М.М. Аббаси, А.П. Бельтюков

Использование алгоритма машинного обучения (кластеризация) для анализа клинических данных пациентов с целью выявления пациентов с редкими симптомами

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Удмуртский государственный университет»
Институт математики, информационных технологий и физики
Кафедра вычислительных технологий и интеллектуальных систем

М.М. Аббаси, А.П. Бельтюков

Использование алгоритма машинного обучения (кластеризация) для анализа клинических данных пациентов с целью выявления пациентов с редкими симптомами

Монография

Ижевск
2025

УДК 004.2(075.8)
ББК 32.971.3
А137
Рекомендовано к изданию редакционно-издательским советом УдГУ

Рецензенты: д-р техн. наук, профессор, зав. каф. автоматизированных систем управления Уфимского университета науки и технологий **В.В. Антонов,**
канд. тех. наук, доцент, каф. вычислительных технологий и интеллектуальных систем больших данных Удмуртского государственного университета **С.Г. Маслов.**

Аббаси М.М., Бельтюков А.П.

А137 Использование алгоритма машинного обучения (кластеризация) для анализа клинических данных пациентов с целью выявления пациентов с редкими симптомами : монография / М.М. Аббаси, А.П. Бельтюков. – Ижевск : Удмуртский университет, 2025. – 112 с.

ISBN 978-5-4312-1296-3
DOI 10.35634/978-5-4312-1296-3-2025-1-112

Монография предназначена для студентов, обучающихся по направлению «Прикладная информатика», и студентов других смежных направлений, изучающих принципы управления проектами и разработку нового компьютерного алгоритма. Монография также может быть использована молодыми исследователями и учёными, которые занимаются разработкой моделей для искусственного интеллекта и интеллектуальным анализом данных.

УДК 004.2(075.8)
ББК 32.971.3

ISBN 978-5-4312-1296-3
DOI: 10.35634/978-5-4312-1296-3-2025-1-112

ВВЕДЕНИЕ

Актуальность темы исследования. Выявление и отделение отклоняющихся, редких или необычных значений от обычных в большой базе данных в последнее время стало очень популярным. Это помогает эффективно и результативно понимать и анализировать данные. Применение в анализе изображений, анализе данных веб-сайтов, биоинформатике, статистике делает его более удобным и приемлемым для использования, особенно в современных исследованиях. Было замечено, что в среднем около 15–20 % значений являются необычными или редко встречающимися. В большом проценте случаев они содержат очень важную и интересную информацию, которую нельзя просто отбросить из обычных данных. Для выявления редких случаев элементов данных, используются методы искусственного интеллекта и машинного обучения. Кластеризация – один из них, широко применяемая технология для понимания и изучения набора данных. В этой исследовательской работе основное внимание уделяется медицинским данным, собранным международной аналитической организацией в различных больницах, расположенных в нескольких странах по всему миру. Данные собираются с помощью опросника, заполняемого пациентами, а затем сохраняются в базе данных. Чтобы применить кластеризацию к собранным данным, нужно было провести подробное исследование алгоритмов кластеризации, а затем выбрать наиболее подходящие из них, которые наилучшим образом соответствуют нашим потребностям и требованиям к анализу баз данных.

Каждый из алгоритмов кластеризации применяется к базе данных, и полученные результаты сравниваются. Цель состоит в том, чтобы предложить модель, которая обеспечивает механизм для эффективной идентификации необычных или редких элементов базы данных. В конце мы проведём их оценку и объясним, почему предложенная методология лучше подходит для выявления редких случаев в наших данных.

Медицинская база данных, используемая для исследования, содержит информацию о пациентах, их истории болезни, личную информацию, текущий диагноз, лекарства и текущую ситуацию. В медицинских исследованиях одним из первых симптомов заболевания человека или нарушения работы его органа является боль. Сама боль иллюстрирует тот факт, здоров организм или нет. Боль обычно определяется как неприятное сенсорное или эмоциональное состояние, возникающее из-за фактического или потенциального повреждения тканей. Это также помогает нам защитить наш организм, мотивируя нас избегать опасных раздражителей и лечить повреждённые участки тела. Боль бывает разной степени и варьируется от незначительной до очень сильной.

В медицинских науках боль рассматривается как основной симптом, отражающий состояние здоровья живого организма. Боль связана с физической, умственной, эмоциональной и всеми другими видами деятельности человека и влияет на все эти виды деятельности. Аналогичным образом, эти виды деятельности также могут вызывать боль. Поскольку боль влияет на нормальную жизнедеятельность, от неё необходимо избавиться как можно раньше.

Как правило, боль классифицируется на основе её продолжительности и тяжести, анатомической локализации, вовлечения систем организма, причин и временных характеристик. Однако есть и несколько других способов. У живых организмов есть порог восприятия боли, и после достижения этого порога боль становится ощутимой, и организм начинает действовать, чтобы её остановить. Порог переносимости боли у одного человека может отличаться от порога переносимости боли у другого человека из-за происхождения, пола, физических особенностей и многих других причин.

Для измерения или оценки боли можно использовать несколько способов. Один из них – попросить пациентов оценить интенсивность боли по шкале от 0 до 10, где 0 означает отсутствие боли, а 10 – наивысшую степень боли. Аналогичным образом, если пациент не может самостоятельно оценить свою боль, для определения интенсивности боли по его поведению используются несколько других методов.

В течение жизни человек страдает от разных видов боли. Один из них – послеоперационная боль. Это боль, которую человек испытывает после небольшой или серьёзной операции. В большинстве случаев эта боль является частью процесса заживления, но иногда она может быть опасной, если возникает по неизвестной причине, которую не видит врач. Иногда эта боль может быть настолько сильной, что может привести к потере жизни.

Избавление от послеоперационной боли способствует более ранней выписке пациентов из больницы, а также уменьшает проявления хронических болевых синдромов. Поскольку боль символизирует наличие повреждения или заболевания в организме, от неё необходимо эффективно избавляться. Основная цель лечения послеоперационной боли – уменьшить или устранить её при минимальных побочных эффектах, насколько это возможно.

Несколько систем поддержки принятия клинических решений созданы для того, чтобы помочь врачам определить уровень и интенсивность послеоперационной боли у пациентов. Некоторые из них могут даже назначать лекарства в зависимости от уровня интенсивности боли. Со временем эти системы постепенно заменяют людей. Основная проблема при назначении лекарств возникает, когда система считывает аномальные или необычные сигналы от пациентов. В этом случае система поддержки принятия решений не может назначить точное лекарство. Поскольку таким пациентам требуется специальное лечение. Чтобы решить эти проблемы и сделать системы поддержки принятия решений полезными для всех, в этой работе представлен подход к выявлению схожих и редких случаев.

ГЛАВА 1. ЦЕЛИ И ВАЖНОСТЬ ИССЛЕДОВАНИЯ, ПОСТАНОВКА ПРОБЛЕМЫ И ЕЕ РЕШЕНИЕ

1.1. Важность исследовательской работы

Исследовательская работа связана с созданием системы поддержки принятия клинических решений (CDSS), которая может работать полностью автоматически или полуавтоматически совместно с врачом для определения пациента, у которого после операции проявляются необычные симптомы. Важно выявить таких пациентов, чтобы назначить им правильное лечение, а также открыть новые горизонты для исследований в области медицины.

CDSS – это инструмент медицинских информационных технологий, который предоставляет врачам, медсестрам и другим медицинским работникам поддержку в принятии клинических решений в режиме реального времени. CDSS может помочь в диагностике, лечении и организации ухода за больными, используя данные о пациентах, научно обоснованные рекомендации и передовой опыт. Было доказано, что CDSS улучшают результаты лечения пациентов за счёт оптимизации клинических процессов и содействия принятию решений, основанных на фактических данных. Они также могут повысить удовлетворённость врачей, предоставляя обратную связь в режиме реального времени и снижая когнитивную нагрузку. Кроме того, CDSS обладает многочисленными преимуществами, в том числе ориентированностью на пациента, сокращением количества медицинских ошибок, улучшением процесса принятия решений, экономией средств, повышенной эффективностью, масштабируемостью, повышением безопасности пациентов, соблюдением руководящих принципов и предписаний, адаптивными подходами, ресурсами оптимизации, совместимостью и обменом данными, сетевым сотрудничеством, глобальным доступом к знаниям и умением прогнозировать ситуации.

1.2. Оценка влияния систем поддержки принятия клинических решений

Текущее состояние оценки воздействия CDSS меняется по мере дальнейшего развития технологий и методологий. Чтобы определить истинную ценность CDSS, крайне важно провести тщательную оценку, которая позволит оценить их влияние на результаты лечения пациентов, процессы оказания медицинской помощи и затраты. Эти оценки должны включать использование соответствующих исследовательских схем и методологий, таких как рандомизированные контролируемые испытания, обсервационные исследования и анализ экономической эффективности. Результаты этих оценок могут быть использованы для принятия обоснованных решений и определения областей для улучшения разработки и внедрения CDSS.

По мере все большего внедрения CDSS в медицинских учреждениях оценка их воздействия становится все более важной для обеспечения положительных результатов и оптимизации производительности системы. Процесс оценки является комплексным и включает в себя множество факторов, таких как клиническая эффективность, удовлетворённость пользователей, рентабельность и интеграция с существующими рабочими процессами.

1.3. Цель исследования

Основная цель данной работы – определить и представить метод или алгоритм, наиболее подходящий для разделения и идентификации как обычных, так и редких случаев послеоперационной боли из элементов базы данных. Было замечено во время исследования, что алгоритмы кластеризации лучше всего работают, когда нужно сгруппировать данные по разным классам на основе общих характеристик.

Алгоритмы кластеризации следуют строгой концепции, согласно которой объекты, схожие по характеристикам, должны принадлежать к одной группе, а объекты с разными характеристиками – к другой.

Объекты определяются по их атрибутам, а различия между объектами измеряются на основе значений атрибутов. Создание такой системы на основе кластеризации, которая может выявлять редкие случаи элементов из базы данных, помогать экспертам и врачам распознавать их и назначать пациентам правильное лечение. Предлагаемая система предоставит экспертам большой объём информации о редких случаях и поможет эффективно принимать решения. В ходе исследования были проанализированы различные методы кластеризации, чтобы определить наилучшие из них для отделения необычных или ненормальных значений элементов из базы данных от обычных.

1.4. Постановка задачи

Системы поддержки принятия клинических решений играют жизненно важную роль в группировании данных в кластеры, когда данные в одном кластере чем-то похожи и отличаются от данных в других кластерах аналогичным образом. Такая группировка данных помогает специалисту проанализировать и идентифицировать редкий случай. Однако существует ряд проблем, связанных с этими системами поддержки принятия решений. Такие, как:

1. Упорядочивание/хранение данных в соответствующем формате или порядке в базе данных.
2. Обеспечение корректировки данных.
3. Анализ данных путём их разделения на различные группы с использованием разных методологий кластеризации.
4. Выбор подходящей методологии является важным компонентом этой работы.
5. Разделение и идентификация редких случаев элементов из базы данных очень важны для правильной диагностики и получения полных данных.

1.5. Решение проблем

Проблемы, рассмотренные ранее, могут быть решены путём разработки соответствующей системы поддержки принятия клинических решений со следующими возможностями:

1. Программное обеспечение должно быть способно к представлению сохранённых данных в надлежащем формате.
2. Представление данных должно быть таким, чтобы похожие данные принадлежали к одной группе.
3. Для сравнения различных методов кластеризации и выбора более подходящего метода или алгоритмов для предлагаемой системы требуется проведение научного исследования.
4. Программное обеспечение должно уметь представлять данные с минимальной сложностью и отделять редкие случаи от распространённых.

Выводы по главе 1

В первой главе рассказывается о важности Использования алгоритма машинного обучения (кластеризации) для анализа клинических данных пациентов с целью выявления пациентов с редкими симптомами, о системах, разработанных в ходе данного исследования, и о том, как это может улучшить процесс лечения. В первой главе объясняются факторы, по которым можно оценивать подобные системы и их эффективность.

В данной главе также описываются шаги, необходимые для любой исследовательской работы, включающие в себя постановку цели и задач, связанных между собой. Из первой главы можно заметить, что для анализа базы данных и выявления полезных результатов из неё, необходимо детально изучить базу данных и проанализировать все её компоненты.

ГЛАВА 2. МЕДИЦИНСКАЯ БАЗА ДАННЫХ, ЕЕ ЭЛЕМЕНТЫ И ХАРАКТЕРИСТИКИ

Основное внимание в этой исследовательской работе уделяется области или теме послеоперационной боли. С этой целью было проведено исследование для получения информации о послеоперационной боли от пациентов со всего мира. Как упоминалось ранее, информация была получена с помощью анкеты, состоящей из нескольких вопросов о личных данных пациентов, истории болезни, принимаемых лекарствах, отношении к боли и многом другом. На основе этой информации можно понять и проанализировать интенсивность послеоперационной боли у разных пациентов. Эта информация, собранная с помощью анкеты, сначала сохраняется в базе данных, содержащей сведения примерно о 3793 людях.

Для анализа наборов данных в этой базе данных использовались различные методы с целью определения наилучшего метода, который может служить целям исследования. База данных о послеоперационной боли состоит из трёх разделов: проблема, решение и выходные данные. Среди них проблема и выходные данные являются основными объектами нашего исследования.

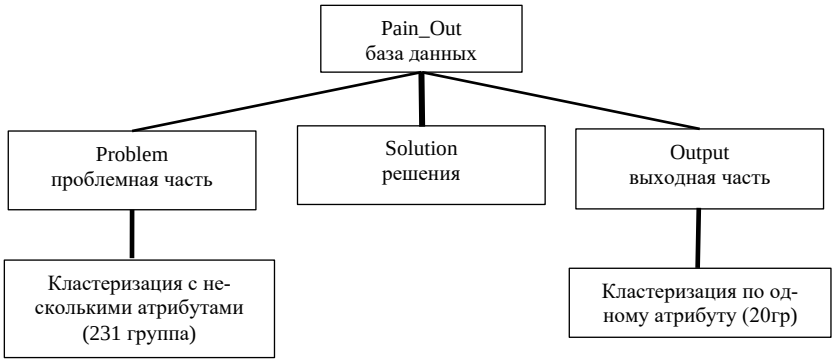

Рисунок 1. Разделение базы данных о послеоперационных болях на различные группы и последующее применение кластеризации на основе характеристик

На рисунке 1 показана древовидная структура, начинающаяся с базы данных Pain Out и спускающаяся вниз к разделению на разделы «Проблема», «Выходная часть» и «Решение». Затем снова вниз к применению методов кластеризации в зависимости от количества атрибутов, рассматриваемых для кластеризации.

2.1. Тип кластеризации

В зависимости от количества атрибутов кластеризация делится на два основных типа или категории.

2.1.1. Одноатрибутная кластеризация

Это тип кластеризации, при котором данные, подлежащие кластеризации, поступают из одного атрибута, а затем для кластеризации этих данных в несколько значимых кластеров используются различные методы кластеризации. С данными, полученными с помощью одного атрибута, легко работать и анализировать их.

2.1.2. Многоатрибутная кластеризация

Это тип кластеризации, при котором данные, подлежащие кластеризации, поступают из различных отдельных атрибутов. Затем для кластеризации данных в значимые кластеры используются различные методы кластеризации. С ним сложнее работать, так как данные генерируются из нескольких атрибутов. Однако он обеспечивает более качественные результаты и содержит гораздо больше информации о данных.

2.2. Атрибуты, учитываемые при решении задачи и выводе

В таблице 1 внизу перечислены атрибуты, которые учитываются при решении проблемной части данной исследовательской работы и выводе.

Таблица 1

Характеристики, выбранные для кластеризации разделов «Проблема» и «Выходная часть» базы данных Pain_Out

PAIN_OUT_QN_ID	**Problem (проблема)**		**Output (выходная часть)**		
	Feature (Особенность)	Description (Описание)	Feature (особенность)	Attributes For Mean (Атрибуты для среднего значения)	
	S1	Inclusion Criteria (Критерии включения)	Mean (среднее значение)	P5.b	Depressed (Pain Affect) (Болевой синдром подавленности)
	S2	Sleep or Seduction Level (Уровень сна или соблазнения)		P5.c	Frightened (Pain Affect) (Болевой синдром испуга)
	D1	Gender (Пол)		P5.d	Helpless(Pain Affect) (Болевой синдром беспомощности)
	D2	Year Of Birth (Год рождения)		P4.c	Falling as sleep (Pain Interference) (Боль, препятствующая засыпанию)
	D3	Weight (Вес)		P10	Pain Treatment Information (Информация о лечении боли)
	D6	Languages (Языки)		P4.b	Out Of Bed Activities (Pain Interference) (Деятельность в период бодрствования)
	D8	Medical History(Comorbidities) (История болезни (Сопутствующие заболевания)		P4.a	In Bed Activities (Pain Interference) Деятельность во время сна)

	D9	Medical History(Existing Conditions) История болезни (существующие заболевания)		P2	Worst Pain Ever (Самая сильная боль за всю историю)
	D10	Chronic Pain (Хроническая боль)		P1	Least Pain Ever (Наименьшая боль за всю историю)
	D11	Opioid Medication (Опиоидные препараты)		P17.a	Severe Persistent Pain (Сильная постоянная боль)
	M1	Sedatives(Pre Medication) (Седативные препараты (перед лечением))		P7	Pain Relief Percentage within 24 hrs (Процент облегчения боли в течение 24 часов)
	M2	Non Opioids(Pre Medication) (Неопиоидные препараты (перед лечением))		P9	Pain Treatment Satisfaction (Удовлетворенность лечением боли)
	M3	Opioids(Pre Medication) (Опиоиды (перед приемом лекарств))			
	Oc	Outcome (Исход)		P6.d	Dizziness level (Side Effect) (Уровень головокружения (побочный эффект))
				P6.b	Drowsiness (Side Effect) (Сонливость (побочный эффект))

				P6.c	Itching (Side Effect) (Зуд (побочный эффект))
				P6.a	Nausea (Side Effect) (Тошнота (побочный эффект))
				P4.d	Staying As sleep (Pain Interference) (Боль, нарушающая сон)

В приведённой выше таблице показаны функции, которые используются для кластеризации проблемных и выходных разделов базы данных pain_out.

Все функции таблицы, включая разделы проблемных и выходных данных, содержат коды, чтобы их было легко идентифицировать в базе данных и реализовать при кодировании. Функция проблемной части включает в себя функции, которые могут иметь различные типы значений, такие как bool (логический) в случае пола, текст в случае языка, числовые значения, такие как год рождения и т. д. Функции проблемной части включают в себя. В то время как функции выходной части имеют числовые значения в основном по шкале от 0 до 10.

В ходе исследования решение сохраняется в исходном виде, поскольку оно представляет собой отображение базы данных. В приведённой выше таблице показано, что для кластеризации выходных данных используется одна функция, а для кластеризации проблемной части базы данных – несколько функций.

2.3. Разделы базы данных о послеоперационных болях пациентов

2.3.1. Проблемная часть

Эта часть базы данных о степени болезненности содержит полную информацию о пациентах, начиная с возраста, пола, веса, языка и заканчивая результатами различных анализов, проведённых перед

операцией. Изначально атрибуты, относящиеся к проблеме, необходимо обработать перед применением к ним методов кластеризации. Обработка атрибутов требуется, потому что:

1. Данные не были должным образом организованы.
2. Некоторые пациенты, вероятно, недостаточно хорошо понимали вопросник при его заполнении.
3. Некоторая информация неоднозначна.
4. Некоторые данные в атрибутах выходят за рамки кластера.
5. Довольно много данных либо отсутствуют, либо заполнены неправильно.
6. Часто встречаются орфографические ошибки.

2.3.2. Атрибуты, рассматриваемые в связи с проблемой

В ходе изучения различных алгоритмов было замечено, что некоторые алгоритмы более подходят и эффективны для одной конкретной области, чем для других. Для кластеризации проблемной части базы данных Pain_out были рассмотрены следующие атрибуты.

