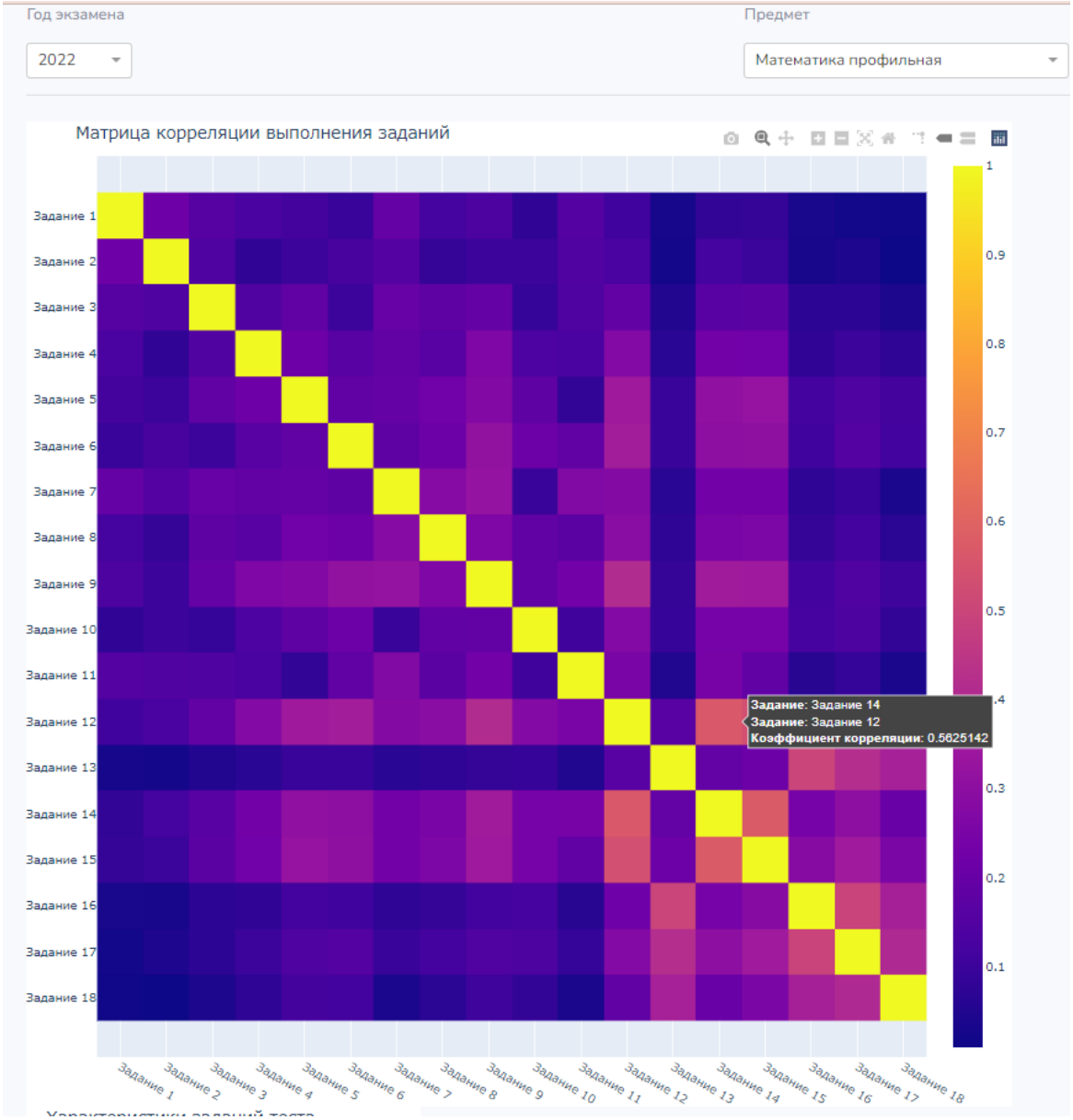

Рис. 14. Приложение Django Dash с аналитикой тестовых заданий (матрица корреляции).