Таблица 2

Особенности проблемного раздела базы данных Pain_out

Feature No.	**Name of Attribute (Имя атрибута)**	**Values Range (Диапазон значений)**
S1	S1_ASSENT	Patient assent/consent takes values: (Согласие пациента имеет значения:) Y = 1 N = 0
	S1_AVAIL	Patient Available takes values: (Наличие пациента имеет значения:) Y = 1 N = 0
	S1_CNSNT_AGE	Patient consenting age or over takes values: (Возраст пациента согласия, имеет значения:) Y = 1 N = 0
	S1_FILL_UNAID	Patient able to fill in questionnaire unaided takes values: (Пациент, способный самостоятельно заполнить анкету, имеет значения:) Y = 1 N = 0

	S1_FILL_UNAID_COG	Cognitively Impaired takes values: (Когнитивные нарушения имеет значения:) Y = 1 N = 0
	S1_FILL_UNAID_ILL	Too ill takes values: (Слишком болен имеет значения:) Y = 1 N = 0
	S1_FILL_UN-AID_OTHR	Other reason takes values: (По другой причине имеет значения:) Y = 1 N = 0
	S1_FILL_UNAID_RW	Cannot read/write takes values: (Не умеет читать/писать имеет значения:) Y = 1 N = 0
	S1_NSAMESURGERY	Same organ surgery takes values: (Операция на одном и том же органе имеет значения:) Y = 1 N = 0
	S1_POD1_6HR	POD1 and 6+ takes values: (Группы 1 и 6+ имеет значения) Y = 1 N = 0
S2	S2_SLP_SED	S2 Sleep or sedation level takes values: (S2 Уровень сна или седативного эффекта имеет значения:) Awake and fully conscious = 0 Normal sleep = 1 Mildly sedated = 2 Moderately sedated = 3 Severely sedated = 4
D1	D1_GENDER	Gender takes values: (имеет значения пола:) Female (Женщина) = 0 Male (Мужчина)= 1
D2	D2_BRTHYR	D2 Year of birth of patient (Год рождения пациента)
D3	D3_WGHT	D3 Weight of patient (Вес пациента)
D6	D6_PRT2_LANG	D6 Language of Part 2 (Язык части 2) (Patient questionnaire, Outcome) ((Анкета пациента, результаты обследования)) takes values: 1 = Arabic 2 = Bahasa Malaysia 3 = Bengali 4 = English 5 = French 6 = German 7 = Hebrew 8 = Italian 9 = Korean 10 = Mandarin 11 = Romanian 12 = Russian 13 = Spanish 14 = Swedish 15 = Turkish
D8	D8_Comorbidities	Medical History D8 takes values: (История болезни) No: 0 Yes: 1

D9	D9_LACTATION	D9 Lactation takes values: (Период лактации имеет значения:) No: 0 Yes: 1
	D9_PREGNANCY	D9 Pregnancy takes values: (Беременность имеет значения:) No: 0 Yes: 1
	D9_PREGNANCY_WK	D9 Pregnancy Week (Неделя беременности)
D9	D10_CHRONIC	D10 Chronic Pain Condition takes values: Not possible to obtain information: (Состояние хронической боли имеет значения:) (Невозможно получить информацию:) -1 No: 0 Yes: 1
D11	BA_OPIOID	D11 Opioids takes values: (Опиоиды D11 имеет значения:) Not possible to obtain information (Невозможно получить информацию) = -1 N = 0 Y = 1
M1	PRE_SED	M1 Sedatives (pre-medication) takes values: (Седативные препараты M1 (премедикация) имеет значения:) Not possible to obtain information (Невозможно получить информацию) = -1 N = 0 Y = 1
M2	PRE_NOPIO	M2 Non-opioids (pre-medication) takes values: (M2 неопиоидных препаратов (премедикация) имеет значения:) Not possible to obtain information (Невозможно получить информацию) = -1 N = 0 Y = 1
M3	PRE_OPIO	M3 Opioids (pre-medication) takes values: (M3 опиоидных препаратов (премедикация) имеет значения:) Not possible to obtain information (Невозможно получить информацию) = -1 N = 0 Y = 1
Oc	Outcome (Исход)	Mean of values (Среднее значение)

В вышеупомянутой таблице приведены сведения об этих атрибутах, включая их номер в анкете, название поля в базе данных pain_out и диапазон их значений. Значение каждого атрибута обрабатывается отдельно перед использованием для вычисления среднего

значения. Некоторые из этих атрибутов требуют специальной обработки, чтобы они не влияли на результаты кластеризации. Например.

Таблица 3

Характеристика, которая потребовала дополнительной обработки

Feature No.	**Name of Attribute (Имя атрибута)**	**Values Range (Диапазон значений)**	**Processed Range (Обрабатываемый диапазон)**
D2	D2_BRTHYR	D2 Year of birth of patient (Год рождения пациента)	Age Calculation (Расчет возраста) =(2005-D2_BRTHYR)
D6	D6_PRT2_LANG	D6 Language of Part 2 (Язык части 2) (Patient questionnaire, Outcome) (Анкета пациента, результаты обследования) 1 = Arabic 2 = Bahasa Malaysia 3 = Bengali 4 = English 5 = French 6 = German 7 = Hebrew 8 = Italian 9 = Korean 10 = Mandarin 11 = Romanian 12 = Russian 13 = Spanish 14 = Swedish 15 = Turkish	For Each language values are 1 or 0. More than one language then Number of languages will be summed. (Для каждого языка значения равны 1 или 0. Если используется более одного языка, количество языков будет суммировано..)

После предварительной обработки атрибутов базы данных к ним применяется алгоритм кластеризации. Это распространённый сценарий, при котором необычные значения одного атрибута могут нарушить работу и повлиять на общие результаты эксперимента. Чтобы этого избежать, перед применением какого-либо алгоритма или метода к набору данных требуется тщательный анализ всех значений и атрибутов.

Это можно хорошо проиллюстрировать с помощью таблицы 4. Значения атрибутов после обработки в соответствии с кодом пациента и pain_out_qn_id указаны ниже. С помощью этих значений вычисляется и оценивается проблемная часть базы данных.

Таблица 4

Пример из раздела «Проблемы»

Patient Code	PAIN_OUT_QN_ID	S1	S2	D1	D2	D3	D8	D9	D10	D11	M1	M2	M3	Oc
RLH18	607	1	1	1	54	-1	-1	0	0	0	0	0	0	5.73
D8HQCEKVJC	626	1	1	1	44	71	-1	0	0	0	-1	0	0	4.3
RLH71	1165	1	1	0	75	70	1	0	0	0	-1	0	0	2.7

2.4. Часть выходных данных

Это разделение баз данных о боли содержит информацию о самочувствии после операции или о симптомах послеоперационной боли и её уровне у разных пациентов. Первоначально атрибуты, относящиеся к выходным данным, должны быть обработаны перед применением к ним методик кластеризации, поскольку:

1. Данные не упорядочены должным образом.
2. Некоторые пациенты, вероятно, были недостаточно подготовлены к тому, чтобы полностью понять вопросник, отвечая на него.
3. Некоторая информация неоднозначна.
4. Некоторые данные в атрибутах выходят за рамки кластеризации.

2.5. Атрибуты, учитываемые при выводе

В приведённой ниже таблице 5 рассматриваются функции и атрибуты, используемые для кластеризации в разделе вывода базы данных pain_out.

Таблица 5

Выходные атрибуты с диапазоном значений

Mean (Средний значений)		
Feature No.	**Name of Attribute (Имя атрибута)**	**Values Range (Диапазон значений)**
P5.b	O_ALWPART	-1 OR 1 to 10 scale
P5.c	O_FEEL_ANX	-1 OR 1 to 10 scale
P5.d	O_FEEL_DEPRS	-1 OR 1 to 10 scale
P4.c	O_FEEL_FRGHT	-1 OR 1 to 10 scale
P10	O_FEEL_HLPLSS	-1 OR 1 to 10 scale
P4.b	O_FLLSLEEP	-1 OR 1 to 10 scale
P4.a	O_INF_HLP	-1 OR 1 to 10 scale
P2	O_INTRFR_INBED	-1 OR 1 to 10 scale
P1	O_INTRFR_OUTBED	-1 OR 1 to 10 scale
P17.a	O_MAXPAIN	-1 OR 1 to 10 scale
P7	O_MINPAIN	-1 OR 1 to 10 scale
P9	O_PERS_INTNSTY	-1 OR 1 to 10 scale
P6.d	O_RELIEF	-1 OR 0,1 to 1,0 scale
P6.b	O_SATISF	-1 OR 1 to 10 scale
P6.c	O_SIDE_DIZZ	-1 OR 1 to 10 scale
P6.a	O_SIDE_DROWS	-1 OR 1 to 10 scale

P4.d	O_SIDE_ITCH	-1 OR 1 to 10 scale
P5.b	O_SIDE_NAUSEA	-1 OR 1 to 10 scale
P5.c	O_STYSLEEP	-1 OR 1 to 10 scale

Для вычисления среднего значения используются несколько выходных атрибутов. В вышеупомянутой таблице приведены сведения об этих атрибутах, включая их номер в анкете, название поля в базе данных pain_out и диапазон их значений. Значение каждого атрибута обрабатывается отдельно перед использованием для вычисления среднего значения. Обработка значений и их представление в рассматриваемом вопросе могут быть подробно описаны с помощью приведённых ниже примеров.

Таблица 6

Примеры раздела вывода базы данных Pain_Out

Patient Code	**PAIN_OUT_QN_ID**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**	**P5.b**
RLH 17	606	9	10	6	9	10	9		10	10	9	7	8	0,9	6	9	5	4	7	10
RLH 20	609	8	8	9	10	8	5	8	5	6	10	8	8	0,6	8	6	9	10	7	7
RLH 11	601	1	5	8	6	9	4	9	3	9	8	4		0,4	8	4	2	1	2	6

В вышеупомянутой таблице показаны код пациента, идентификатор пациента и ответы пациентов на различные вопросы анкеты. Как упоминалось выше, значение каждого атрибута находится в пределах указанного диапазона. Отсутствующие значения указывают на то, что пациент либо пропустил, либо не ответил на этот конкретный вопрос.

Поэтому при вычислении среднего значения мы можем исключить эти пустые значения из наших расчётов. Среднее значение вышеупомянутых случаев после обработки выглядит следующим образом:

Таблица 7

Средние значения случаев из Таблицы 6 после расчётов

Patient Code (Код пациента)	**PAIN_OUT_QN_ID**	**Mean (средний)**
RLH17	606	7.666666667
RLH20	609	7.368421053
RLH11	601	4.944444444

После обработки всех атрибутов случаев из Таблицы 6, значения суммируются для получения среднего значения для каждого случая. Затем это среднее значение используется в качестве источника входных данных для кластеризации данных с помощью нескольких методов кластеризации. Чтобы понять процесс усреднения, в таблице 7 выше приведены 3 строки из базы данных Pain-out. Среднее значение используется в качестве выходного параметра, потому что значения всех выходных параметров находятся в диапазоне от 0 до 10.

Выводы по главе 2

Во второй главе анализируется база данных, которая используется в данной исследовательской работе. Важно понимать атрибуты, которые используются для извлечения полезной информации из базы данных. Как упоминалось ранее, база данных, используемая в рамках этой исследовательской работы, относится к медицине и включает в себя различные атрибуты, относящиеся к пациентам.

Необходимо изучить и понять структуру базы данных, поскольку она играет решающую роль в разработке методологии её анализа. Правильное проектирование базы данных способствует оптимизации процессов управления данными и повышению общей эффективности.

В хорошо спроектированных базах данных используются такие методы, как проверка данных и ограничения, для обеспечения соблюдения правил и точности хранимой информации. Сводя к минимуму несоответствия и ошибки, базы данных могут предоставлять точную информацию о данных, что приводит к улучшению процесса принятия решений. Своевременная и точная информация имеет решающее значение для лиц, принимающих решения. Правильная структура базы данных облегчает доступ к соответствующей информации на протяжении всего процесса её анализа.

ГЛАВА 3. РАБОТЫ, СВЯЗАННЫЕ С ВЫЯВЛЕНИЕМ РЕДКИХ ИЛИ НЕОБЫЧНЫХ ЭЛЕМЕНТОВ ДАННЫХ ИЗ БАЗЫ ДАННЫХ

В этом разделе представлены работы, которые были выполнены в этой области. Он включает в себя методологии, которые используются для выявления редких случаев и основаны на подробном изучении различных методологий; для этой работы выбрана наиболее подходящая модель для наших данных.

3.1. Выявление исключительных или редких случаев послеоперационных болей из базы данных

Выявить исключительные или редкие случаи послеоперационной боли можно с помощью наблюдения, которое сильно отличается от других значений в случайно сгенерированной выборке из генеральной совокупности, что кажется, будто оно было получено с помощью другого механизма. Тем не менее, точное определение исключительного или редкого случая в основном зависит от скрытой структуры данных и метода выявления таких случаев или редких случаев. Барнет и Льюис (Barnett and Lewis, 1994) указывают, что выбросом, или аномалией, называется наблюдение, которое заметно отличается от других наблюдений в выборке, в которой оно встречается [01]. Джонсон (Johnson, 1992) определяет выброс как наблюдение в наборе данных, которое, по-видимому, не согласуется с остальной частью этого набора данных [02].

При анализе набора данных становится очень интересно наблюдать за некоторыми значениями, которые настолько сильно отличаются от остальных, что их результаты становится подозрительным. Эти значения обычно называются редкими случаями, поскольку они обычно находятся вне набора обычных наблюдений. В процентном отношении они значительно меньше, чем в обычных случаях. Для получения более качественного набора данных и их более эффектив-

ного анализа необходимо отделять те отклоняющиеся данные, которые находятся очень далеко от остальных. Как правило, изучение этих случаев даёт очень интересные результаты.

Прежде чем рассматривать значения таких данных как редкие случаи, необходимо подумать о некоторых проблемах: являются ли эти значения результатом человеческой ошибки или сбоя оборудования, или же проблема заключается в алгоритме или механизме, используемом для анализа данных. Эти вопросы имеют огромную важность, поскольку потеря хорошего значения из-за сбоя оборудования или человеческой ошибки может сильно повлиять на результаты.

3.2. Применение обнаружения редких случаев в различных областях науки

Обнаружение редких случаев имеет важное значение в различных сферах, начиная от выявления мошенничества и заканчивая очисткой данных. Аналогичным образом, от обнаружения проблем в сети до клинической диагностики заболеваний. Несколько других областей применения: обнаружение вторжений, мониторинг работоспособности системы, прогнозирование неблагоприятных погодных условий, ГИС (географическая информационная система). В медицине ценятся редкие случаи, такие как описаны Ахмедом М.Ю. [3].

3.3. Методы обнаружения редких случаев

Методы обнаружения редких случаев можно разделить на классические и современные. Классические методы можно назвать одномерными, а современные – многомерными. Другим способом их классификации могут быть параметрические (статистические) методы и непараметрические методы.

Статистические методы, как правило, основаны на статистических оценках неизвестных параметров распределения (Roiz & Caussinus, 1990) [7] или предполагают известное базовое распределение наблюдений (Barnett and Lewis, 1994, Rousseuw, 1987 [8]). В противном

случае они основаны на некоторых статистических оценках неизвестного распределения. С помощью этих методов отклонения от нормального распределения можно выявить редкие случаи. Однако эти методы не очень хорошо подходят для наборов данных с высокой размерностью, если неизвестно распределение исходных данных.

В то же время непараметрические методы выявления редких случаев гораздо лучше подходят для анализа данных в больших базах. Эти методы, как правило, основаны на измерении локального расстояния (Мушель и Шонлау, 1998; Джин и др., 2001 [9]; Хокинс и др., 2002). Другой способ поиска редких случаев основан на методах кластеризации, где кластер меньшего размера считается редким случаем (Нг и Хан, 1994; Барбара и Чен, 2000 [10]; Шекхар и Лу, 2002 [11]).

Аналогичным образом в 2003 году Ху и Санг предложили метод для различения кластеров с высокой и низкой плотностью. Он разделяет данные на два набора, которые можно назвать редкими случаями и обычными [12].

3.4. Одномерные статистические методы

В начале работы с одномерными методами предполагалось, что в основе лежит идентичное и независимое известное распределение данных. Ожидалось, что будут известны даже редкие случаи и параметры распределения.

Одной из частей статистических методов являются предположения, и одним из основных предположений является создание модели, которая позволяет случайным образом отбирать некоторые наблюдения из распределений, отличающихся от целевого или нормального распределения $N(\mu,\sigma^2)$. Затем обнаружение редких случаев преобразуется в идентификацию наблюдений, которые находятся в области редких случаев. Для коэффициента достоверности α, 0 < α <1, область редких случаев нормального распределения составляет

$$\boldsymbol{out}(\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma}^{\mathbf{2}}) = \{\boldsymbol{x} : \ |\boldsymbol{x} - \boldsymbol{\mu}|\} > \boldsymbol{z}_{\mathbf{1}-\boldsymbol{\alpha}/\mathbf{2}^{\boldsymbol{\sigma}}}$$

где z_p – p-квантиль нормального распределения N (0, 1). Число x будет считаться α-редким случаем по отношению к F, если

x ϵ $out(\alpha, \mu, \sigma^2)$. Проблема этого традиционного подхода заключалась в том, что вместо того, чтобы выявлять затронутые наблюдения, он указывал на те наблюдения, которые находились в области редких случаев.

Позже, в 1993 году (Гэзер и Дэвис) предложили два разных способа устранения редких случаев [13]. Один из них состоит из одноэтапного исключения всех значений сразу, а другой может быть последовательным тестированием одного наблюдения для выявления и удаления редких случаев за раз. Правило, которое они разработали для выявления редких случаев за один шаг,

$$Out\ (\alpha_n, \mu_n, \sigma_n^2) = \{x: |x - \mu_n| > g(n, \alpha_n)\ \sigma_n\}$$

где *'n'* обозначает размер выборки, *'μ_n'* и *'σ_n'* – оценочное среднее значение и стандартное отклонение выборки; α_n обозначает коэффициент достоверности после поправки на множественные тесты, а $g(n, \alpha_n)$ – границы областей редких случаев. Обычно *'μ_n'* и *'σ_n'* оцениваются по соответствующему среднему значению $\overline{\overline{Xn}}$ и стандартному отклонению S_n.

В то время как для одношаговой процедуры лучше всего подходит поправка Бонферрони для одного сравнения. Он устанавливает значение α для всего набора из n сравнений равным α, учитывая, что значение α для отдельного сравнения равно α/n. Другой популярный метод использует простую поправку:

$$\alpha_n = 1-(1-\alpha)^{1/n}$$

Последовательную процедуру можно разделить на внутреннюю, также называемую процедурой прямого отбора, и процедуру обратного отбора. При прямом отборе на каждом этапе проверяется, является ли наиболее экстремальное наблюдение выбросом, и если оно является выбросом, то удаляется из набора данных, а процедура продолжается с другими наблюдениями.

При обратном отборе большая выборка сначала сокращается до меньшей, а наблюдения, удалённые из выборки, сохраняются отдельно. Затем сокращённая выборка оценивается на основе статистики, а затем снова проверяется удалённая выборка наблюдений, чтобы определить, являются ли они редкими случаями. Если наблюдение не является редким случаем, оно должно быть удалено из списка редких и снова добавлено в сокращённую выборку. Затем оценивается следующее наблюдение, и процедура продолжается до следующих наблюдений.

Концепция точки разрыва была введена в 1971 году Хэмпелом [14]. По его словам, «точка разрыва – это мера устойчивости оценки к редким случаям». Это наименьший процент редких случаев в генеральной совокупности, который приводит к тому, что генеральная совокупность принимает произвольно большие значения. Это означает, что оценка является более надёжной, если у неё более высокий порог сбоя.

В 1977 году Тьюки представил диаграмму размаха, на которой можно отобразить редкие случаи [15]. Диаграмма размаха основана на квадрантах распределения, где среднее значение $\mu_n = (Q1+Q3)/2$, а стандартное отклонение $\sigma_n = Q3-Q1$.

3.5. Многовариантное обнаружение редких случаев

При многовариантных наблюдениях рассмотрение одной переменной по отдельности затрудняет обнаружение редких случаев. В этом случае единственный способ выявить редкий случай – провести многовариантный анализ. Простой график с двумя измерениями в двумерном пространстве по осям ‘*x*’ и ‘*y*’ представляет собой многовариантный анализ редких случаев.

Акуна и Родригес в 2004 году представили концепцию маскирования и замены в наборах данных с несколькими кластерами редких случаев [16].

Эффект маскировки: один редкий случай маскирует другой редкий случай, если таковой можно считать независимо от первого,

но не в присутствии первого редкого случая. Только после удаления первого редкого случая другой редкий случай становится таковым. Основная причина маскировки заключается в том, что кластер, состоящий из редких случаев, смещает ковариацию и среднее значение в свою сторону, и в результате отклонение редкого случая от среднего значения становится небольшим.

3.5.1. Эффект размытия

Эффект размытия противоположен эффекту маскировки. Здесь один редкий случай перекрывает другой, только в том случае, если другой редкий случай может рассматриваться как редкий случай под влиянием первого редкого случая. При удалении первого редкого случая второе наблюдение становится наблюдением, не являющимся редким случаем. Здесь смещение происходит только тогда, когда группа удалённых экземпляров искажает ковариацию и средние значения в её сторону и в сторону от внешних наблюдений, и в результате расстояние от экземпляров до среднего значения велико. В результате они выглядят как редкие случаи.

3.6. Статистический метод обнаружения редких случаев

Статистический метод – это один из методов многомерного обнаружения редких случаев, основанный на оценке параметров распределения и методах интеллектуального анализа данных без использования параметров. Обычно он выявляет те наблюдения, которые находятся очень далеко от центра распределения данных. Существует несколько мер расстояния, которые можно использовать для вычисления расстояния. Одна из них – расстояние Махаланобиса, которое определяется следующим образом:

$$V_n = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - x_n)(x_i - x_n)^T$$

Где n – количество наблюдений из p-мерного набора данных, – среднее значение вектора, а V_n – ковариационная матрица. Расстояние

Махаланобиса для каждого многомерного наблюдения, обозначенного *Mi*, определяется как:

$$M_i = \left(\sum_{i=1}^{n} (x_i - x_n)^T V_n^{-1} (x_i - x_n) \right)^{1/2}$$

Значения данных с большим расстоянием Махаланобиса обычно исключаются как редкие случаи. Маскировка и сглаживание играют важную роль в расчёте расстояния Махаланобиса. Маскировка может уменьшить расстояние Махаланобиса между обычными случаями, а сглаживание иногда может увеличить расстояние между редкими случаями и не редкими.

Дисперсия и среднее значение распределения – два наиболее известных метода, используемых в статистике для выявления редких случаев в одномерных процедурах. Хади в 1992 году первым представил концепцию устойчивых оценок для параметров многомерного распределения [17]. Он заменяет средние значения вектора вектором, содержащим медиану переменных, и подчёркивает важность расчёта ковариационной матрицы только для тех подмножеств, которые имеют наименьшее расстояние Махаланобиса.

3.7. Непараметрические методы обнаружения редких случаев

Непараметрические методы используются для обнаружения редких случаев в больших базах данных. Непараметрический метод не предполагает наличия базовой модели для генерации данных. Среди них наиболее распространёнными являются метод, основанный на расстоянии, метод, основанный на плотности, и метод кластеризации.

3.7.1. Методы, основанные на расстоянии

Кнорр и Нг в 1997 году представили методы, основанные на расстоянии, которые базируются на глобальном критерии, определяемом параметрами β и r [18]. Наблюдение в генеральной совокупности считается выбросом, если существует доля β наблюдений в генеральной

совокупности, которые находятся на расстоянии *r* от него. Однако в этом определении есть несколько проблем, которые значительно усложняют процедуру, например, определение правильного значения r, временная сложность и т. д.

В 2000 году Рамасвами и др. представили другое определение для выявления редких случаев на основе расстояния [19]. Для этого требуются два целых числа *V* и *I*. Редкими случаями считаются первые *I* наблюдений, уже отсортированных по наибольшему расстоянию до их *V*-го ближайшего соседа.

Это относительно непараметрический подход, который, как было показано, хорошо масштабируется для больших наборов данных умеренной или высокой размерности [20]. Его базовый алгоритм состоит из вложенного цикла, который вычисляет расстояние между каждой парой объектов, а затем определяет объекты, находящиеся на большом расстоянии от большинства других объектов, как редкие случаи. Этот базовый алгоритм имеет квадратичную сложность по отношению к общему количеству объектов, что делает его очень нестабильным. Поэтому в дальнейшем основное внимание в алгоритме на основе расстояния будет уделяться выявлению субквадратичных алгоритмов.

3.7.2. Подход, основанный на плотности данных

Подход, основанный на плотности данных рассматривает совокупность как кластеры небольшого размера, а редкие случаи – как один из таких кластеров. Примерами таких методов являются кластеризация больших приложений (CLARA) и разбиение вокруг медиан (PAM). Он может работать более эффективно в случае сочетания нескольких методик для кластеризации данных.

В рамках этого метода каждому объекту, принадлежащему к группе, присваивается степень выделения. Эта степень также называется коэффициентом локального редкого случая (*LOF*). Она называется локальной, потому что здесь степень того, что объект является редким случаем, зависит от его окружения, чтобы показать, насколько далеко объект находится от своих соседей [21].

Здесь редкие случаи зависят от плотности их соседей. Кроме того, экземпляр не классифицируется как редкий случай или нередкий случай; вместо этого для каждого экземпляра вычисляется локальный коэффициент редкого случая (*LOF*), который показывает, насколько сильно экземпляр можно считать редким [22].

Этот алгоритм был впервые представлен Маркусом М. Брейнигом, Хансом-Петером Кригелем, Раймондом Т. Нгом и Йоргом Сандером. Здесь они использовали некоторые концепции других методов кластеризации, таких как «достижимость или базовое расстояние», которые изначально были разработаны в *DBSCAN* и *OPTICS*. Основная концепция этого метода обнаружения редких случаев заключается в определении местоположения объектов на основе их k ближайших соседей.

Эти соседи определяют плотность вокруг объекта. Таким образом, измеряя и сравнивая локальную плотность объекта с плотностью его соседей, можно легко определить те объекты, плотность которых ниже, чем у их соседей, как редкие случаи. Существует несколько определений, используемых для алгоритмов обнаружения редких случаев на основе плотности. Где: экземпляр x в наборе данных D является редким случаем с параметрами p и λ, если по крайней мере часть объектов p находится на расстоянии большем, чем λ, от x [23] [24]. У него есть несколько проблем, таких как определение значения λ, и у него отсутствует возможность ранжировать выброс. Таким образом, объект с небольшим количеством соседей на расстоянии λ может рассматриваться как такой же редкий случай, как и объект на том же расстоянии с большим количеством соседей.

Другой вариант – для работы с целым числом k, где $k < n$, редкие случаи – это первые n экземпляров с наибольшим расстоянием до их k-го ближайшего соседа [25]. Самая большая проблема этого определения заключается в том, что оно учитывает только соседей на *k-м* расстоянии и игнорирует информацию о более близких точках.

Допустим, у нас есть объект A. Расстояние от объекта до его k ближайших соседей обозначается как k-расстояние *(A)*. Все объекты

на этом расстоянии входят в набор k ближайших соседей. Набор ближайших соседей можно обозначить как $N_k(A)$. Это расстояние можно назвать расстоянием достижимости, которое выражается как:

$$Reachability\_Distance_k(A, C) = max\{k\text{-}distance(C), d(A,C)\}$$

Таким образом, расстояние достижимости от *A* до *C* – это фактическое расстояние между ними или k-расстояние от C. Объект, принадлежащий к k ближайшим соседям B, считается находящимся на равном расстоянии [26]. Таким образом, локальная достижимость объекта – это:

$$LRD(A) = 1 \Big/ \left(\frac{\sum_{B \epsilon\, N_k(A)} Reachability - Distance_k(A,B)}{|N_k(A)|} \right)$$

Это среднее расстояние, на которое объект *A* может быть досягаем для своих соседей. Следующий этап – сравнение локальных плотностей досягаемости соседей для определения локального коэффициента редких случаев, который равен:

$$LOF_K(A) = \frac{\sum_{B \epsilon\, N_k(A)} \frac{LRD_K(B)}{LRD_K(A)}}{|N_k(A)|}$$

Для определения локального коэффициента редких случаев (*LOF*) объекта, необходимо измерить две величины. Одна из них – это среднее значение локальной плотности достижимости соседей, делённое на собственную локальную плотность достижимости объекта. Если значение локального коэффициента редких случаев *(LOF*) примерно равно 1, то это означает, что объект имеет сопоставимую с соседями плотность. Таким образом, он сам не является выбросом. *LOF* считается очень эффективным методом выявления редких случаев, которые в противном случае было бы трудно обнаружить. Например, объект, находящийся на очень небольшом расстоянии от группы кластеров, может быть редким случаем, в то время как объект в разреженном кластере может иметь свойства, схожие со свойствами его соседей. Таким образом, он не является редким случаем. Однако

возникает проблема с определением значения, которое следует считать редким случаем. Должно ли оно быть равно 1,1 или больше 2? Было замечено, что это зависит от набора данных. В одном наборе данных значение 1,1 может быть редким случаем, в то время как в другом даже объект с *LOF 2* не может быть редким случаем.

Выводы по главе 3

Из приведённого выше исследования следует, что кластеризация – это метод, который используется не только для группировки данных, но и для выявления редких случаев в данных. Таким образом, в конце этого исследования мы пришли к выводу, что кластеризацию можно использовать для группировки данных, а также для выявления редких случаев в наших данных. Методы кластеризации наиболее широко используются, и среди них выбрали четыре основных метода кластеризации. Сначала было проведено исследование литературы по четырём наиболее часто используемым методам кластеризации. Затем рассмотрели и сравнили их, чтобы увидеть их применение, преимущества и недостатки.

ГЛАВА 4. КЛАСТЕРИЗАЦИЯ, ЕЕ ТИПЫ И ВАЖНОСТЬ ПРИ АНАЛИЗЕ ДАННЫХ

Кластеризация – это метод, с помощью которого набор различных объектов делится на подмножества, в которых объекты в одном подмножестве в каком-то смысле похожи друг на друга и отличаются от объектов в другом подмножестве в том же смысле. Это метод обучения без учителя, который означает, что нужно понять, как организованы данные, а представленные данные всегда содержат примеры без пометок. Кластеризация в основном работает со статистическими данными и включает в себя такие методы, как интеллектуальный анализ данных, анализ изображений, машинное обучение и многое другое.

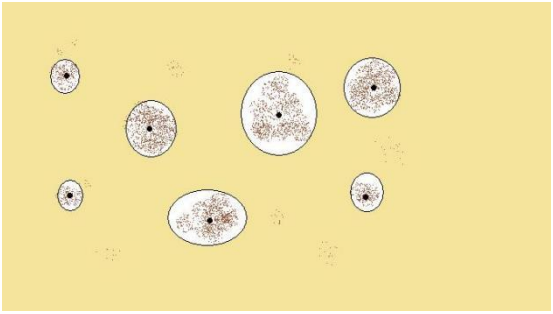

Рисунок 2. Группировка данных в различные кластеры

4.1. Разделение данных на кластеры

Разделение данных на кластеры – первый шаг в анализе данных. В кластеризации, необходимо разделить данные на соответствующее количество кластеров. Количество кластеров может быть задано пользователем или получено с помощью критерия достоверности кластеров.

4.2. Задача кластеризации

В целом кластеризация данных или объектов делится на три основные подзадачи. Первая из них – выбор функции оценки для алгоритма кластеризации. Второстепенным является принятие решения о подходящем количестве кластеров, а третьим может быть выбор наиболее подходящего алгоритма кластеризации.

4.2.1. Функция оценки

Функция оценки также называется целевой функцией. Выбор этой целевой функции во многом зависит от области применения. Среди всех целевых функций наиболее часто используемой является:

$$f(P,C)=\sum d\,(x_i,\, c_{pi})^2$$

где

P = раздел;

C = кластер;

d = функция расстояния.

Наиболее часто используемыми функциями расстояния в кластеризации являются евклидово расстояние и манхэттенское расстояние. Они определяются следующим образом:

Euclidean Distance

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (x_1^i - x_2^i)^2}$$

Manhattan Distance

$$d(x_1, x_2) = \sum_{i=1}^{n} \left\| x_1^i - x_2^i \right\|$$

4.2.2. Количество кластеров

Основная проблема, связанная с выбором подходящего количества кластеров, заключается в том, что в большинстве случаев знания предметной области не очень очевидны, а данные имеют больше измерений. Это влияет на результаты кластеризации и качество кластеров. Если количество кластеров слишком мало, это может привести к недостаточной кластеризации, то есть будет очень сложно правильно разделить объекты в данных.

Аналогично, если количество кластеров слишком велико, это может привести к разделению относительных областей на несколько более мелких областей. Эта проблема называется проблемой валидации кластеров. Для решения этих проблем используются несколько подходов, таких как внешний подход, внутренний подход и относительный подход. Среди них наиболее широко используется относительный подход, при котором структура кластеризации оценивается путём создания различных кластеров с разным количеством элементов, а затем их сравнения по определённому критерию оценки.

4.2.3. Верная кластеризация

Верная кластеризация – это кластеризация, которая позволяет получить кластеры хорошего качества с высоким внутриклассовым сходством и низким межклассовым сходством. Её качество также может зависеть от меры сходства, используемой в рамках метода и во время его реализации. Верная кластеризация достаточно хороша даже для кластеризации скрытых шаблонов данных. На это может повлиять определение, выбранное для представления кластеров.

4.3. Требования к кластеризации

Кластеризация – это очень важный процесс, особенно при интеллектуальном анализе данных, и она должна соответствовать следующим требованиям.

1. Она не должна зависеть от порядка или способа ввода данных в систему.

2. Она должна быть достаточно гибкой, чтобы работать с различными типами атрибутов.

3. Она может работать с кластерами разных форм.

4. Она должна быть способна работать с шумом и редкими случаями.

5. Она должна быть удобной в использовании и интерпретируемой.

6. Она должна быть способна работать с многомерными данными.

4.4. Примеры кластеризации

Мы можем найти различные примеры кластеризации в нашей повседневной жизни. Вот некоторые из них.

1. Кластеризация помогает в разработке маркетинговых стратегий. С помощью кластеризации маркетологи могут легко идентифицировать различные группы в своей клиентской базе и соответствующим образом формулировать для клиентов политику.

2. В страховых агентствах кластеризация используется для выявления держателей страховых полисов с высокой и низкой средней стоимостью страховых случаев.

3. Кластеризация помогает наблюдать и классифицировать различные линии разломов, вызывающие землетрясения, по всему миру.

4. Для городского планирования полезно выделять различные группы домов по типу, конструкции и расположению.

4.5. Основные методы кластеризации

Существует пять основных категорий методов кластеризации.

1. Алгоритм разбиения: здесь множество разбивается на различные подмножества, а затем оценивается по какому-либо критерию.

2. Иерархический алгоритм: критерий используется для создания иерархической декомпозиции множества данных. Он может быть объединяющим или разделяющим.

3. Основанный на плотности: он основан на функции плотности и связности, существующей между различными данными в наборе.

4. Основанный на сетке: состоит из функции детализации на нескольких уровнях.

5. На основе модели: формулируются различные модели кластеров, а затем определяется наиболее подходящая модель.

4.5.1. Иерархическая кластеризация

Как следует из названия, иерархическая кластеризация – это метод, при котором кластеризация данных выполняется в определённом иерархическом порядке, и эта иерархия создаётся в результате итеративного процесса. Эта иерархия кластеризации, при которой кластеризация выполняется одна за другой, воспринимается как древовидная структура, также известная как дендрограмма.

Иерархическая кластеризация далее подразделяется на две категории: восходящая, также известная как агломеративная, и нисходящая, также известная как дивизимная кластеризация.

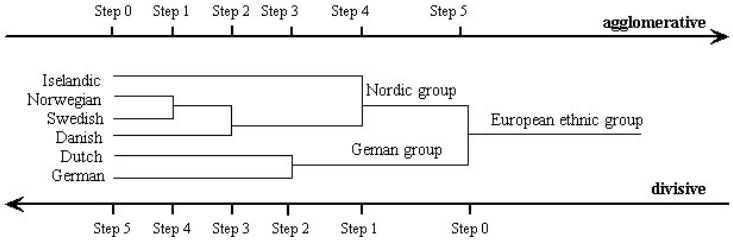

Рисунок 3. Пример дендрограммы (1)

4.5.2. Агломеративная кластеризация

Агломеративная иерархическая кластеризация была впервые предложена Мёртагом и Рафтери [27], Банфилдом и Рафтери [28]. Это вложенная иерархическая кластеризация, которую лучше всего можно представить с помощью дендрограммы, ветви которой совпадают с исходными точками данных, подлежащими кластеризации. Для объединения данных в группы используется подходящая мера близости, которая сначала оценивает сходство между точками, а затем сходство между группами точек.

У агломеративной кластеризации есть несколько преимуществ, таких как то, что она начинается с количества кластеров, равного количеству исходных точек данных, а затем, в результате итеративного процесса группировки похожих точек данных, в конечном итоге получается один кластер, содержащий все заданные точки данных. Это позволяет легко определить расстояние между кластерами. Если объединение происходит между кластерами, расположенными на большем расстоянии, чем предыдущее объединение, можно решить, следует ли сокращать расстояние, когда кластеры находятся слишком далеко друг от друга или когда количество кластеров достаточно мало. Однако это не очень эффективно, когда речь идет о работе с большими данными.

В последний год большую популярность приобрела агломеративная иерархическая кластеризация, основанная на гауссовой вероятности. Здесь для объединения в новый кластер на каждом этапе или итерации выбираются только кластеры с максимальной вероятностью. Этот метод был очень эффективно использован в нескольких практических приложениях [29]. Он находит применение в геофизических, биологических и химических науках, финансовых системах, обработке изображений и многих других областях.

Классический агломеративный метод, как правило, основан на сумме квадратов, одиночном и полном соединении [30], где существует стоимость, основанная на геометрических характеристиках кластера, связанная с объединением каждой пары кластеров. Эта стоимость пары сохраняется до тех пор, пока ни один из элементов этого кластера не участвует в каком-либо другом объединении. В случае классического метода существует простое рекуррентное соотношение для обновления стоимости объединяющейся пары. Например, в случае метода суммы квадратов рекуррентное соотношение выглядит так:

$$\Delta\big((p,q),o\big) = \frac{\big(n_p + n_o\big)\Delta(p,o) + \big(n_q + n_o\big)\Delta(q,o) - n_o\,\Delta(p,q)}{\big(n_p + n_q + n_o\big)}$$

где *Δ(p,q)* – стоимость объединения групп *p* и *q*. Количество пространства, необходимое для объединения кластеров, уменьшается по мере увеличения количества групп.

Здесь множество или популяция состоит из *G* различных подмножеств; плотность *p*-мерных наблюдений, принадлежащих к *k*-му подмножеству. В случае алгоритма Гаусса, представленного в виде «*F*», которая состоит из k-мерных компонентов, заданных следующим образом:

$$F(y) = \sum_{i=1}^{k} \alpha_i \, N(y;\, \mu_i,) = \sum_{i=1}^{k} \alpha_i \, f_i(y)$$

4.5.3. Дивизиональная кластеризация

Дивизиональная кластеризация отличается от агломеративной. Она начинается с одного кластера для всех точек данных, а затем рекурсивно разделяет данные на непересекающиеся кластеры. Затем этот процесс продолжается до тех пор, пока не будет достигнуто определённое количество кластеров. Её главный недостаток в том, что здесь мы должны указать условие завершения и количество кластеров, что может привести к проблеме проверки кластеров.

4.5.4. Разбивочная кластеризация

В разбивочной кластеризации данные распределяются по разным кластерам на основе какого-либо критерия. Обычно количество кластеров задаётся заранее. Для оценки качества кластера можно использовать несколько принципов. Среди них широко распространённым и приемлемым критерием является среднее расстояние между кластерами и объектами внутри кластеров. Для этой цели лучше всего использовать евклидовы расстояния. Результирующее разбиение должно обладать следующими свойствами: однородностью внутри кластеров, т. е. данные, принадлежащие одному кластеру, должны быть как можно более схожими, и неоднородностью между кластерами, т. е. данные, принадлежащие разным кластерам, должны быть как можно более разными [31].

Кластеризацию на основе секционирования можно разделить на алгоритмы вероятностной и нечёткая кластеризации. Наиболее распространенными типами методов секционированной кластеризации являются кластеризация с использованием K- Means (Метод k-средних), Нечёткая (C Means)-кластеризация (кластеризация методом C-средних), алгоритм кластеризации QT и т. д.

4.5.5. K- Means (Метод k-средних) кластеризация

Метод k-средних очень простой итеративный процесс. Здесь количество кластеров «K» задано заранее [32–34]. Выбор правильного количества кластеров и соответствующего размещения их центров должен быть очень продуманным, так как это влияет на будущие результаты. Одна из идей может заключаться в том, чтобы изначально разместить их как можно дальше друг от друга. Затем алгоритм будет итеративно изменять и в конечном итоге фиксировать их, вычисляя значения расстояний по формуле Евклида. Алгоритм завершится, когда положение центров масс перестанет меняться. Основная цель алгоритма K-средних – минимизировать целевую функцию. То есть,

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

где $\left\| x_i^{(j)} - c_j \right\|^2$ – это расстояние между точкой данных $x_i^{(j)}$ и центром кластера c_j, которое указывает на расстояние между n точками данных и их собственными центрами кластеров [35]. Это один из наиболее часто используемых алгоритмов кластеризации. Даже в таких нейронных сетях, как RBFNN, метод K-средних очень эффективен для определения структуры сети и таких параметров, как весовые коэффициенты, центр и ширина как скрытого, так и выходного слоя.