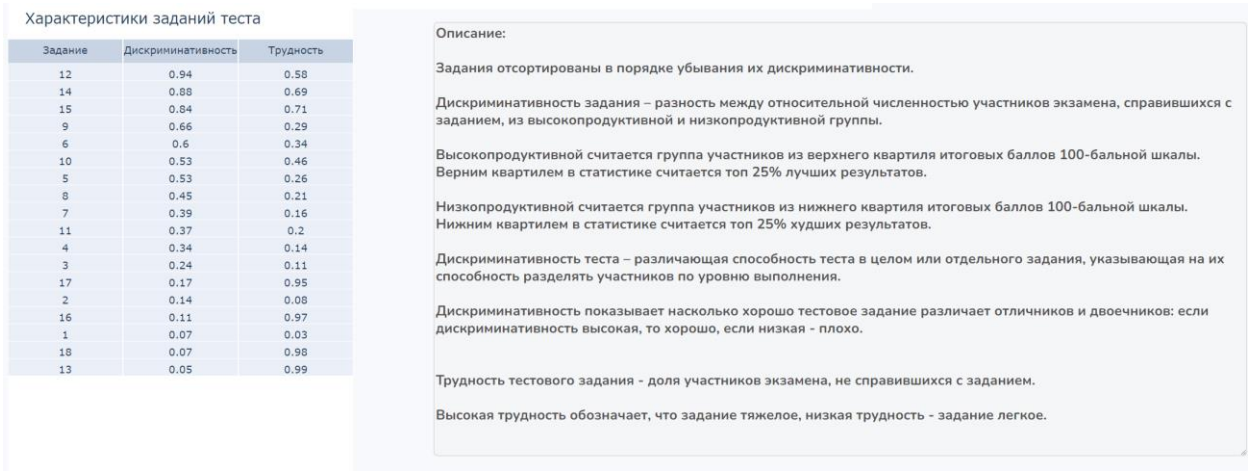

Рис. 15. Приложение Django Dash с аналитикой тестовых заданий (характеристики и справочная информация).

Система содержит Dash-приложение, в котором отображаются учебные заведения, в которых имеет место на выбранный год количество участников экзамена, не набравших пороговый балл по выбранному предмету, более 1 участника (Рисунок 16). Метки учебных заведений на географической карте имеют различные цвета. Цвет зависит от доли участников экзамена из данного учебного заведения, не набравших пороговый балл выбранного экзамена за определенный год.

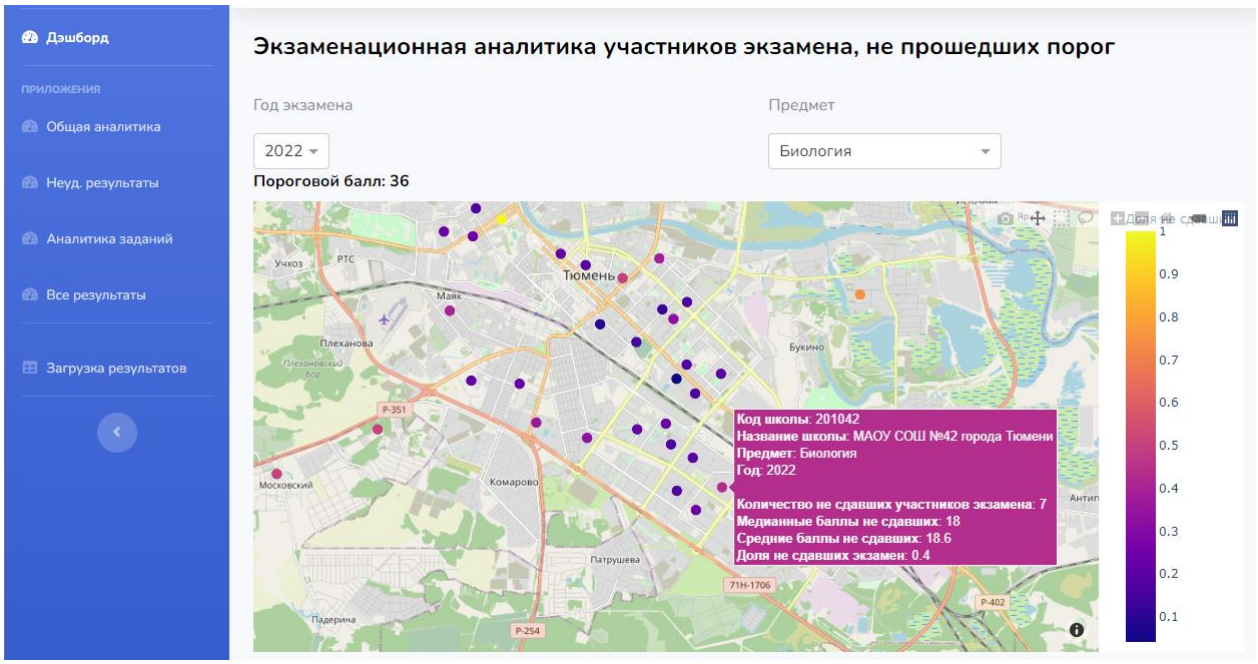

Рис. 16. Приложение Django Dash, отображающее результаты экзамена с баллами ниже пороговых.

Также, одно из Dash-приложений содержит географическую визуализацию общего количества участников всех экзаменов за все годы в БД относительно учебных заведений, их медианные, средние баллы, разброс баллов (Рисунок 17).

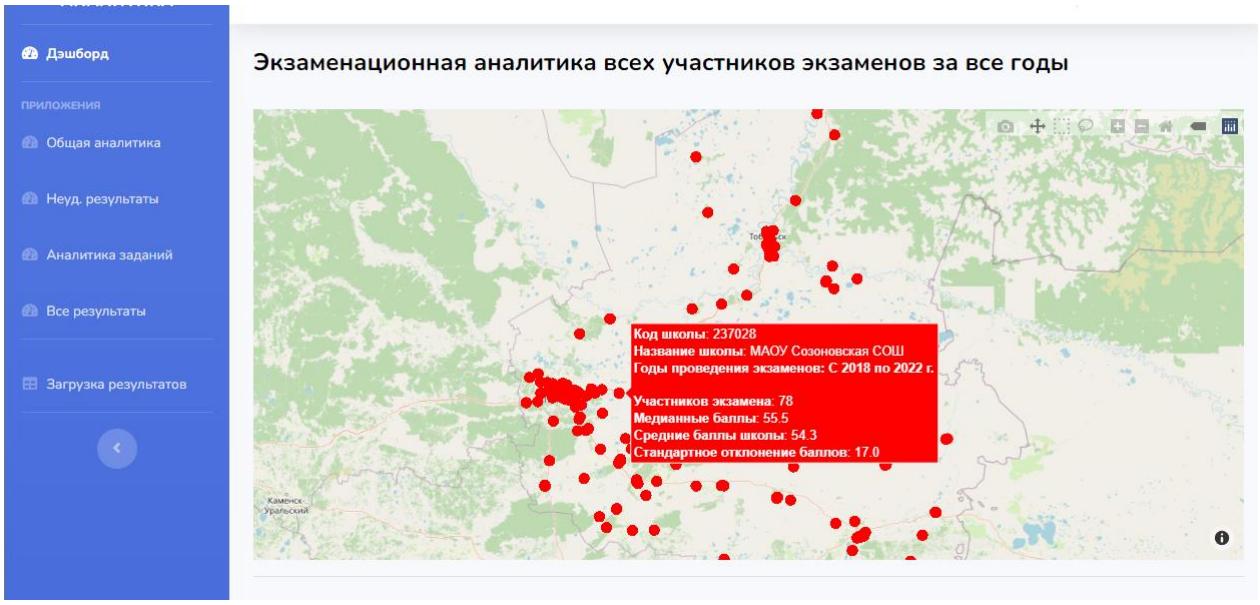

Рис. 17. Приложение Django Dash, отображающее все результаты экзаменов.