У алгоритма K-средних есть несколько проблем, таких как застревание в локальных оптимумах [36] и появление мёртвых ячеек, особенно когда данные имеют больше измерений. Чтобы преодолеть эти проблемы, используются несколько методов, таких как поэтапное автоматическое сопоставление с штрафом [34], Q-битовый квантовый [33], K-гармоническое среднее [37] и многие другие.

4.5.6. Нечёткая (C Means)-кластеризация (кластеризация методом С-средних)

При методе С-средних каждый элемент данных имеет степень принадлежности к каждому кластеру, а не к одному кластеру целиком. Это означает, что точки данных, расположенные ближе к центру кластера, имеют более высокую степень принадлежности к этому кластеру по сравнению с точками данных, расположенными на границах кластера. С каждой точкой *x* связан коэффициент, определяющий степень принадлежности к некоторому *k*-му кластеру *Uk (X)*. FCM минимизирует целевую функцию с учётом некоторых ограничений. Например, сумма степеней принадлежности каждой точки данных ко всем кластерам должна быть равна единице [38, 39].

$$\forall x \left(\sum_{k=1}^{num.of\ clusters} u_k\ (x) = 1 \right)$$

Для вычисления центроида кластера в методе размытых С-средних вычисляется среднее значение всех точек с учётом степени их принадлежности к кластеру [38,40,41]. Степень принадлежности точки данных $u_k(x)$ равна обратному значению расстояния до центра кластера.

$$center_k = \frac{\sum_x u_k\ (x)^m\, x}{\sum_x u_k\ (x)^m}$$

$$u_k\,(x) = \frac{1}{d(center_k, x\,)}$$

Затем коэффициенты нормализуются и образуют fuzzy поле с реальным параметром $m > 1$, так что их сумма равна *1*.

Таким образом

$$u_k\,(x) = \frac{1}{\sum_j \left(\frac{d\left(center_{k\,,}x\right)}{d\left(center_{j\,,}x\right)} \right)^{2/(m-1)}}$$

Если *m* равно или близко к *1*, то это просто означает, что ближайшему к точке центру кластера присваивается гораздо больший вес, чем остальным. В случае, когда m равно *2*, это эквивалентно линейной нормализации коэффициентов, чтобы их сумма равнялась 1.

4.5.7. Генетический алгоритм

Термин «генетический алгоритм» возник из-за того, что этот алгоритм имитирует процесс биологической эволюции. Здесь организмы рассматриваются как хромосомы, где каждая хромосома представляет собой возможное решение в определённой области. Группа таких организмов составляет популяцию, которую необходимо поддерживать от поколения к поколению. Генетический алгоритм хорошо справляется с формированием необходимого количества кластеров и обеспечивает надлежащую кластеризацию там, где другие алгоритмы терпят неудачу. Изначально он генерирует случайную популяцию, а затем, следуя принципу эволюции «выживает сильнейший», генерирует новую популяцию.

Каждый член популяции оценивается и получает показатель своей пригодности в качестве решения [42]. Здесь используются три генетических оператора: отбор, скрещивание и мутация. Отбор – это присвоение хромосомам возможности воспроизводства на основе их пригодности [43]. Скрещивание – это объединение признаков двух родителей для получения двух потомков. Мутация – это изменение одного или нескольких генов потомка с низкой вероятностью, чтобы избежать застревания в локальном оптимуме [44, 45].

4.5.8. Кластеризация на основе плотности и сетки

Метод кластеризации на основе плотности использует объекты-соседи в пределах определённого радиуса внутри заданного кластера. Наиболее распространёнными и эффективными алгоритмами, основанными на плотности, являются DBSCAN (пространственная кластеризация приложений с шумом на основе плотности) и (OPTICS) упорядочивание точек для определения структуры кластеризации.

Алгоритмы, основанные на плотности, рассматривают кластеры как плотные области объектов, разделённые менее плотными областями. Их преимущество перед алгоритмами, основанными на разделении, заключается в том, что они не ограничиваются поиском кластеров только сферической формы, а могут находить кластеры произвольной формы [46, 47].

4.5.9. Кластеризация DBSCAN

Кластеризация DBSCAN основана на концепции достижимости и связности по плотности [48]. У нее есть два входных параметра: эпсилон и минимальное количество точек. Эпсилон – это расстояние вокруг объекта, определяющее его окрестности [49], а минимальное количество точек необходимо для того, чтобы считать объект основным, а объекты в его окрестностях – достижимыми по плотности из этой точки.

Это означает, что точка *S* достижима по плотности из точки *Q,* если она находится на расстоянии не более *R,* которое также называется пороговым значением. Две точки могут располагаться в несимметричном порядке или быть соединены через другую точку *P*, образуя несимметричную фигуру. Это называется соединением по плотности. Средняя точка *P* должна быть соединена по плотности как с точкой *S,* так и с точкой *Q*. В этом случае результирующий кластер должен содержать точки данных, которые связаны друг с другом по плотности, и любая другая точка, связанная по плотности с любой из этих точек, также является частью того же кластера. Этот алгоритм непрерывно увеличивает кластеры, добавляя в них все объекты, до которых можно добраться от основного объекта.

4.5.10. OPTICS Алгоритм

OPTICS Алгоритм – это усовершенствованная версия DBSCAN, которая устраняет его недостатки. У DBSCAN существует проблема с определением значимых кластеров, особенно в данных с разной плотностью. Здесь база данных упорядочена в соответствии со структурой кластеризации. Важным свойством реальных наборов данных

является то, что их внутренняя кластерная структура не может быть охарактеризована глобальными параметрами плотности. Для выявления кластеров в разных областях пространства данных требуются очень разные значения плотности.

Оптика обеспечивает особый порядок базы данных в соответствии со структурой кластеризации на основе плотности, которая содержит информацию о каждом уровне кластеризации набора данных и становится очень простой для анализа. Она также известна как алгоритм иерархической кластеризации на основе плотности. Это очень полезно при работе с неопределёнными данными, когда расстояние или сходство между неопределёнными объектами измеряется одним числовым значением. На основе таких однозначных функций расстояния OPTICS, как и другие стандартные алгоритмы интеллектуального анализа данных, может работать без каких-либо изменений.

4.5.11. Алгоритм кластеризации по Гауссу (EM)

Кластеризация по Гауссу связана с подбором набора гауссовских кривых для данных. Она определяет набор гауссовских распределений, которые с наибольшей вероятностью соответствуют данным. Здесь изначально k компонентов генерируются случайным образом путём объединения компонентов многомерной нормальной плотности [50].

Случайный вектор считается многомерно нормально распределённым, если каждая комбинация компонентов имеет одномерное нормальное распределение. Как правило, многомерное распределение описывает, по крайней мере, приблизительно набор случайных величин, каждая из которых группируется вокруг среднего значения [51]. Затем данные подгоняются с помощью алгоритма максимизации ожидаемого значения, который присваивает вероятность каждому компоненту на основе отдельных наблюдений. Эту вероятность иногда также называют показателем принадлежности или рангом. Каждая точка данных имеет показатель принадлежности к каждому кластеру.

Это позволяет решить многие проблемы, связанные с другими методами кластеризации, и получить более стабильные кластеры, особенно когда запрашиваемое количество кластеров меняется [52].

В общем случае модель гауссовской системы состоит из *'n'* компонентов. Среди них *k-й* компонент можно называется W_k со средним вектором μ_k [53]. Компонент генерирует данные по нормальному распределению со средним вектором μ и вариационной матрицей σ^2. Таким образом, нормальное распределение *k*-мерного случайного вектора *X* выглядит следующим образом:

$$X \sim \mathrm{N}_k\,(\,\mu_k\,,\,\sigma^2{}_k\,)$$

Гауссова система оценивает функцию плотности вероятности (PDF) для каждого класса данных, а затем для классификации применяется правило Байеса.

$$P(C_i\,|X) = P(X|C_i).\frac{P(C_i)}{P(X)}$$

В приведённом выше уравнении $P(X|C_i)$ – это функция плотности вероятности класса «i», которая рассчитывается в некоторой точке *'X'*, тогда как $P(C_i)$ – это предшествующая вероятность класса «*i*», а $P(X)$ – это общая вероятность функции плотности вероятности, рассчитанная в точке *'X'*. Здесь функция $P(X|C_i)$оценивается с помощью модели гауссовской системы как указано ниже:

$$P(X|C_i) = \frac{1}{n}\sum_{k=1}^{N} w_k G_k$$

Формула представляет собой вес *k*-го гауссова распределения. Для каждого класса генерируется уникальная модель функции плотности вероятности, где каждый компонент определяется как:

$$G_k = \frac{1}{(2\pi)^{\frac{n}{2}}|\,V_k|^{\frac{1}{2}}}\,.exp\,(\,-\frac{1}{2}\,(X-\mu_k)^T\,V_K{}^{-1}(X-\mu_k))$$

Здесь G_k – среднее значение, а V_k – её ковариационная матрица. Три параметра модели гауссовой системы – это среднее значение, ковариационная матрица компонентов гауссовой системы и вес, указывающий на вклад гауссовой системы в приближение вероятности.

В целом мы имеем:

$$P(C_i\,|X) = P(X|C_i).\frac{P(C_i)}{P(X)}$$

$$P(X|C_i) = \frac{1}{n}\sum_{k=1}^{N} w_k G_k$$

$$G_k = \frac{1}{(2\pi)^{\frac{n}{2}}|\,V_k|^{\frac{1}{2}}}\,.exp\,(-\frac{1}{2}\,(X-\mu_k)^T\,{V_K}^{-1}(X-\mu_k))$$

Мы использовали метод оценки и максимизации (EM) для аппроксимации этих переменных [54]. Цель этих алгоритмов EM – максимизировать вероятность обучающего набора, сгенерированного функцией плотности вероятности [55]. Функция вероятности *L* для каждого класса *j* может быть определена как:

$$Likelihood_j\; = \prod_{i=0}^{N} P(x_i\,|C_j\,)$$

4.6. Сравнение известных методов кластеризации

4.6.1. K-Means кластеризация (метод K-средних), его преимущества и ограничения

Данный алгоритм сначала инициализирует / определяет начальные центральные точки для всех кластеров набора данных. Затем алгоритм распределяет данные по различным кластерам, чтобы минимизировать расстояние между ними и центром кластера.

Преимущества:

- количество кластеров необходимо определить заранее;
- его очень просто реализовать;
- он отличается высокой скоростью, что позволяет работать с большими наборами данных.

Ограничения:

- это не приводит к одному и тому же результату при каждом выполнении, поскольку результирующие кластеры зависят от начальных случайных назначений;
- это минимизирует дисперсию внутри кластера, но не гарантирует, что результат будет иметь глобальную минимальную дисперсию;
- требование к понятию среднего значения должно быть четким, что в данном случае не всегда возможно;
- это самый популярный метод, но он очень чувствителен к инициализации, и чем лучше мы выбираем центры, тем лучшие результаты получаем;
- он попадает в локальное оптимальное решение и плохо работает с различными формами.

4.6.2. Нечёткая C-Means-кластеризация (Метод C-средних), его преимущества и ограничения

Этот алгоритм основан на оптимизации функции стоимости, необходимой для определения центров кластеров. Функция стоимости зависит от исходных элементов набора данных и их характеристик. Изначально кластеры набора данных, созданные алгоритмом, являются нечёткими (Fuzzy). Это означает, что элементы набора данных не относятся к определённому кластеру данных. Вместо этого элементам набора данных присваивается степень принадлежности к каждому кластеру, созданному алгоритмом.

Преимущества:

- здесь каждый элемент набора данных имеет степень принадлежности к кластерам, а не полностью принадлежит только одному кластеру;
- это минимизирует внутрикластерную дисперсию.

Ограничения:

- алгоритм является локальным, и результат зависит от первоначального выбора веса. Локальные алгоритмы – это наибольшие или наименьшие значения функции в непосредственной близости;

- алгоритм Нечёткая (C Means)-кластеризация (кластеризация методом С-средних) имеет тенденцию к попаданию в локальный оптимум;
- алгоритм Нечёткая (C Means)-кластеризация (кластеризация методом С-средних) получает глобальные оптимальные центры кластеров. В основном используется при распознавании фигур, тенденций данных и т. д.

4.6.3. Иерархическая кластеризация, её преимущества и ограничения

Этот алгоритм создаёт иерархию кластеров, а не фиксирует определённое количество кластеров в начале. На начальном уровне каждый элемент набора данных создаёт свой собственный кластер. На каждом последующем уровне два «ближайших» кластера объединяются в один более крупный кластер. Здесь используется метод, который называется «среднее». Это означает, что расстояние между двумя кластерами – это среднее значение расстояний между точками в одном кластере и точками в другом кластере. Он также известен как UPGMA (метод невзвешенных парных групп со средним арифметическим).

Преимущества:

- в алгоритме количество кластеров не фиксируется заранее;
- алгоритм прост в реализации;
- алгоритм хорошо работает, особенно когда требуется выявить редкие или необычные элементы набора данных.

Ограничения:

Это медленный метод, и для создания кластеров требуется больше времени по сравнению с другими методами кластеризации.

4.6.4. Алгоритм кластеризации по Гауссу (Gaussian), его преимущества и ограничения

Это генеративная модель, в которой каждый кластер соответствует распределению Гаусса. Для набора данных, метод максимального правдоподобия используется для изучения модели гауссовского

смешивания (Gaussian Mixture Model - GMM) с целью максимизации вероятности данных.

Преимущества:

- алгоритм даёт хорошие результаты для задач реального мира;
- алгоритм лучше работает с многомерными данными, где другие алгоритмы кластеризации не справляются из-за того, что расстояние между точками становится более равномерным.

Ограничения:

- алгоритм очень сложен по своей природе. Он может застрять в локальных минимумах;
- проблема возникает из-за слишком большого количества свободных параметров в ковариационных матрицах кластерного распределения;
- это может привести к чрезмерной подгонке данных.

Выводы по главе 4

Вышеописанные алгоритмы были сопоставлены и применены к набору данных, который используется в данной исследовательской работе. Цель применения всех известных методов кластеризации к нашему набору данных – определить метод, который лучше всего подходит для выявления редких или необычных случаев в наборе данных. В следующей главе обсудим разработку программного обеспечения и реализацию различных методов.

ГЛАВА 5. РАЗРАБОТКА И ВНЕДРЕНИЕ ПРОГРАММНОЙ МОДЕЛИ ДЛЯ ИДЕНТИФИКАЦИИ РЕДКИХ ЭЛЕМЕНТОВ ДАННЫХ ИЗ БАЗЫ ДАННЫХ

Программное обеспечение для реализации кластеризации было создано с помощью Matlab. Matlab был выбран потому, что это одна из наиболее широко используемых программ, в которой есть множество встроенных функций для математического анализа данных. Кластеризация включает в себя математические вычисления, такие как определение расстояния между элементами набора данных и местоположения их центров тяжести.

На первом этапе мы запускаем программное обеспечение Matlab. С помощью программного интерфейса файл набора данных, расположенный на компьютере, загружается в программу для анализа. На рисунках 4 и 5 показан начальный графический интерфейс программы.

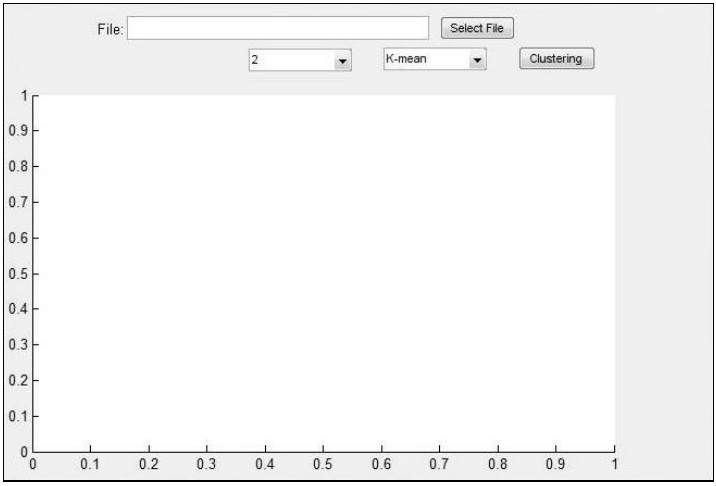

Рисунок 4. Графический интерфейс программного обеспечения для кластеризации

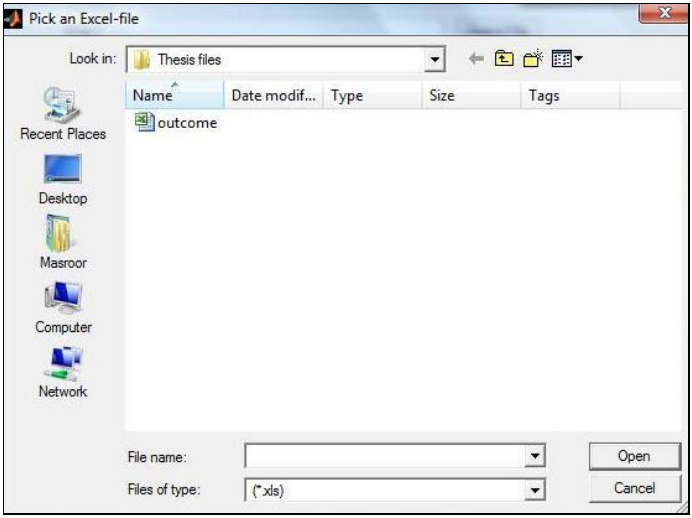

Рисунок 5. Загрузка файла набора данных для кластеризации

После поиска файла набора данных он открывается с помощью графического пользовательского интерфейса программы, как показано на Рисунке 6.

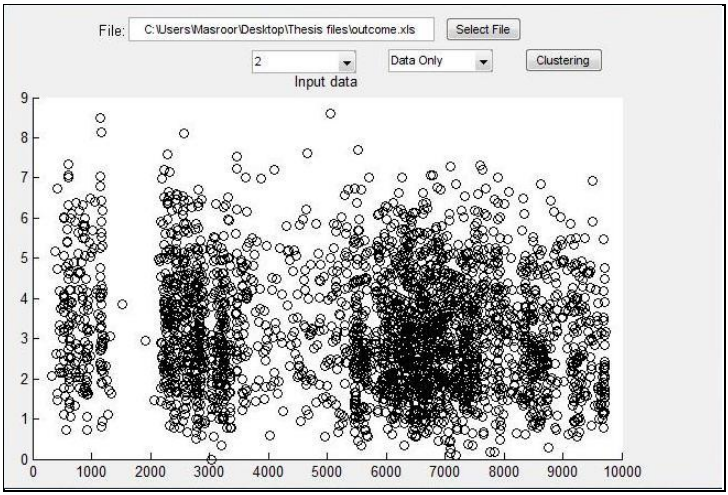

Рисунок 6. Представление данных в виде графика

Набор данных на Рисунке 6 в графическом интерфейсе программы представлен одним цветом. Набор данных одного цвета показывает, что кластеризация ещё не началась и набор данных не распределён по группам. В пункте меню над графиком мы выбрали два кластера для процесса кластеризации, как представлено на Рисунке 7:

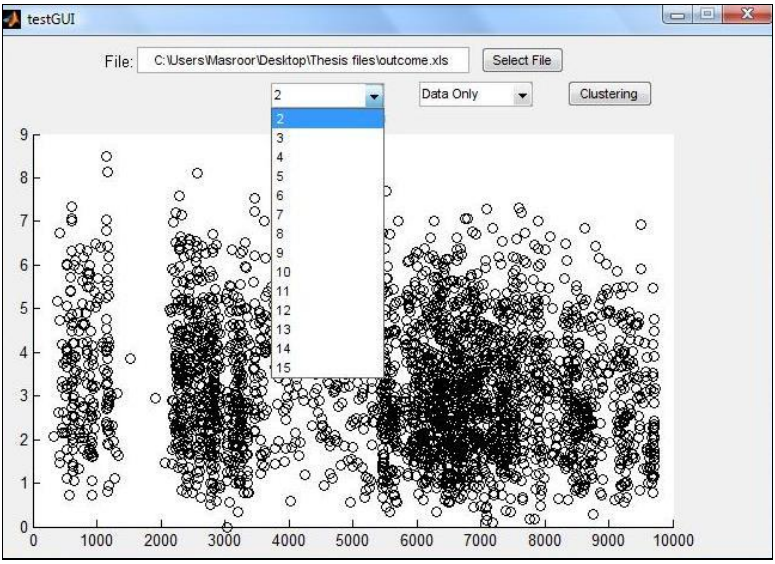

Рисунок 7. Графический интерфейс для выбора количества кластеров для кластеризации

Дальше необходимо выбрать тип кластеризации из выпадающего меню для разделения набора данных на кластеры, как показано на Рисунке 8.

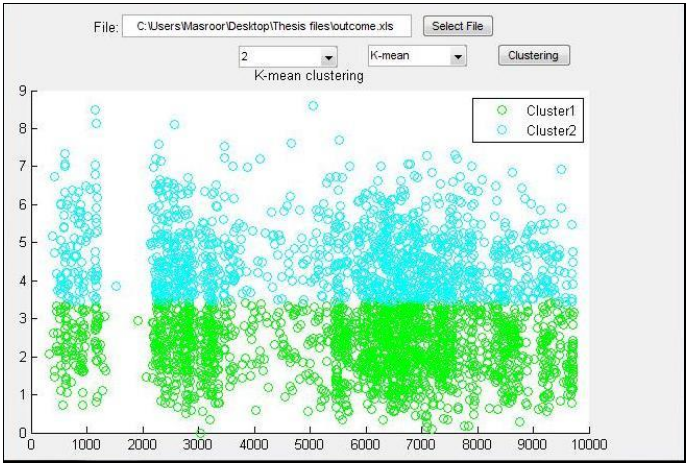

Рисунок 8. Разделение набора данных на два кластера с использованием кластеризации по методу K-Means (Метод K-средних) кластеризация

Чтобы увеличить количество кластеров в наборе данных, мы можем выбрать количество кластеров в меню «Количество кластеров», как описано выше и показано ниже на Рисунке 9.

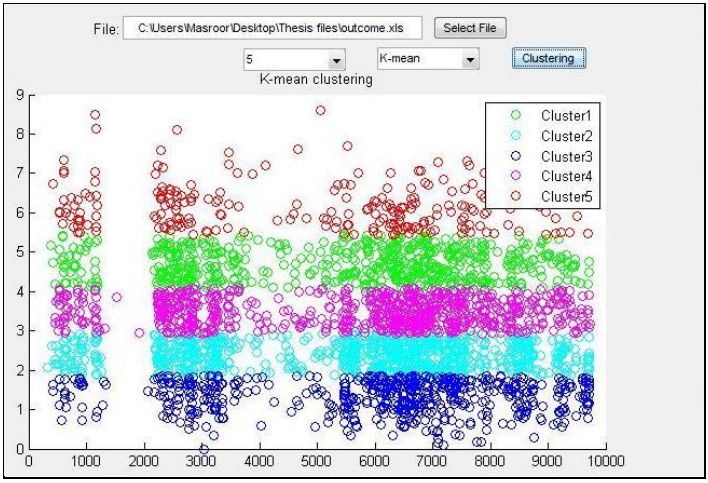

Рисунок 9. Набор данных, сгруппированный в пять кластеров

На Рисунке 9 разными цветами обозначены разные кластеры в наборе данных. Элементы набора данных, обозначенные одним цветом, принадлежат к одному кластеру, а данные, обозначенные разными цветами, принадлежат к разным кластерам. Чтобы было понятнее, используется легенда, которая показывает, какой цвет соответствует каждому номеру кластера.

После кластеризации методом K-Means (Метод K-средних) кластеризация на наборе данных с 5 желаемыми кластерами были использованы другие алгоритмы кластеризации такие как Нечёткая (C Means)-кластеризация, Гауссова и Иерархическая кластеризация, чтобы оценить их эффективность при разделении набора данных на разные кластеры. Результаты кластеризации с использованием других методов кластеризации дополнительно представлены на рисунках ниже.

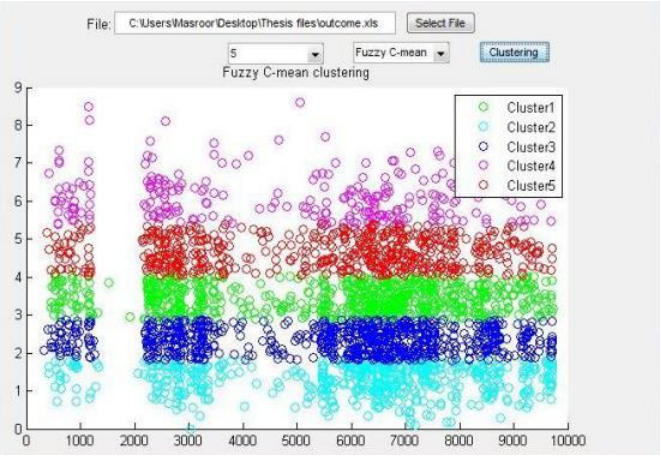

Рисунок 10. Набор данных, сгруппированный в пять кластеров с помощью кластеризации методом Нечёткая (C Means)-кластеризация (кластеризация методом С-средних)

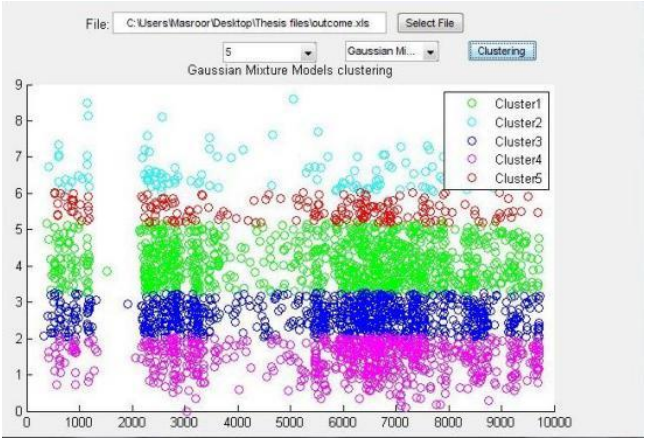

Рисунок 11. Набор данных, сгруппированный в пять кластеров с помощью кластеризации методом Гаусса (Guassian)

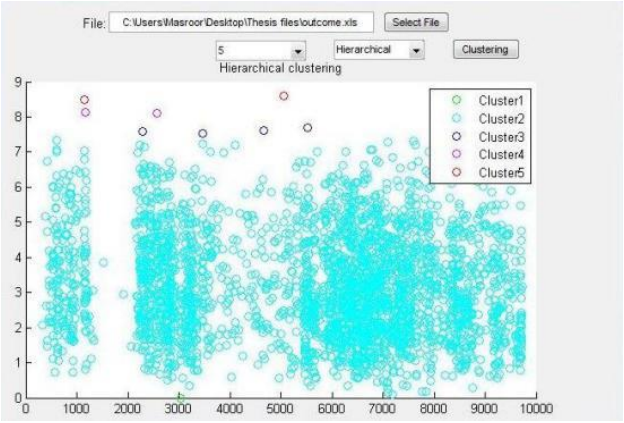

Рисунок 12. Набор данных, сгруппированный в пять кластеров с помощью Иерархической кластеризации (Hierarchical clustering)

На приведённых выше рисунках можно наблюдать, как различные алгоритмы группируют данные в разные кластеры. Другие характеристики алгоритмов, такие как время выполнения, Иерархической кластеризации редкие элементы набора данных, описаны в следующем разделе.

Выводы по главе 5

В главе 5 обсуждается интерфейс программной системы, разработанной в ходе данной исследовательской работы. В ней описывается платформа программирования Matlab, которая используется при разработке системы. Она предоставляет пользователю системы пошаговую информацию с помощью рисунков о том, как загрузить базу данных в систему и как применять различные методы кластеризации в системе с различным количеством кластеров, выбранных для каждого алгоритма кластеризации. Графический интерфейс помогает визуализировать результаты кластеризации по каждому алгоритму и помогает дифференцировать и сравнивать результаты различных алгоритмов кластеризации.

ГЛАВА 6. АНАЛИЗ И СРАВНЕНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ ДЛЯ ИДЕНТИФИКАЦИИ РЕДКИХ ЭЛЕМЕНТОВ ДАННЫХ ИЗ БАЗЫ ДАННЫХ

Прежде чем анализировать алгоритмы кластеризации на наборе данных, необходимо разобраться с набором данных и его атрибутами. Каждый набор данных имеет свою природу и характеристики, которые влияют на выбор алгоритма для его обработки.

6.1. Набор данных и его атрибуты

Набор данных, используемый в этой работе, связан с послеоперационной болью у пациентов. В наборе данных подробно описаны различная информация, такая как история болезни пациентов, их возраст, пол и т. д., для анализа факторов, которые приводят к усилению боли у пациентов после операции. Набор данных состоит из двух основных частей: проблемы и решения. В проблемной части обсуждается история болезни и другая информация о пациентах. Это многоатрибутный набор данных. С другой стороны, в выходной части есть только один атрибут, и этот атрибут будет использоваться для идентификации редких или необычных случаев из набора данных. Редкие или необычные случаи – это случаи, в которых наблюдается отклонение в поведении от нормального, даже при приеме тех же лекарств и лечении.

Проблемная часть набора данных сначала предварительно обрабатывается, а затем значения, полученные в рамках задачи, используются для группировки данных в различные кластеры.

Следующий этап кластеризации начинается с использования выходных значений каждого кластера для одновременного выявления редких случаев или элементов набора данных в каждом из кластеров. Редкие случаи – это элементы набора данных, которые, по-видимому, находятся на границе своего кластера. Они расположены далеко от других элементов того же кластера.

6.2. Оценка производительности алгоритмов кластеризации

С помощью программного обеспечения, разработанного в ходе этого исследования, каждый алгоритм кластеризации применяется отдельно к набору данных о послеоперационной боли. После их раздельного анализа проводится сравнение для того, чтобы понять характеристики алгоритмов кластеризации и выбрать метод кластеризации, который лучше подходит для анализа набора данных, используемого в этом исследовании.

Проблемная часть набора данных о послеоперационной боли состоит из 231 общей характеристики, которые зависят от 13 основных характеристик. Таким образом, значения всех 231 характеристик обрабатываются и сопоставляются с 13 значениями характеристик. После этого к проблемным признакам была применена кластеризация первого порядка для группировки данных в несколько кластеров. Для кластеризации проблемной части набора данных были использованы четыре наиболее известных метода кластеризации, выявленных в ходе исследования.

6.2.1. Частотное распределение набора данных по кластерам

Первой характеристикой проанализированных алгоритмов является их способность распределять набор данных по кластерам. Сначала наблюдается частотное распределение с двумя кластерами, а затем количество кластеров увеличивается до пяти. Результаты частотного распределения при использовании различных методов с фиксированным количеством кластеров, равным 2, можно увидеть в Таблице 8.

Таблица 8

Элементы данных частотного распределения для двух кластеров с использованием различных методов кластеризации

Frequency Distribution (Частотное распределение)		
Количество кластеров = 2		
Метод кластеризации	**Кластер 1**	**Кластер 2**
K-Means (Метод k-средних)	2902	891
Нечёткая (C Means)-кластеризация (кластеризация методом C-средних)	916	2877
Метод Гаусса (Guassian)	61	3632
Иерархическая кластеризация (Hierarchical clustering)	3	3790

Из приведённой выше таблицы видно, что алгоритмы кластеризации k-средних дают лучшие результаты группировки данных, в то время как Иерархическая кластеризация является наиболее ограниченным методом группировки данных в кластеры. Это можно хорошо проиллюстрировать с помощью графического представления данных на Рисунке 13.

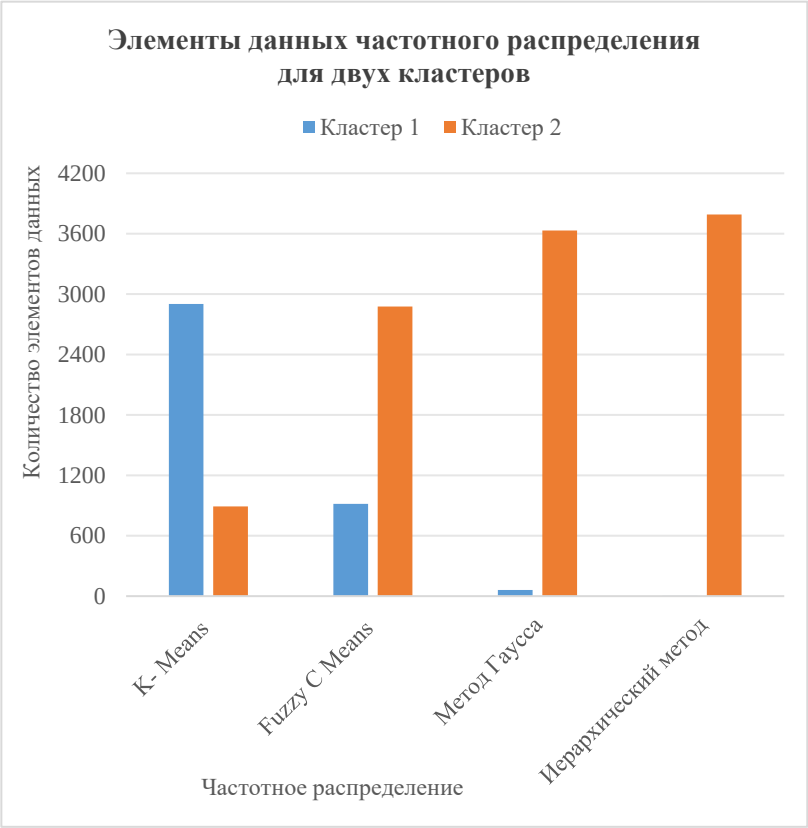

Рисунок 13. Частотное распределение проблемных данных для двух кластеров с использованием различных методов кластеризации

Теперь общее количество кластеров увеличено до трёх, чтобы более точно проанализировать эффективность всех алгоритмов. Таким образом, выбор наиболее эффективного метода кластеризации станет проще. В таблице 9 показано распределение данных по трём кластерам.

Таблица 9

Частотное распределение данных по трем кластерам

Frequency Distribution(Частотное распределение)			
Количество кластеров = 3			
Метод кластеризации	**Кластер 1**	**Кластер 2**	**Кластер 3**
K- Means (Метод k-средних)	335	2873	585
Нечёткая (C Means)-кластеризация (кластеризация методом C-средних)	1701	1192	900
Метод Гаусса (Guassian)	250	300	3243
Иерархическая кластеризация	2	3788	3

Из приведённой выше таблицы видно, что кластеризации K-средних и Нечёткая (C Means)-кластеризация (кластеризация методом C-средних) обеспечивают более эффективную группировку или распределение набора данных по кластерам. В то время как Иерархическая кластеризация является наиболее ограниченной методикой при группировке данных. Результат показан на Рисунке 14.

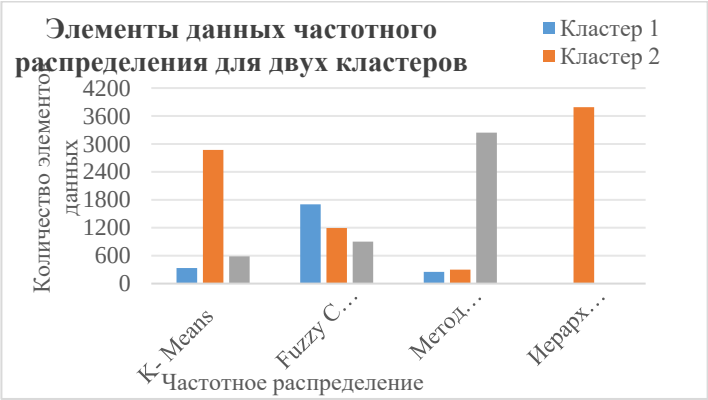

Рисунок 14. Частотное распределение данных по трём кластерам с использованием различных методов кластеризации

Общее количество кластеров снова увеличивается до 4. В Таблице 10 показана способность алгоритмов кластеризации распределять набор данных по 4 кластерам.

Таблица 10

Частотное распределение набора данных по 4 кластерам с использованием различных методов кластеризации

Frequency Distribution (Частотное распределение)				
Количество кластеров = 4				
Метод кластеризации	**Кластер 1**	**Кластер 2**	**Кластер 3**	**Кластер 4**
K- Means (Метод k-средних)	319	2885	585	4
Нечёткая (C Means)-кластеризация (кластеризация методом C-средних)	759	1366	791	877
Метод Гаусса (Guassian)	12	40	140	3767
Иерархическая кластеризация (Hierarchicalclustering)	1	2	2	3788

Из приведённой выше таблицы очень легко понять, что кластеризация с помощью Нечёткая (C Means)-кластеризация (кластеризация методом C-средних) обеспечивает более эффективную группировку или распределение данных по кластерам. В то время как Иерархическая кластеризация по-прежнему демонстрирует свои ограничения

при группировке данных по кластерам. Это можно заметит на графическом представлении кластеров на Рисунке 15.

Рисунок 15. Частотное распределение данных по четырём кластерам с использованием различных методов кластеризации

Теперь количество кластеров для распределения данных по группам увеличено до 5. Это поможет более эффективно проанализировать работу всех алгоритмов. В Таблице 11 показано распределение набора данных по пяти кластерам с использованием различных методов кластеризации.

Таблица 11

Частотное распределение данных по пяти кластерам с использованием различных методов кластеризации

Frequency Distribution (Частотное распределение)					
Количество кластеров = 5					
Метод кластеризации	**Кластер 1**	**Кластер 2**	**Кластер 3**	**Кластер 4**	**Кластер 5**
K- Means (Метод k-средних)	900	3	4	1511	1375
Нечёткая (C Means)-кластеризация (кластеризация методом C-средних)	628	570	675	820	1100
Метод Гаусса (Guassian)	3	2	610	78	3100
Иерархическая кластеризация (Hierarchical clustering)	2	3786	1	2	2

Заметно, что Нечёткая (C Means)-кластеризация (кластеризация методом C-средних) обеспечивает более эффективную группировку или распределение данных по кластерам. В то время как Иерархическая кластеризация является наиболее ограниченным методом группировки данных. Результаты дополнительно проиллюстрированы на Рисунке 16.

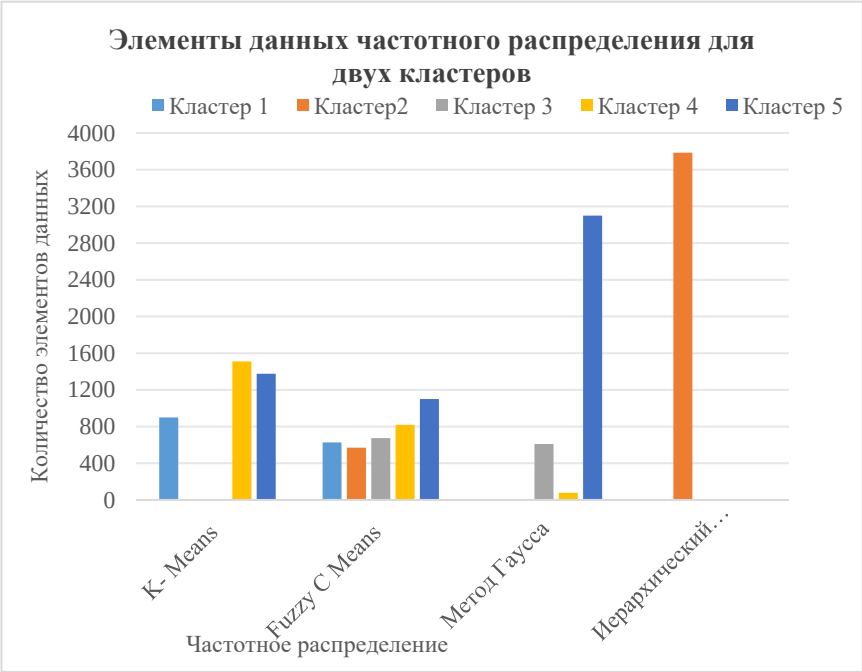

Рисунок 16. Частотное распределение данных по пяти кластерам с использованием различных методов кластеризации

6.2.2. Затраченное время (Elapsed time) работы алгоритмов кластеризации

Другим параметром, который используется для измерения и сравнения эффективности алгоритмов кластеризации, является затраченное время. Алгоритм, который требует меньше времени для группировки данных в желаемое количество кластеров, определённо лучше подходит для группировки данных в различные кластеры. В Таблице 12 показано затраченное время для различных алгоритмов кластеризации при увеличении количества кластеров в наборе данных.