Данные загружаются в хранилище (Рисунок 18) 1 раз в год после подсчета и получения результатов экзаменов. При повторной загрузке данных за конкретный год, существующие в БД данные удаляются. Периодическая подгрузка данных не производится.

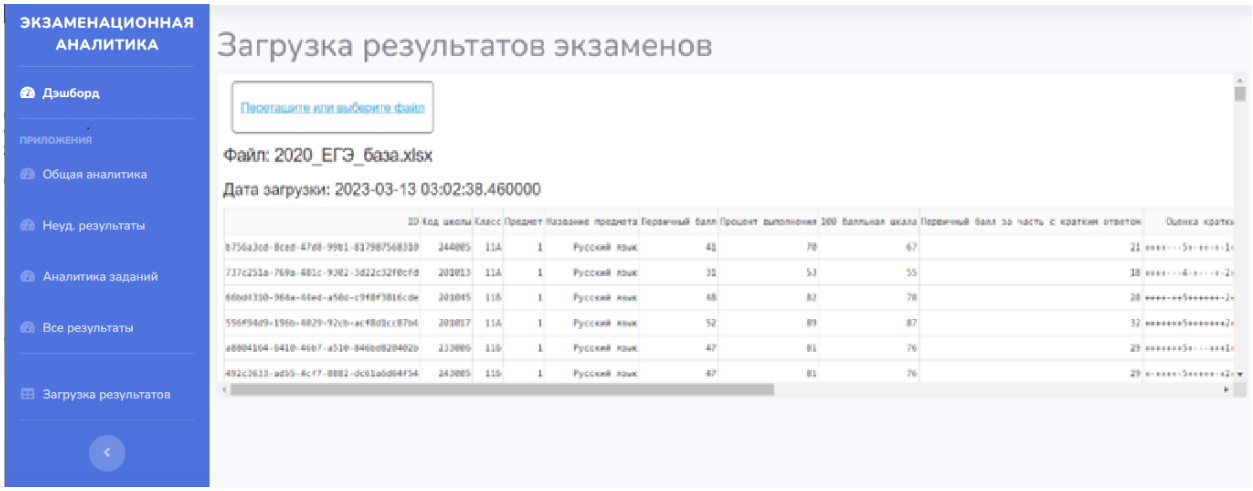

Рис. 18. Форма загрузки результатов экзамена.

ЗАКЛЮЧЕНИЕ

В результате выполнения ВКР все поставленные задачи были успешно выполнены.

В начале работы были изучены предоставленные ТОГИРРО данные экзаменационных работ и данные о школах Тюменской области. Затем, исходные данные были предобработаны. Были выделены особенности исходных данные, некоторые записи были дополнительно обработаны.

В рамках обработки экзаменационных данных были изучены и выбраны необходимые инструменты, технологии и библиотеки для извлечения, предобработки и валидации исходных данных; были выделены новые признаки, выполнена визуализация значений различных числовых признаков.

Были отобраны и сравнены между собой различные популярные сервисы геокодирования. Затем, было реализовано прямое геокодирования учебных заведений по их адресам, представленным в строковом виде.

После обработки и валидации данные с результатами экзаменов по всем общеобразовательным предметам для Тюменской области в период с 2018 по 2022 включительно годы были загружены в виде документов коллекций документно-ориентированной базы данных MongoDB.

Итоговое приложение для автоматизированной обработки и визуализации экзаменационных данных итоговой аттестации было реализовано с помощью веб-фреймворка Django. Для визуализации данных были созданы 4 Dash приложения с использованием библиотеки Django-dash-plotly, содержащие формы, диаграммы и графики различных распределений тестовых данных.