Таблица 12

Время, затраченное на увеличение общего числа кластеров

Затраченное время (секунды)				
Количество кластеров	**K- Means (Метод k-средних)**	**Нечёткая (C Means)-кластеризация (кластеризация методом С-средних)**	**Кластеризация методом Гаусса (Guassian)**	**Иерархическая кластеризация (Hierarchical clustering)**
2	0.046111	0.210114	0.08542	17.047294
3	0.033135	1.184806	0.2463	17.556769
4	0.078967	1.135368	0.5631	18.008661
5	0.299539	1.878679	3.7589	16.037917

Приведённая выше Таблица 12 помогает проанализировать и сравнить производительность различных технологий кластеризации с точки зрения затраченного времени на группировку данных в кластеры. На Рисунке 17 показано графическое представление затраченного времени алгоритмов на группировку данных в кластеры.

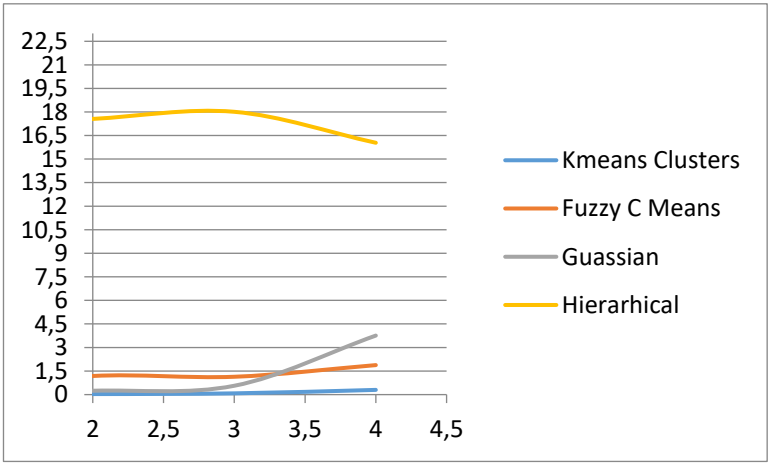

Рисунок 17. Затраченное время на различные методы кластеризации для разделения данных на разное количество кластеров

Из Таблицы 12 и Рисунка 17 выше можно проанализировать и сравнить время, затраченное на алгоритмы кластеризации. Заметно, что методы кластеризации K-средних требуют меньше всего времени при разделении данных на различные группы, в то время как Иерархическая кластеризация требует больше всего времени при разделении набора данных на кластеры.

6.3. Зависимость между количеством кластеров и затраченным временем работы алгоритмов

Прошедшее время и его взаимосвязь с количеством кластеров – важный фактор при выборе оптимального количества кластеров для достижения наилучших результатов.

Зависимость между количеством кластеров и временем их создания при использовании алгоритма кластеризации

Важно проанализировать время, затрачиваемое алгоритмом кластеризации, в зависимости от количества кластеров, создаваемых алгоритмом. Производительность и результативность алгоритма в значительной степени зависят от затраченного на него времени. Это также демонстрирует способность алгоритма изучать набор данных и сокращать затрачиваемое время по мере увеличения количества итераций на одном и том же наборе данных. Чтобы понять взаимосвязь между затрачиваемым временем и количеством кластеров в наборе данных, используется программа, разработанная в ходе этого исследования и описанная в предыдущих главах.

6.3.1. Прошедшее время создания кластеров с использованием алгоритма кластеризации K-Means (Метод K- средних)

В случае алгоритма кластеризации K-средних с помощью программного обеспечения можно наблюдать, что при увеличении количества кластеров результаты кластеризации K-средних становятся более понятными, а задача выявления необычных случаев или кластеров необычных случаев становится гораздо более очевидной. В таблице 13

показано время выполнения или затраченное время, необходимое алгоритму K-средних при увеличении количества кластеров.

Таблица 13

Время, затраченное на различные количества кластеров K-средних

	Количество кластеров	**Затраченное время (за считанные секунды)**
K- Means (Метод k-средних) кластеризация	2	0.046111
	3	0.033135
	4	0.078967
	5	0.299539
	6	0.0813504
	7	0.0439034
	8	0.0817532
	9	0.0667596
	10	0.1314968

Из приведённой выше таблицы 13 видно, что время, необходимое для выполнения алгоритма кластеризации K-средних, мало – от 0,03 до 0,13 секунды. Это также означает, что время, необходимое для разбиения данных на кластеры с разным количеством элементов, мало по мере увеличения количества итераций алгоритма. Это можно увидеть на рисунке 18.

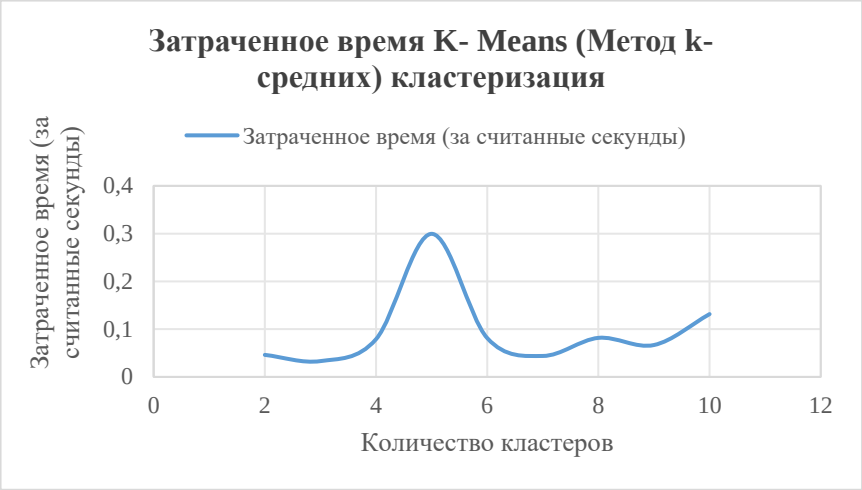

Рисунок 18. Зависимость времени от количества кластеров при использовании алгоритма кластеризации K-средних

Было замечено, что при увеличении количества кластеров в K-средних кластерах время вычислений сокращается. Также видно, что с увеличением количества кластеров идентификация необычных или редких случаев становится проще. Из приведённой выше таблицы 13 видно, что с увеличением количества кластеров возрастает вероятность обнаружения редких случаев. Однако частота появления любого кластера не настолько мала, чтобы его можно было считать кластером редких случаев. Даже многократные итерации не сильно влияют на результаты кластеризации методом K-средних. Графическое представление с помощью таблицы очень хорошо иллюстрирует результаты кластеризации методом K-средних. Таким образом, кластеризация методом K-средних хорошо подходит для разделения случаев на равное или примерно равное количество кластеров, но на основе частотного распределения она недостаточно хорошо подходит для выделения редких случаев из данных.

6.3.2. Прошедшее время создания кластеров с использованием алгоритма кластеризации Нечёткая (C Means)-кластеризация (кластеризация методом C-средних)

Алгоритм кластеризации с использованием Нечёткая (C Means)-кластеризация (кластеризация методом C-средних) изучался путем подбора различного количества кластеров с использованием программного обеспечения, разработанного в ходе данной исследовательской работы. Вначале было выбрано два кластера, затем их количество постепенно увеличивалось до 10 кластеров. В таблице 14 показано время выполнения или затраченное время, необходимое алгоритму Нечёткая (C Means)-кластеризация (кластеризация методом C-средних) для группировки набора данных в различное количество кластеров.

Таблица 14

Время, затраченное на различные количества кластеров Нечёткая (C Means)-кластеризация (кластеризация методом C-средних)

	Количество кластеров	**Затраченное время (за считанные секунды)**
Нечёткая (C Means)-кластеризация (кластеризация методом C-средних) кластеризация	2	0.210114
	3	1.184806
	4	1.135368
	5	1.878679
	6	1.521132
	7	1.229234
	8	1.353776
	9	0.968356
	10	0.911012

Из приведенной выше таблицы 14 также видно, что затраченное время увеличивается по мере последовательного увеличения числа кластеров вплоть до количества кластеров 5. Затем происходит

сокращение времени выполнения алгоритма, это означает, что алгоритмы Fuzzy C-Means изучают структуру данных вплоть до распределения данных по 5 кластерам, и из-за их способности к обучению происходит изменение времени выполнения. Это можно лучше увидеть, изучив рисунок 19, приведенный ниже.

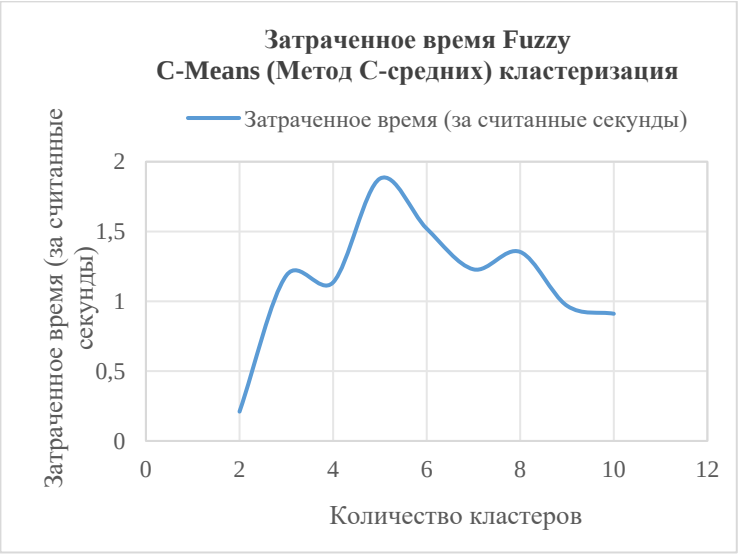

Рисунок 19. Зависимость времени от количества кластеров при использовании Нечёткая (C Means)-кластеризация (кластеризация методом С-средних)

6.3.3. Прошедшее время создания кластеров с использованием алгоритма кластеризации методом Гаусса (Guassian)

Алгоритм гауссовой кластеризации в основном используется для сложных задач, связанных с большим количеством числовых данных и функций. Производительность и результативность алгоритма в значительной степени зависят от затраченного на него времени. Требуется больше времени, чтобы понять структуру набора данных,

применив различные математические модели, и построить взаимосвязь между ними. Алгоритм Гаусса применяется к нашему набору данных для первоначального разделения данных на два кластера, а затем количество кластеров последовательно увеличивается, чтобы наблюдать за влиянием алгоритма на время выполнения по мере увеличения количества кластеров. Время выполнения алгоритма можно проследить, используя таблицу 15.

Таблица 15

Время, затраченное на различные количества кластеров с использованием кластеризации методом Гаусса (Guassian)

	Количество кластеров	**Затраченное время (за считанные секунды)**
Кластеризации методом Гаусса (Guassian)	2	0.08542
	3	0.2463
	4	0.5631
	5	3.7589
	6	3.819664
	7	3.978695
	8	4.2061
	9	4.272087
	10	4.471027

Из приведенной выше таблицы видно, что алгоритму Гуассиана требуется гораздо больше времени для группировки данных в кластеры по сравнению с алгоритмом кластеризации с использованием K-средних (Метод k-средних значений) Кластеризация и алгоритм кластеризации с использованием C-средних значений. Это может быть дополнительно продемонстрировано на рисунке 20.

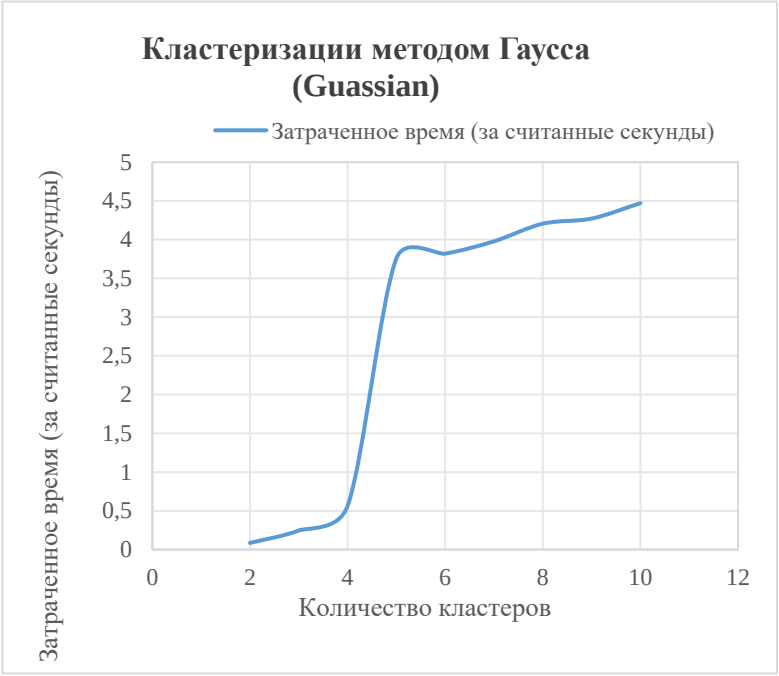

Рисунок 20. Зависимость времени от количества кластеров при использовании алгоритма кластеризации методом Гаусса (Guassian)

На рисунке 20 выше видно увеличение времени выполнения алгоритма кластеризации методом Гаусса (Guassian) в зависимости от увеличения количества кластеров.

6.3.4. Прошедшее время создания кластеров с использованием алгоритма иерархической кластеризации (Hierarchical clustering)

Алгоритм иерархической кластеризации работает по принципу, согласно которому каждый элемент набора данных принадлежит отдельному кластеру, а затем он начинает устанавливать связь между элементами набора данных, находящимися в разных кластерах. Это делает процесс создания кластеров набора данных более сложным и трудоемким, что можно наблюдать в таблице 16.

Таблица 16

Время, затраченное на различные количества кластеров с использовании иерархической кластеризации (Hierarchical clustering)

	Количество кластеров	Затраченное время (за считанные секунды)
Иерархической кластеризации (Hierarchical clustering)	2	17.047294
	3	17.556769
	4	18.008661
	5	16.037917
	6	19.1021826
	7	22.1230242
	8	23.0476956
	9	28.7997632
	10	30.2370962

Из приведенной выше таблицы 16 видно, что время, затрачиваемое иерархической кластеризации (Hierarchical clustering) на объединение набора данных в две группы, очень велико и продолжает увеличиваться по мере увеличения количества кластеров для набора данных. Это хорошо видно на рисунке 21.

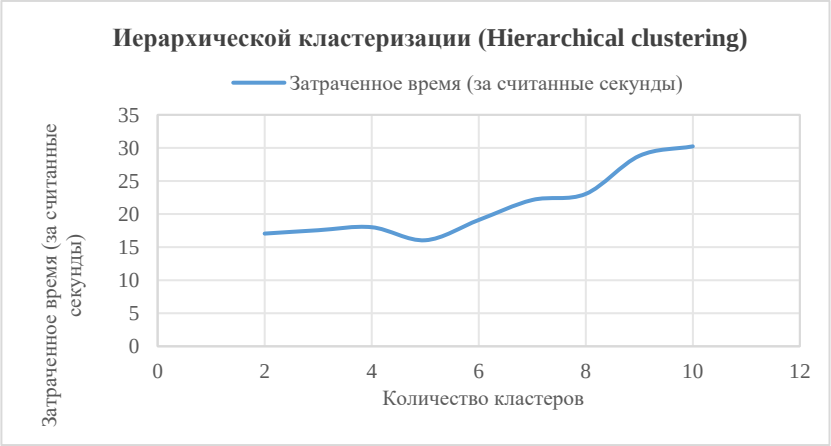

Рисунок 21. Зависимость времени от количества кластеров при использовании алгоритма иерархической кластеризации (Hierarchical clustering)

Как видно из рисунка 21 выше, время выполнения алгоритма иерархической кластеризации (Hierarchical clustering) очень велико по сравнению с другими алгоритмами, использовавшимися ранее для кластеризации набора данных.

Выводы по главе 6

Основываясь на результатах алгоритма кластеризации, можно заметить, что время выполнения K-средних (метод k-средних) и нечетких C-средних (метод C-средних) меньше по сравнению с гауссовой и иерархической кластеризацией. Это связано с их способностью быстро изучать структуру набора данных на этапе обучения машинного обучения по сравнению с гауссовой и иерархической кластеризацией. Способность алгоритма Нечёткая (C Means)-кластеризация (кластеризация методом C-средних) к кластеризации является лучшей среди всех четырёх популярных алгоритмов кластеризации, которые были проанализированы в ходе этой работы. Он делит набор данных на кластеры с почти одинаковым количеством элементов или данных в каждом кластере. Он превосходит другие алгоритмы кластеризации по равномерному разделению элементов набора данных на группы.

ГЛАВА 7. ПРЕДЛАГАЕМАЯ СИСТЕМА, РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЙ И ЭКСПЕРИМЕНТА

Исследование основано на выявлении уникальных характеристик набора данных. Оно сосредоточено на элементах набора данных, которые демонстрируют необычные или редкие характеристики. Эти элементы набора данных обладают некоторыми особыми свойствами, которые необходимо изучить, чтобы правильно понять набор данных и его характеристики. Большая часть исследований, посвящённых набору данных, сосредоточена на изучении характеристик наиболее распространённых элементов набора данных. Это исследование предоставляет уникальную возможность проанализировать и изучить необычные характеристики набора данных. Ниже рассматривается система, разработанная в результате этого исследования.

7.1. Проектирование и схема системы

Выявление редких случаев происходит в соответствии с завершенным процессом. Сначала набор данных разбивается на определенное количество кластеров, а затем внутри каждого кластера выявляются редкие случаи. Весь процесс следует циклу, показанному на рисунке 22.

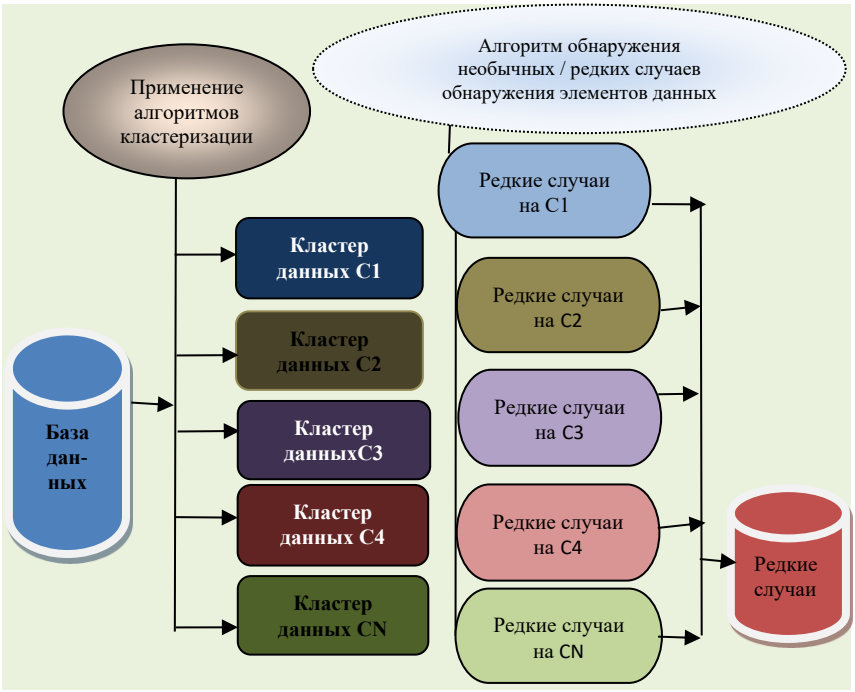

Рисунок 22. Представлена предлагаемая системная схема процесса выявления редких случаев из базы данных

Основная цель этой системы – идентифицировать и отличать редкие элементы данных от обычных элементов данных из базы данных. Она начинается с базы данных, содержащей все обычные и необычные элементы данных вместе взятые. Затем, при применении алгоритма кластеризации, элементы данных разделяются на несколько групп кластеров на основе некоторых критериев сходства, которые определяют процедуру размещения каждого элемента данных в одной из групп кластеров. Затем, на следующем этапе, необычные элементы данных идентифицируются в каждом кластере с использованием другого алгоритма. Таким образом, к концу этой процедуры будет создана новая база данных меньшего размера, содержащая только набор всех редких элементов данных, выявленных на предыдущем этапе. Эта новая созданная база данных может быть использована для анализа скрытых характеристик исходной базы данных.

7.2. Результаты исследований

Исследование содержит несколько выводов, позволяющих идентифицировать редкие или нетипичные элементы данных из набора данных. Такие как выбор подходящей методологии кластеризации для разделения набора данных на различные кластеры, в которых члены каждого кластера демонстрируют характеристики, отличные от членов других кластеров набора данных. Для этого были проанализированы четыре различных метода, и на основе времени выполнения, возможностей кластеризации был выбран метод, который лучше подходит для группировки элементов набора данных.

Затем было исследовано подходящее количество кластеров для разделения набора данных. Для этого набор данных сначала распределяется по двум кластерам, затем по трем кластерам и так далее, вплоть до десяти кластеров. Для выбора подходящего количества кластеров для кластеризации данных сравнивается дисперсия данных в каждой группе данных. Нечёткая (C Means)-кластеризация (кластеризация методом C-средних) минимальной дисперсией рассматривается как соответствующее количество кластеров в соответствии с определением.

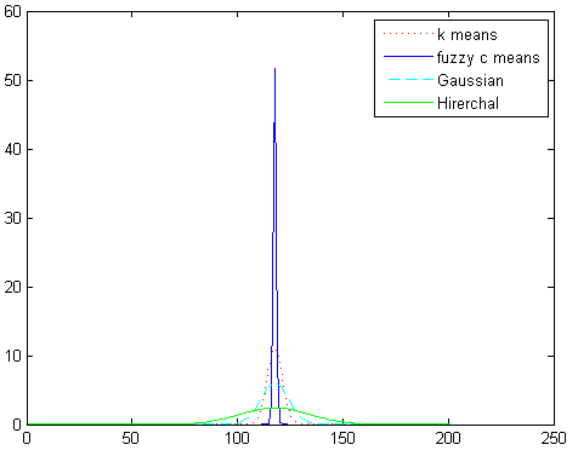

Рисунок 23. Функция плотности вероятности различных методов кластеризации

7.3. Подходы к идентификации редких или необычных элементов данных из набора данных

В ходе этой исследовательской работы были изучены и проанализированы различные алгоритмы для идентификации редких элементов данных из набора данных. На основе анализа и результатов экспериментов предложены и обсуждаются ниже следующие методы для идентификации редких элементов данных из набора данных. За каждой методологией или подходом следуют результаты экспериментов, иллюстрирующие их эффективность.

7.3.1. Подход/ Модель 1

Эта модель основана на распределении набора данных по пяти кластерам с использованием алгоритма кластеризации Нечёткая (C Means)-кластеризация (кластеризация методом С-средних). После кластеризации набора данных в каждом кластере применяется иерархическая кластеризация для идентификации редких элементов данных в каждом кластере. Предложенная модель применяется к проблемной части набора данных. Результаты можно увидеть в следующих таблицах 17 и 18.

Применение Нечёткая (C Means)-кластеризация (кластеризация методом С-средних) в наборе данных

Набор данных разделен на пять кластеров. Каждый кластер содержит элементы данных, имеющие сходные характеристики и отличающиеся от элементов данных других кластеров. Результаты кластеризации подробно представлены в таблице 17.

Таблица 17

Результаты применения Нечёткая (C Means)-кластеризация (кластеризация методом С-средних) при кластеризации данных в пяти кластерах

Номер кластера	Количество элементов данных
Кластер 1	922
Кластер 2	708
Кластер 3	974

Кластер 4	839
Кластер 5	350
Общее качество	3793

Из таблицы 17 можно заметит, что Нечёткая (C Means)-кластеризация (кластеризация методом С-средних) разделает элементов набора данных в кластеры с почти одинаковым количеством данных элемент. Следующим этапом этой модели является применение иерархической кластеризации по отдельности ко всем кластерам, созданным на первом этапе этого подхода. Результаты иерархической кластеризации для кластеров представлены в таблице 18.

Таблица 18

Результаты иерархической кластеризации для выявления редких элементов данных из пяти кластеров

Номер кластера	**Количество элементов данных**	**Необычные или редкие элементы данных (X)**	**Обычные элементы базы данных**
Кластер 1	922	1	921
Кластер 2	708	7	701
Кластер 3	974	1	973
Кластер 4	839	1	838
Кластер 5	350	2	348
Общее качество	3793	12	3781

Из приведенной выше таблицы 18 можно увидеть редкие элементы данных из каждого кластера. Эти редкие элементы данных на графиках кластеризации находятся на границе каждого кластера и обладают характеристиками, отличными от других элементов данных-членов кластера. Кроме того, извлекается идентификатор пациента с необычным элементом данных, как показано ниже в таблице 19.

Таблица 19

Идентификаторы элементов данных, отнесенных к редким случаям

Выявлены необычные или редкостные элементы данных	
Номер кластера	**Идентификация необычных или редких элементов данных**
Кластер 1	1160
Кластер 2	7890,2241,5711,6786,6749,2290,6266
Кластер 3	2562
Кластер 4	6950
Кластер 5	1156, 5063

7.3.2. Подход/ Модель 2

Это самый простой подход, при котором вместо группировки данных в группы и последующего выявления редких элементов данных мы напрямую применяем иерархическую кластеризацию к проблемной части набора данных. Результаты этого подхода подробно описаны в таблице 20.

Таблица 20

Результаты идентификации редких элементов данных непосредственно с помощью иерархической кластеризации

Выявлены необычные или редкостные элементы данных			
Общее качество	**Необычные или редкие элементы данных (Y)**	**Идентификация необычных или редких элементов данных**	**Обычные элементы базы данных**
3793	2	6142, 9720	3782

Из приведенной выше таблицы 20 видно, что кластеризация по наследству позволяет идентифицировать 2 редких элемента данных из набора данных. В нем также указывается, что применение иерархической кластеризации непосредственно к набору данных для идентификации редких элементов данных не дало желаемых результатов.

7.3.3. Подход/ Модель 3

В рамках этого подхода иерархическая кластеризация напрямую применяется не к проблемной части набора данных, а к результирующей части набора данных, чтобы напрямую идентифицировать редкие случаи. Результаты этого метода подробно описаны в таблице 21.

Таблица 21

Результаты идентификации редких элементов данных с помощью иерархической кластеризации

Общее качество	**Необычные или редкие элементы данных (Z)**	**Идентификация необычных или редких элементов данных**	**Обычные элементы базы данных**
3793	3	1156,1160,5063	2892

Из приведенной выше таблицы 20 можно заметить, что аналогично подходу 2 в данном случае из набора данных идентифицируются несколько редких элементов данных.

7.3.4. Подход/ Модель 4

Этот подход в чем-то похож на первый подход и отличается от него в чем-то другом. В рамках этого подхода данные группируются в пять групп путем применения алгоритма нечетких Си-средних к проблемной части набора данных. Затем вместо того, чтобы выбирать выходные данные для каждого идентификатора наблюдения с целью идентификации редких элементов набора данных, снова используйте все атрибуты проблемной части для идентификации редких элементов набора данных при применении иерархической кластеризации. Результаты этого метода подробно представлены в таблице 21.

Таблица 21

Результаты идентификации редких элементов данных с помощью подхода 4

Номер кластера	Количество элементов данных	Необычные или редкие элементы данных (W)	Идентификация необычных или редких элементов данных	Обычные элементы базы данных
Кластер 1	922	1	5302	921
Кластер 2	708	4	8669,2285,7195,3325	704
Кластер 3	974	1	939	973
Кластер 4	839	2	8456,2771	838
Кластер 5	350	2	6142,9720	348

Исходя из этого подхода, мы можем заметить, что *Y* является подмножеством *W*.

$$Y \subset W$$

Это означает, что редкие элементы данных, идентифицированные как Y в ходе подхода 2, являются частью редких элементов данных, идентифицированных в ходе подхода 4.

7.4. Обсуждение и результаты

Из приведенных выше подходов к идентификации редких элементов данных в наборе данных было замечено, что подход № 1 является лучшим и позволяет идентифицировать максимальное количество редких элементов данных из набора данных по сравнению с другими подходами. Подход 3 наименее эффективен при минимальном количестве выявленных редких случаев, и ни один из них не относится ни к одному из других подходов. Подход 3 рассматривается

как подмножество подхода 1, поскольку он содержит те же редкие элементы данных, что и подход 1, и проиллюстрирован на рисунке 23.

$$Z \subset X$$

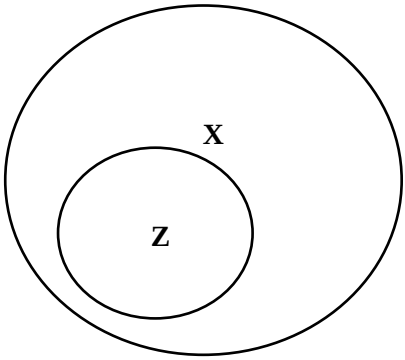

Рисунок 23. Редкие элементы данных, идентифицированные с помощью подхода 3, являются частью редких элементов данных, идентифицированных с помощью подхода 1

Редкие элементы данных, идентифицированные из набора данных подхода 1 и подхода 2, полностью отличаются друг от друга, что означает, что ни один из них не является подмножеством других.

$$X \not\subset Y \ \& \ Y \not\subset X$$
$$X \neq Y$$

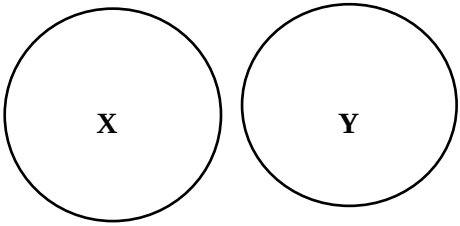

Рисунок 24. Редкие элементы данных, идентифицированные с помощью метода оценки 1 и подхода 2, совершенно различны

Аналогично, редкие элементы данных, идентифицируемые набором данных с использованием подхода 2 и подхода 3, различаются, что означает, что они также не являются подмножествами друг друга.

$$Y \not\subset Z \;\&\; Z \not\subset Y$$

$$Y \neq Z$$

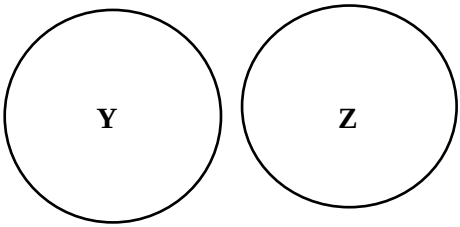

Рисунок 25. Редкие элементы данных, идентифицированные с помощью подхода 2 и подхода 3, совершенно различны

Проанализировав результаты всех четыре подходов, мы можем сделать вывод, что

$$Y \subset W$$
$$Z \subset X$$

7.5. Выводы

В результате сравнения всех четырёх методик кластеризации, реализованных в рамках этой исследовательской работы с целью кластеризации нашего набора данных, были получены следующие наблюдения.

● K-Means (Метод K-средних) кластеризация – это самый простой и эффективный способ кластеризации данных. Он хорошо группирует данные. Однако гораздо сложнее идентифицировать отклонения результатов от данных, поскольку основной целью кластеризации с использованием метода K-Means (Метод k-средних значений) является группировка данных, а центроиды каждого кластера всегда пытаются приблизить данные к ним, используя методологии,

основанные на расстоянии. С увеличением числа кластеров затраченное время уменьшается, и становится легче разделять данные на все более мелкие группы.

- Нечёткая (C Means)-кластеризация (кластеризация методом С-средних) означает, что кластеризация всегда берет данные и представляет их в виде каждого элемента данных, имеющего определенную степень принадлежности к каждому кластеру. Это упрощает группирование данных в кластеры, показывающие принадлежность данных к каждой группе. Однако элемент данных будет считаться принадлежащим к тому кластеру, для которого он имеет более высокую степень принадлежности. При увеличении числа кластеров очень важно сопоставлять данные. Для кластеризации набора данных требуется значительно больше времени, чем для алгоритма Нечёткая (C Means)-кластеризация (кластеризация методом С-средних). Однако он гораздо эффективнее при группировании данных в группы с одинаковой частотой с низким значением дисперсии. Это считается лучшим методом из всех, когда речь идет о группировке данных в кластеры.

- Алгоритм гауссовой кластеризации хорошо работает для распределения данных по различным гауссианам и последующего вычисления степени их правдоподобия для каждого кластера. Здесь данные присваиваются каждому кластеру на основе степени их правдоподобия для каждого кластера. Однако он все еще недостаточно эффективен для идентификации редких или необычных элементов данных из набора данных. В отличие от K-средних (метод k-средних значений) и Нечёткая (C Means)-кластеризация (кластеризация методом С-средних), здесь затраченное время увеличивается по мере увеличения числа кластеров.

- Алгоритм иерархической кластеризации работает совершенно иначе, чем остальные алгоритмы кластеризации. Он начинается с присвоения каждому элементу данных отдельного кластера. Затем кластеры, расположенные близко друг к другу, объединяются в единый кластер, и этот процесс продолжается до тех пор, пока у нас не будет

необходимого количества кластеров. Это подход «снизу вверх». В данном случае необычный элемент данных или кластер данных добавляется в стандартную иерархию кластеризации данных с большим опозданием из-за их удалённости от обычных элементов данных. В данном случае разница во времени, затраченном на увеличение или уменьшение количества кластеров, не так уж и мала. Он распознает редкие элементы данных очень рано, с небольшим количеством кластеров, и поддерживает их с увеличением количества кластеров. Однако, в целом, затраченное время намного превышает результат применения трёх алгоритмов кластеризации.

7.6. Будущая работа

Редкие данные, выявленные в ходе исследовательской работы, планируется применить или ввести в систему CBR [56], чтобы получить надлежащие рекомендации по послеоперационному лечению. Применение этой исследовательской работы к реальным жизненным проблемам делает дальнейшую работу над ней более интересной и практичной. Пригодность этих данных и результаты, касающиеся обнаружения редких элементов данных из набора данных в рамках этой исследовательской работы, также могут быть проанализированы с помощью комментариев медицинского специалиста об эффективности и точности результатов.

Основываясь на комментариях, система может быть улучшена, кроме того, эта автономная система может быть преобразована в онлайн-систему, где каждый новый набор данных при поступлении в систему сначала сравнивается с уже сохранёнными результатами или изученным поведением, и на основе его значений и характеристик он может быть классифицирован как часть обычных данных. предметы или редкие данные. В будущем это исследование будет распространено на набор данных, не связанных с медицинскими исследованиями.

ПРИЛОЖЕНИЯ

Фрагмент программного кода

```
function varargout = testGUI(varargin)

% TESTGUI M-file for testGUI.fig
%      TESTGUI, by itself, creates a new TESTGUI or raises the existing
%      singleton*.
%
%      H = TESTGUI returns the handle to a new TESTGUI or the handle to
%      the existing singleton*.
%
%      TESTGUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in TESTGUI.M with the given input arguments.
%
%      TESTGUI('Property','Value',...) creates a new TESTGUI or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before testGUI_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to tezstGUI_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help testGUI

% Last Modified by GUIDE v2.5 29-May-2024 20:42:37

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @testGUI_OpeningFcn, ...
                   'gui_OutputFcn',  @testGUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
```

```
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before testGUI is made visible.
function testGUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to testGUI (see VARARGIN)

% Choose default command line output for testGUI
handles.output = hObject;

 % Update handles structure
handles.inputXml = 0;
handles.outputXml = 0;
%
set(hObject,'toolbar','figure');
guidata(hObject, handles);

% UIWAIT makes testGUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
```

```
% --- Outputs from this function are returned to the command line.
function varargout = testGUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgr
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in btnSelectFile. For Selecting File
```

```
% --- Executes on button press in btnSelectFile. For Selecting File
function btnSelectFile_Callback(hObject, eventdata, handles)
% hObject    handle to btnSelectFile (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[fileName,pathName,filterIndex] = uigetfile('*.xls', 'Pick an Excel-file');
if fileName ~= 0
    set(handles.edit1, 'String', strcat(pathName,fileName));
    %    data = xlsread(strcat(pathName,fileName),'sheet1','a1:b350');
    data = xlsread(strcat(pathName,fileName));
    set(handles.inputXml,'UserData',data);
    [row,col] = size(data);
    dataToCluster = data(:,2:col);
    [row,col] = size(dataToCluster);
    if col == 1
      % axes(handles.axes1);
        hold off;
        plot(linspace(1,row,row),dataToCluster(:,1)','ro');
        title('Input data');
        %axis([0 1 0 1]);
        hold on;
    end
    if col == 2
       % axes(handles.axes1);
        hold off;
        plot(dataToCluster(:,1)',dataToCluster(:,2)','ro');
        title('Input data');
        %axis([0 1 0 1]);
        hold on;
    end
    if col >2
        set(handles.edit1, 'String', 'Not possible to plot');
    end
else
    set(handles.edit1, 'String', 'No file selected');
```

```
% --- Executes on button press in btnSelectFile. For Selecting File
function btnSelectFile_Callback(hObject, eventdata, handles)
% hObject    handle to btnSelectFile (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[fileName,pathName,filterIndex] = uigetfile('*.xls', 'Pick an Excel-file');
if fileName ~= 0
    set(handles.edit1, 'String', strcat(pathName,fileName));
    %    data = xlsread(strcat(pathName,fileName),'sheet1','a1:b350');
    data = xlsread(strcat(pathName,fileName));
    set(handles.inputXml,'UserData',data);
    [row,col] = size(data);
    dataToCluster = data(:,2:col);
    [row,col] = size(dataToCluster);
    if col == 1
      % axes(handles.axes1);
        hold off;
        plot(linspace(1,row,row),dataToCluster(:,1)','ro');
        title('Input data');
        %axis([0 1 0 1]);
        hold on;
    end
    if col == 2
       % axes(handles.axes1);
        hold off;
        plot(dataToCluster(:,1)',dataToCluster(:,2)','ro');
        title('Input data');
        %axis([0 1 0 1]);
        hold on;
    end
    if col >2
        set(handles.edit1, 'String', 'Not possible to plot');
    end
else
    set(handles.edit1, 'String', 'No file selected');
```

```
% handles    structure with handles and user data (see GUIDATA)
%gets the selected option
% axes(handles.axes1);
noOfCluster = get(handles.noOfCluster,'Value')+1;
data = get(handles.inputXml,'UserData');
[row,col] = size(data);
dataToCluster = data(:,2:col);
[row,col] = size(dataToCluster);
switch get(handles.chClusterName,'Value')
    case 1
        title('K-mean clustering');
        tic;
        [cidx, ctrs] = kmeans(dataToCluster, noOfCluster);
        toc;
        if(col == 1)
            plotCluster(cidx,ctrs,[linspace(1,row,row)',dataToCluster],noOfClust
        end
        if(col == 2)
            plotCluster(cidx,ctrs,dataToCluster,noOfCluster);
        end
        if col >2
            set(handles.edit1, 'String', 'Clustering is done');
            for i = 1 : noOfCluster
                frequency(i) = length(data(cidx==i,2));
            end
            display(frequency);
           %
           %
           id = data(:,1);
           %
           result = [id,cidx];
          %
          set(handles.outputXml,'UserData',result);
            %showStatistics(cidx,ctrs,dataToCluster,noOfCluster);
        end
```

```
case 2
    title('Fuzzy C-mean clustering');
    data = get(handles.inputXml,'UserData');
    tic;
    [center,U,obj_fcn] = fcm(dataToCluster,noOfCluster);
    toc;
    maxU = max(U);
    cidx = zeros(length(dataToCluster),1);
    for i = 1:noOfCluster
        index = find(U(i,:) == maxU);
        for j = 1:length(index)
            cidx(index(j)) = i;
        end
    end

    if(col == 1)
        plotCluster(cidx,center,[linspace(1,row,row)',dataToCluster],noOfClu
    end
    if(col == 2)
        plotCluster(cidx,center,dataToCluster,noOfCluster);
    end
    if col >2
        set(handles.edit1, 'String', 'Clustering is done');
        for i = 1 : noOfCluster
            frequency(i) = length(data(cidx==i,2));
        end
        display(frequency);
        %showStatistics(cidx,center,dataToCluster,noOfCluster);
    end
    id = data(:,1);
    result = [id,cidx];
    %
    set(handles.outputXml,'UserData',result);
case 3
    title('Gaussian Mixture Models clustering');
```

```
case 3
    title('Gaussian Mixture Models clustering');
    tic;
    options = statset('Display','final');
    gm = gmdistribution.fit(dataToCluster,noOfCluster,'Options',options);
    idx = cluster(gm,dataToCluster);
    toc;
    center = data(1:noOfCluster,:);

    if(col == 1)
        plotCluster(idx,center,[linspace(1,row,row)',dataToCluster],noOfClus
    end
    if(col == 2)
        plotCluster(idx,center,dataToCluster,noOfCluster);
    end
    if col >2
        set(handles.edit1, 'String', 'Clustering is done');
        for i = 1 : noOfCluster
            frequency(i) = length(data(idx==i,2));
        end
        display(frequency);
        %showStatistics(idx,center,dataToCluster,noOfCluster);
    end
    id = data(:,1);
    result = [id,idx];
    %
    set(handles.outputXml,'UserData',result);
case 4
    title('Hierarchical clustering');
    tic;
    Y = pdist(dataToCluster);
    Z = linkage(Y);
    T = cluster(Z,'maxclust',noOfCluster);
    toc;
    idx = T;
```

```
case 4
    title('Hierarchical clustering');
    tic;
    Y = pdist(dataToCluster);
    Z = linkage(Y);
    T = cluster(Z,'maxclust',noOfCluster);
    toc;
    idx = T;
    center = data(1:noOfCluster,:);

    if(col == 1)
        plotCluster(idx,center,[linspace(1,row,row)',dataToCluster],noOfClus
    end
    if(col == 2)
        plotCluster(idx,center,dataToCluster,noOfCluster);
    end
    if col >2
        set(handles.edit1, 'String', 'Clustering is done');
        for i = 1 : noOfCluster
            frequency(i) = length(data(idx==i,2));
        end
        display(frequency);
        %showStatistics(idx,center,dataToCluster,noOfCluster);
    end
    id = data(:,1);
    result = [id,idx];

 set(handles.outputXml,'UserData',result);
case 6
    [row,col] = size(dataToCluster);
    if col == 1
       axes(handles.axes1);
        hold off;
        plot(linspace(1,row,row),dataToCluster(:,1)','ko');
        title('Input data');
```

```
    case 6
        [row,col] = size(dataToCluster);
        if col == 1
           axes(handles.axes1);
            hold off;
            plot(linspace(1,row,row),dataToCluster(:,1)','ko');
            title('Input data');
            %axis([0 1 0 1]);
            hold on;
        end
        if col == 2
            axes(handles.axes1);
            hold off;
            plot(dataToCluster(:,1)',dataToCluster(:,2)','ko');
            title('Input data');
            %axis([0 1 0 1]);
            hold on;
        end
        if col >2
            set(handles.edit1, 'String', 'Not possible to plot');
            for i = 1 : noOfCluster
                frequency(i) = length(data(cidx==i,2));
            end
            display(frequency);
        end
        hold on;
    case 7
        title('Hierarchical clustering');
        data = get(handles.inputXml,'UserData');
        X = data(:,2:col);
        Y = pdist(X);
        Z = linkage(Y);
        dendrogram(Z);
    otherwise
end
```

```
    case 6
        [row,col] = size(dataToCluster);
        if col == 1
           axes(handles.axes1);
            hold off;
            plot(linspace(1,row,row),dataToCluster(:,1)','ko');
            title('Input data');
            %axis([0 1 0 1]);
            hold on;
        end
        if col == 2
            axes(handles.axes1);
            hold off;
            plot(dataToCluster(:,1)',dataToCluster(:,2)','ko');
            title('Input data');
            %axis([0 1 0 1]);
            hold on;
        end
        if col >2
            set(handles.edit1, 'String', 'Not possible to plot');
            for i = 1 : noOfCluster
                frequency(i) = length(data(cidx==i,2));
            end
            display(frequency);
        end
        hold on;
    case 7
        title('Hierarchical clustering');
        data = get(handles.inputXml,'UserData');
        X = data(:,2:col);
        Y = pdist(X);
        Z = linkage(Y);
        dendrogram(Z);
    otherwise
end
```

```
    for i = 1 : noOfCluster
        rnames(i) = {strcat('Cluster ',num2str(i))};
        %plot(X(cidx==i,1),X(cidx==i,2),'Color', ColorSet(i,:));
        x(i) = plot(data(cidx==i,1),data(cidx==i,2),'o','Color', ColorSet(i,:));
        hold on;
        %plot(X(ctrs==i,1),X(ctrs==i,2),'*','Color', ColorSet(i,:));
        distance_max = dist(max_value, ctrs(i));
        distance_min = dist(min_value, ctrs(i));
        meanValue = mean(data(cidx==i,2));
        stdValue = std(data(cidx==i,2));
        frequency = length(data(cidx==i,2));
        percent = frequency/noOfDataPoint*100;
       properties(i,:) = [frequency percent meanValue stdValue distance_max dist
    end
    legend(x,rnames);
    %f = figure('Position',[200 200 650 200]);
    %format ('shortG')
    cnames = {'Frequency','% of data','Mean','StdDev','Distance_max','Distance_m
    display(cnames);
    display(properties);
    get(0,'format')
    %sprintf('%6.1f ', properties) ;
    %shortg(properties);
    %t = uitable('Parent',f,'Data',properties,'ColumnName',cnames,...
     %           'RowName',rnames,'Position',[20 20 600 150]);

function showStatistics(cidx,ctrs,data,noOfCluster)


    %ColorSet = varycolor(noOfCluster);
    properties = zeros(noOfCluster,6);
    noOfDataPoint = length(data);
    max_value = max(data);
    %display (max_value);
    min_value = min(data);
```

```
        %x(i) = plot(data(cidx==i,1),data(cidx==i,2),'o','Color', ColorSet(i,:))
        %hold on;
        %plot(X(ctrs==i,1),X(ctrs==i,2),'*','Color', ColorSet(i,:));
        distance_max = sum((max_value-ctrs(i,:)).^2).^0.5;
        %display(distance_max);
        %return;
        %distance_min = dist(min_value, ctrs(i,:));
        distance_min = sum((min_value-ctrs(i,:)).^2).^0.5;
        meanValue = mean(data(cidx==i,:));
        stdValue = std(data(cidx==i,:));
        frequency = length(data(cidx==i,:));
        percent = frequency/noOfDataPoint*100;
        %display (meanValue);
        sprintf('%6.1f ', meanValue) ;
        %properties(i,:) = [frequency percent meanValue stdValue distance_max di
    end
    %legend(x,rnames);
    %f = figure('Position',[200 200 650 200]);
    %cnames = {'Frequency','% of data','Mean','StdDev','Distance_max','Distance_
    %display(cnames);
    %display(properties);
    %t = uitable('Parent',f,'Data',properties,'ColumnName',cnames,...
              % 'RowName',rnames,'Position',[20 20 600 150]);


% --- Executes on button press in btnSaveXls.
function btnSaveXls_Callback(hObject, eventdata, handles)
% hObject    handle to btnSaveXls (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
result = get(handles.outputXml,'UserData');
[fileName,pathName,FilterIndex] = uiputfile('*.xls');
if FilterIndex ~= 0
    xlswrite(strcat(pathName,fileName),result);
end
```