В рамках разработанной системы были реализованы средства анализа различных характеристик тестов и создан инструмент кластеризации учебных заведений по степени схожести выполнения отдельных тестовых заданий, позволяющий выделять учебные заведения с похожими проблемами в выполнении заданий ЕГЭ и отображать их на географической карте.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. 2ГИС Geocoder API. URL: https://docs.2gis.com/ru/api/search/geocoder/overview (Дата обращения: 28.04.2023).

2. BEST Version Control Software (Source Code Management Tools). URL: https://www.softwaretestinghelp.com/version-control-software (Дата обращения 10.05.2023).

3. Compass. The GUI for MongoDB. URL: https://www.mongodb.com/products/compass (Дата обращения 20.02.2023).

4. DaData Геокодирование (координаты по адресу). URL: https://dadata.ru/api/geocode (28.04.2023).

5. Dash Plotly User Guide. URL: https://dash.plotly.com (Дата обращения: 29.03.2023).

6. Django-plotly-dash. Plotly Dash applications served up in Django templates using tags. URL: https://django-plotly-dash.readthedocs.io/en/latest (Дата обращения: 03.05.2023).

7. Google Maps Platform Geocoding API overview. URL: https://developers.google.com/maps/documentation/geocoding/overview (Дата обращения: 01.05.2023).

8. How to Use Python with MongoDB. URL: https://www.mongodb.com/languages/python (Дата обращения: 11.04.2023).

9. Kate Brush, Ed Burns. What is data visualization? URL: https://www.techtarget.com/searchbusinessanalytics/definition/data-visualization (Дата обращения: 03.04.2023).

10. MongoEngine User Guide 2.3.4. Document collections. URL: https://docs.mongoengine.org/guide/defining-documents.html#document-collections (Дата обращения: 28.05.2023).

11. NumPy documentation. URL: https://numpy.org/doc/stable (Дата обращения: 03.03.2023).

12. Pandas. Getting Started. URL: https://pandas.pydata.org/docs/getting_started/index.html (Дата обращения: 02.03.2023).

13. Pandas.DataFrame.hist. URL: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.hist.html (Дата обращения: 19.04.2023).

14. PyMongoArrow Documentation. URL: https://mongo-arrow.readthedocs.io/en/latest (Дата обращения: 19.04.2023).

15. SciPy Developer Documentation. URL: https://docs.scipy.org/doc/scipy/dev/index.html (Дата обращения: 03.02.2023).

16. The Python Programming Language. URL: https://www.tiobe.com/tiobe-index/python (Дата обращения: 26.04.2023).

17. Top 10 Best Container Software In 2023. URL: https://www.softwaretestinghelp.com/container-software (Дата обращения 17.04.2023).

18. Wes McKinney and the Pandas Development Team: pandas: powerful Python data analysis toolkit. Release 1.4.4. URL: https://pandas.pydata.org/pandas-docs/version/1.4.4/pandas.pdf (Дата обращения: 10.04.2023).

19. Визуализация данных — что это: виды, способы и средства визуального представления информации. 24.06.2022. URL: https://alexkolokolov.com/ru/blog/vizualizaciya-dannyh-chto-eto (Дата обращения: 25.02.2023).

20. Геокодирование адреса, координаты по адресу, API maps, карты. URL: https://geotree.ru/geocoder (Дата обращения: 01.05.2023).

21. Клюев М. В., Поздняков Е. А. РАЗЛИЧИЯ BOOTSTRAP4 И BOOTSTRAP5 ПРИ РАЗРАБОТКЕ WEB-САЙТОВ // Цифровая наука. 2020. №7. URL: https://cyberleninka.ru/article/n/razlichiya-bootstrap4-i-bootstrap5-pri-razrabotke-web-saytov (Дата обращения: 21.03.2023).

22. Лазарева Е. Г., Устинова И. Г. Ранжирование трудности тестовых заданий с учетом угадывания // Образовательные ресурсы и технологии. 2016. №2. С. 44–50.