СПИСОК ЛИТЕРАТУРЫ

1. Barnett V., Lewis T. Outliers in Statistical Data. John Wiley, 1994.
2. Johnson R. Applied Multivariate Statistical Analysis. Prentice Hall, 1992.
3. Ahmed M.U., Begum S., Funk P., Xiong N (2011). A Multi-Module Case Based Biofeedback System for Stress Treatment // Artificial Intelligence in Medicine, 51(2). Pp. 107–115.
4. Begum S., Ahmed M.U., Funk P., Xiong N (2006). A Case-Based Decision Support System for Individual Stress Diagnosis Using Fuzzy Similarity Matching // Computational Intelligence (CI), 25(3). Pp. 180–195.
5. Аббаси М.М. Анализ влияния отрицаний на текст с использованием кластеризации и колеса эмоций Плутчика // Научно-технический вестник Поволжья. – 2019. – 12 (66). С. 23–28.
6. Beltiukov A.P., Abbasi M.M (2019). Logical analysis of Emotions in Text from Natural language // Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki. – Izhevsk, 1 (29). Pp. 106–116.
7. Caussinus H., Roiz A (1990). Interesting Projections Of Multidimensional Data By Means Of Generalized Component Analysis // In Compstat 90. Pp. 121–126, Heidelberg: Physica.
8. Rousseeuw P., Leory A (1987). Robust Regression & Outlier Detection // Wiley Seriesin Probability and Statistics.
9. Jin W., Tung A., Han J (2001). Mining Top-N Local Outliers In Large Databases // In Proceedings of the 7th International Conference on Knowledge Discovery and Data-mining (KDD01), San Francisco.
10. Barbara D., Chen P (2000). Using The Fractal Dimension To Cluster Datasets // In Proceedings of the .ACM KDD 2000. Pp. 260–264.
11. Shekhar S., Lu C., Zhang P (2002). Detecting Graph-Based Spatial Outlier // IntelligentData Analysis: An International Journal, 6(5). Pp. 451–468.
12. Hu T., Sung S (2005). Detecting Pattern-Based Outliers // Pattern Recognition Letters, 24. Pp. 3059–3068.

13. Davies L., Gather U (1993). The Identification Of Multiple Outliers // Journal of the American Statistical Association, 88(423). Pp. 782–792.

14. Hampel F. (1971). A general qualitative definition of robustness // Annals of Mathematics Statistics 42(1). Pp. 1887–1896.

15. Tukey J.W (1977). Exploratory Data Analysis // Addison-Wesley.

16. Acuna E., Rodriguez C (2004). Meta Analysis Study Of Outlier Detection Methods In Classification // In Proceedings of the IPSI 2004, Venice.

17. Hadi A. (1992). Identifying Multiple Outliers In Multivariate Data // Journal of the Royal Statistical Society. Series B, 54 (1). Pp. 761–771.

18. Knorr E., Ng R (1997). A Unified Approach For Mining Outliers // In Proceedings of the Knowledge Discovery KDD. Pp. 219–222.

19. Ramaswamy S., Rastogi R., Shim K (2000).Efficient Algorithms For Mining Outliers From Large Data Sets // In Proceedings of the ACM SIGMOD International Conference on Management of Data, Dalas, TX.

20. Gustavo H., Ye W (2010). Distance Based Outlier Detection: Consolidation and Renewed Bearing // 36th International Conference on Very Large Data Bases, September 1317, 2010, Singapore. Proceedings of the VLDB Endowment, 3(2), VLDB Endowment 21508097/10/09.

21. Markus M., Hans-Peter K., Raymond T., Jörg S (2000). LOF: Identifying Density-Based Local Outliers // ACM 2000 1-58113-218-2/00/05.

22. Elio L., Edgar A (2005). Parallel Algorithms For Distance-Based & Density-Based Outliers // Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05) 1550-4786/05.

23. Knorr E., Ng R., Tucakov V (2000). Distance-Based Outliers: Algorithms & Applications // VLDB Journal: Very Large Data Bases, 8(3). Pp. 237–253.

24. Ng R., Han J (1994). Efficient & Effective Clustering Methods For Spatial Data Mining // In the Proceeding of the 20th International Conference on Very Large Databases. Morgan and Kaufmann Publishers, San Francisco, 8(3). Pp.44–155.

25. Rocke D., Woodruff D (2002). Computational Connections Between Robust Multivariate Analysis & Clustering // In COMPSTAT 2002 Proc. Of the Computational Statistics, Wolfgang H.

26. Zaisheng D., Liusheng H., Youwen Z., Wei Y (2010). Privacy Preserving Density-Based Outlier Detection // International Conference on Communications and Mobile Computing, 978-0-7695-3989-8/10 2010, DOI 10.1109/CMC.2010.274

27. Murtagh F., Raftery A.E (1984). Fitting Straight Lines To Point Patterns // Pattern Recognition, 17 (1). Pp. 479–483.

28. Banfield J.D., Raftery A.E (1993). Model-Based Gaussian & Non-Gaussian Clustering // Biometrics, 49 (1). Pp.803–821.

29. Raftery A.E (1993). Time Series and Image Analysis // Department of Statistics, University of Washington, Transitions from ONR Contract N00014-91-J-1074.

30. Andrii Y. S (2028). Using The Agglomerative Method Of Hierarchical Clustering As A Data Mining Tool In Capital Market // International Journal of Information Theories & Applications, 15 (1).

31. Rahila H. S., Raghuwanshi M., Anil N. J (2008). Genetic Algorithm Based Clustering: A Survey // First International Conference on Emerging Trends in Engineering and Technology, 978-0-7695-3267-7/08.

32. Hua J., Shenghe Y., Jing L., Fengqin Y., Xin H (2010). A clustering algorithm with K-harmonic means clustering // Expert Systems with Applications 37 (1). Pp. 8679–8684. 0957-4174/2010 ,doi:10.1016/j.eswa. 2010.06.061.

33. Hongyang L., Jia H (2009). The Application of Dynamic K-means Clustering Algorithm in the Center Selection of RBF Neural Networks // 2009 Third International Conference on Genetic and Evolutionary Computing, 978-0-7695-3899-0/09, DOI 10.1109/2009.

34. Yiu-Ming C (2003). K-Means: A new generalized k-means clustering algorithm // Pattern Recognition Letters 24 (2003) 2883–2893, 0167-8655/2003, doi:10.1016/S0167-8655(03)00146-6.

35. Fang W., Zhang Q. J (1995). An Improved K-Means Clustering Algorithm & Application To Combined Muetic-C Odeu OK // MLP Neural Network Speech Recognition, Cecuccgei ’95, 0-7803-2766-7-9/951.

36. Dingxi Q (2010). A Comparative Study Of The K-Means Algorithm & Thenormal Mixture Model For Clustering: Bivariate Homoscedastic Case // Journal of Statistical Planning and Inference 140 (1). Pp. 1701–1711.

37. Аббаси М.М., Бельтюков А.П. Изучение доступных систем для анализа эмоций, извлеченных из текста, и обеспечения механизма для улучшения взаимодействия человека с машиной // Интеллектуальные системы в производстве. – 2019. –17 (4). Http://Dx.Doi.Org/10.22213/2410-9304-2019-4-53-62.

38. Lin Z., Fu-Lai C., & Shitong W (2008). Generalized Fuzzy C-Means Clustering Algorithm With Improved Fuzzy Partitions // Transaction On Systems, Man & Cybernetics- Part B: Cybernetics, 39(3).

39. Yuanping Z (2010). An Efficient Supervised Clustering Algorithm Based on Neural Networks// 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 978-1-4244-6542-2/ 2010.

40. Jingwei L., Meizhi X (2008) . Kernelized Fuzzy Attribute C-Means Clustering Algorithm // Fuzzy Sets and Systems 159 (1). Pp. 2428–2445, 0165-0114/2008 ,doi:10.1016/j.fss.2008.03.018.

41. Somporn C., Chidchanok L., Peraphon S., Suchada S (2010) Fuzzy C-Means: A Statistical Feature Classification of Text and Image Segmentation Method.

42. Jing X., YuPing Y., Jun Z. Yong T (2010). A quantum-inspired genetic algorithm for k-means clustering // Expert Systems with Applications, 37 (1). Pp. 4966–4973.

43. Yu-C., Lawrence W (2001). Theory and Methodology, Genetic Clustering Algorithm // European Journal Of Operational Research, 135 (2001). Pp. 413–427.

44. Ujjwal M (2000). Genetic algorithm-based clustering technique // Pattern Recognition, 33 (2000). Pp. 1455–1465.

45. Gautam G., Chaudhuri B. A novel genetic algorithm for automatic clustering // Pattern Recognition Letters. – 2004. – 25. – Pp. 173–187.

46. Stefan B., Hans-Peter K., Martin P. Multi-Step Density-Based Clustering // Knowledge and Information Systems (KAIS). – 2006. – 9 (3).

47. Stefan B., Hans-Peter K., Martin P (2006). Parallel Density-Based Clustering of Complex Objects // W.K. Ng et al. (eds.): PAKDD 2006, LNAI 3918. Pp.179–188.

48. Christian B., Karin K., Hans-Peter K., Peer K. Density Connected Clustering with Local Subspace Preferences // In Proceedings of the 4th IEEE Int. Conf. on Data Mining (ICDM 04). – 2004. – Brighton UK.

49. Mihael A., Markus M., Hans-Peter K., Jörg S. (1999). OPTICS: Ordering Points To Identify the Clustering Structure // In Proceedings of the ACM SIGMOD'99 Int. Conf. on Management of Data, Philadelphia PA, 1999.

50. Naounori U., Ryohei N. Split & Merge EM Algorithm For Improving Gaussian Mixture Density Estimates // Journal of VLSI Signal Processing. – 2000. – 26(1). Pp. 133–140.

51. Ananth S., Venkata R. R. Parameter Tying and Gaussian Clustering for Faster, Better & Smaller Speech Recognition // Sponsored by DARPA through the Naval Command and Control Ocean Surveillance Center under contract. – 2010. – N6600194C6048.

52. Rougui J. E., Rziza M., Aboutajdine D (2006). Fast Incremental Clustering Of Gaussian Mixture Speaker Models For Scaling Up Retrieval In On-Line Broadcast, 142440469X/ 06/ 2006.

53. Zhiwen Y., Hau-San W. Fast Gaussian Mixture Clustering For Skin Detection. – 2006. – 1-4244-0481-9/06/ 2006.

54. Jeyhan K. (1993). An Unsupervised Gaussian Cluster Formation Technique as a Bussgang Blind Deconvolution Algorithm, 0-7803-1254-6/93 G00 .

55. Xin L., Arindam M., Jing Z. Fast Likelihood Computation Using Hierarchical Gaussian Shortlists. – 2010. – 978-1-4244-4296-6/10/2010.

56. Ahmed M.U, Funk P. Case-Based Retrieval System for Post-operative Pain Management, accepted in the International Workshop Case-Based Reasoning CBR 2011, IBaI, Germany, New York/ USA, Editor(s):Petra Perner, September, 2011.

57. Abiodun M. Ikotun, Absalom E. Ezugwu, Laith Abualigah, Belal Abuhaija, Jia Heming, K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data, Information Sciences,Volume 622. – 2023. Pp. 178–210.

58. Amber Abernathy, M. Emre Celebi, The incremental online k-means clustering algorithm and its application to color quantization, Expert Systems with Applications,Volume 207, 2022, 117927, ISSN 0957-4174.

59. Liang Bai, Jiye Liang, Fuyuan Cao, A multiple k-means clustering ensemble algorithm to find nonlinearly separable clusters, Information Fusion, Volume 61, 2020. Pp. 36–47, ISSN 1566-2535.

60. Fatéma Zahra Benchara, Mohamed Youssfi, A new scalable distributed k-means algorithm based on Cloud micro-services for High-performance computing, Parallel Computing, Volume 101, 2021, 102736, ISSN 0167-8191.

61. Patrícia Alves, André Martins, Francisco Negrão, Paulo Novais, Ana Almeida, Goreti Marreiros, Are heterogeinity and conflicting preferences no longer a problem? Personality-based dynamic clustering for group recommender systems, Expert Systems with Applications, Volume 255, Part D, 2024, 124812, ISSN 0957-4174.

62. Delic A, Emamgholizadeh H, Nguyen TN, Ricci F. CHARM: a Group Decision Making Support Chatbot. InCompanion Proceedings of the 29th International Conference on Intelligent User Interfaces 2024 Mar 18. Pp. 7–10.

63. Wang, Y., Qian, J., Hassan, M., Zhang, X., Zhang, T., Yang, C., Zhou, X. and Jia, F., 2024. Density peak clustering algorithms: A review on the decade 2014–2023. Expert Systems with Applications, 238. P. 121860.

64. Ren Y, Pu J, Yang Z, Xu J, Li G, Pu X, Yu PS, He L. Deep clustering: A comprehensive survey. IEEE transactions on neural networks and learning systems. 2024 Jul 4.

65. Tu, Wenxuan, Renxiang Guan, Sihang Zhou, Chuan Ma, Xin Peng, Zhiping Cai, Zhe Liu, Jieren Cheng, and Xinwang Liu. "Attribute-missing graph clustering network." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 14. Pp. 15392–15401. 2024.

66. Daneshfar, F., Soleymanbaigi, S., Yamini, P. and Amini, M.S., 2024. A survey on semi-supervised graph clustering. Engineering Applications of Artificial Intelligence, 133. P. 108215.

Оглавление

Научное издание

Аббаси Мохсин Маншад
Бельтюков Анатолий Петрович

Использование алгоритма машинного обучения (кластеризация) для анализа клинических данных пациентов с целью выявления пациентов с редкими симптомами

Монография

Авторская редакция
Компьютерная верстка: Т.В. Опарина

Подписано в печать 25.09.2025. Формат 60x84/16.
Усл. печ. л. 6,6. Уч. изд. л. 5,9.
Тираж 27 экз. Заказ № 1493.

Издательский центр «Удмуртский университет»
426034, г. Ижевск, ул. Ломоносова, 4Б, каб. 021
Тел.: + 7 (3412) 916-364, E-mail: editorial@udsu.ru

Типография Издательского центра «Удмуртский университет»
426034, г. Ижевск, ул. Университетская, 1, корп. 2.
Тел. 68-57-18