23. Расчёт и анализ характеристик теста для повышения уровня педагогического измерения. URL: https://tester.quali.me/help.php#diskr (Дата обращения: 12.04.2023).

24. Что такое дискриминативность и как ее использовать? URL: https://www.opentest.ru/support/260-chto-takoe-diskriminativnost-i-kak-ee-ispolzovat (Дата обращения: 14.03.2023).

25. Яндекс Геокодер API для перевода географических координат в адрес и наоборот. URL: https://yandex.ru/dev/maps/geocoder/?from=mapsapi (Дата обращения: 28.04.2023).

Приложение 1

Функция прямого геокодирования координат школ по их адресам

```
     def fetch_coords_by_geocode(url: str, api_key: str,
address: str) -> dict[str, float]:
     import requests
     try:
     url =
f'{url}/?apikey={api_key}&format=json&lang=ru_RU&results=1&geoco
de={address}'
     response = requests.get(url).json()
     coords =
response['response']['GeoObjectCollection']['featureMember'][0][
'GeoObject']['Point']['pos']
     lon, lat = coords.split()
     except Exception as e:
     print(f'Error while fetching coords by geocode: {e}')
     lon, lat = 0, 0
     return {'lon': float(lon), 'lat': float(lat)}
```

Приложение 2

Описание модели MongoEngine для хранения полной информации о результате экзамена и учебного заведения

```
    class ExamResult(DynamicDocument):
        school_code = IntField(max_value=999999, required=True)
        law_address = StringField(required=True)
        short_name = StringField(required=True)
        school_kind_name = StringField(required=True)
        school_type_name = StringField(required=True)
        township_name = StringField(required=True)
        school_property_name = StringField(required=True)
        towntype_name = StringField(required=True)
        area_name = StringField(required=True)
        government_name = StringField(required=True)
        school_lat = FloatField(required=False)
        school_lon = FloatField(required=False)

        class_code = StringField(max_length=14, required=True)
        subject_name = StringField(max_length=100,
required=True)
        primary_score = IntField(min_value=0, required=True)
        completion_percent = IntField(min_value=0,
max_value=100, required=True)
        exam_score_100 = IntField(min_value=0, max_value=100,
required=True)
        simple_tasks_score = IntField(min_value=0,
required=True)
        extra_tasks_score = IntField(min_value=0, required=True)
        oral_exam_score = IntField(min_value=0, required=True)
        exam_year = IntField(min_value=2000, required=True)

        meta = {'indexes': [('exam_year', '-subject_name')]}
```

Приложение 3

Пример обработанных результатов экзаменов с информацией о школах


```
{
  "_id": {
    "$oid": "64163d083adc4e12aa206bfd"
  },
  "school_info": {
    "school_code": 201200,
    "law_address": "625000, Тюменская обл., г. Тюмень, ул.
Республики, д. 17",
    "short_name": "Департамент образования Администрации
г.Тюмени",
    "school_kind_name": "Иное",
    "school_type_name": "Иные",
    "township_name": "Тюмень",
    "school_property_name": "Иное",
    "towntype_name": "Населенный пункт городского типа",
    "area_name": "г.Тюмень",
    "government_name": "Департамент образования
Администрации города Тюмени",
    "school_coords": {
      "lon": 65.531309,
      "lat": 57.157993
    },
    "year": 2018
  },
  "class_code": "11",
  "subject_name": "Русский язык",
  "primary_score": 33,
  "completion_percent": 56,
  "exam_score_100": 57,
  "simple_tasks_score": 23,
  "simple_tasks_result": [
    {
      "result": 1,
      "max_possible_result": 2,
      "task_number": 1,
      "task_local_number": 1
    },
    ….
  ],
  "extra_tasks_score": 10,
```

Продолжение приложения 3


```
 "extra_tasks_result": [
    {
      "result": 1,
      "max_possible_result": 1,
      "task_number": 26,
      "task_local_number": 1
    },
…
  ],
  "oral_exam_score": 0,
  "oral_exam_result": [],
  "exam_year": 2018
}
```

Приложение 4

Функция извлечения данных ЕГЭ из коллекции базы MongoDB

```
def fetch_all_to_df(exam_year: int, subject_name: str):
    query_filter = {"$and": [{"exam_year": int(exam_year)},
{"subject_name": subject_name}]}

    fetched_document_fields: Mapping[str, Any] =
__get_mongo_db_collection().find_one(filter=query_filter)

    fetched_document_dict = {k: v for i, (k, v) in
enumerate(fetched_document_fields.items()) if i >= 1}  # skip id
    field_names_with_types_dict: dict = {field: type(value) for
field, value in fetched_document_dict.items()}

    result: pd.DataFrame =
__get_mongo_db_collection().find_pandas_all({
        'subject_name': {'$regex': subject_name},
        'exam_year': int(exam_year)
    }, schema=Schema(field_names_with_types_dict))

    return result
```

Приложение 5

Функция формирования географической карты учебных заведений

```
def _mapbox(df: pd.DataFrame) -> Figure:
    mapbox = px.scatter_mapbox(
        data_frame=df,
        lat="school_lat",
        lon="school_lon",
        hover_name="law_address",
        hover_data=["school_code", "short_name",
"subject_name"],
        color_discrete_sequence=["red"],
        zoom=5,
        mapbox_style="open-street-map",
        opacity=1,
    )
    mapbox.update_traces(lat=df['school_lat'],
                         lon=df['school_lon'],
                         text=df['law_address'],
                         customdata=np.stack(
                             (
                                 df['school_code'],
                                 df['short_name'],
                                 df['subject_name'],
                                 df['exam_year'],
df.groupby('school_code')['exam_score_100'].transform('count'),
df.groupby('school_code')['exam_score_100'].transform('median'),
df.groupby('school_code')['exam_score_100'].transform('mean'),
df.groupby('school_code')['exam_score_100'].transform('std'),
                             ), axis=-1),
                         hovertemplate="<b>Код школы</b>:
%{customdata[0]}<br>" +
                                       "<b>Название школы</b>:
%{customdata[1]}<br>"+
                                       "<b>Предмет</b>:
%{customdata[2]}<br>" +
                                       "<b>Год</b>:
%{customdata[3]}<br>" +
                                       "<b>Участников
экзамена</b>: %{customdata[4]}<br>" + "<b>Медианные баллы</b>:
%{customdata[5]:.1f}<br>" +                              "<b>Средние
баллы школы</b>: %{customdata[6]:.1f}<br>" +
"<b>Стандартное отклонение баллов</b>:
%{customdata[7]:.1f}<br>",
overwrite=True)
    mapbox.update_layout(margin={"r": 0, "t": 0, "l": 0, "b":
0})
    return mapbox
```

Приложение 6

Функция формирования графика популярности выбора предмета ЕГЭ

```
def _popularity_subjects_stack_chart(df: pd.DataFrame) ->
Figure:
        figure = go.Figure()

        years = df['exam_year'].unique().tolist()
        for exam_year in years:
            df_by_year: pd.DataFrame = df[df['exam_year'] ==
    int(exam_year)]
            df_grouped =
    df_by_year.groupby('subject_name')['exam_score_100'].agg(['
    count']).reset_index()

            df_grouped = df_grouped.drop(
    df_grouped[(df_grouped['subject_name'].str.contains('Матема
    тика')
                                |
    df_grouped['subject_name'].str.contains('Русский'))].index
            )
            figure.add_trace(
                go.Bar(
                    x=df_grouped['subject_name'],
                    y=df_grouped['count'],
                    orientation='v',
                    name=str(exam_year),
                    hovertemplate="<b>Предмет</b>: %{x}<br>" +
                                  "<b>Количество участников
    экзамена</b>: %{y}<br>"
                )
            )
        figure.update_layout(title="Количество участников
     экзаменов по выбору по предмету по годам", height=600)
    return figure
```