**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное**

**учреждение высшего образования**

**Рязанский государственный радиотехнический университет**

**имени В.Ф. Уткина**

**Кафедра САПР ВС**

**К защите**

Руководитель КП

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.И. Орешков

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ**

по дисциплине

**«ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И МЯГКИЕ ВЫЧИСЛЕНИЯ»**

**Тема**

«Классификация данных с использованием деревьев решений»

Выполнил студент группы 2047M

Халимова А.Р. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата сдачи на проверку, подпись

Руководитель проекта

к.т.н., доцент Орешков В.И. \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

оценка дата защиты, подпись

Рязань 2023 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**Рязанский государственный радиотехнический университет**

**имени В.Ф. Уткина**

**Кафедра САПР ВС**

**ЗАДАНИЕ**

**на курсовое проектирование**

**по дисциплине «Интеллектуальные системы и мягкие вычисления»**

**Студенту Халимовой Алине Равшановне***,* ***гр****.* ***2047М***

**1. Тема проекта: «Классификация данных с использованием деревьев решений»**

**2. Срок сдачи студентом законченного проекта** *23 июня 2023 г.*

**3. Руководитель проекта:** *Орешков Вячеслав Игоревич, к.т.н., доцент кафедры САПР ВС*

**4. Исходные данные к проекту**

*1. Операционная система: Windows XP/7/8/10*

*2. Среда обработки данных: аналитическое приложение Deductor Studio*

**5. Содержание пояснительной записки**

Задание.

Содержание.

Введение.

1. Теоретический материал по теме.
2. Загрузка данных в аналитическое приложение.
3. Построение и обучение деревьев решений.
4. Анализ результатов обучения.

Заключение.

Список использованных источников.

Приложения.

Задание выдано «27» января 2023 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ доцент каф. САПР ВС В.И. Орешков

**Содержание**

[**1.** **Введение** 4](#_Toc138784411)

[**2.** **Теоретический материал по теме** 7](#_Toc138784412)

[2.1. Интеллектуальный анализ данных. 7](#_Toc138784413)

[2.2. Классификация данных с использованием деревьев решений 14](#_Toc138784414)

[**3.** **Загрузка данных в аналитическое приложение** 23](#_Toc138784415)

[**4.** **Построение и обучение деревьев решений** 29](#_Toc138784416)

[**5.** **Анализ результатов обучения ДР.** 33](#_Toc138784417)

[**6.** **Заключение** 38](#_Toc138784418)

[**7.** **Список использованных источников** 39](#_Toc138784419)

[**8.** **Приложение А** 40](#_Toc138784420)

# **Введение**

Естественным результатом развития информационных технологий стал быстрый рост количества данных, накопленных человечеством в электронном виде. Данные являются ценным ресурсом, который хранит в себе большие потенциальные возможности по извлечению полезной аналитической информации. Существует множество таких данных в различных видах. Большая часть доступной для обработки информации не несет для человека какой-либо пользы, поскольку он не в состоянии переработать такое количество сведений. Возникает задача «добычи данных» (Data Mining) или «открытия знаний в базах данных» (Knowledge Discovery in Date), полезных для пользователя, из большого объема необработанных данных различной природы[[1]](#footnote-1). В связи с этим появляется необходимость внедрения средств автоматизации в системы анализа данных, которые способны сократить сроки, повысить качество и эффективность обработки данных.

Технологии интеллектуального анализа данных могут рассматриваться как результат естественного развития информационных технологий. Этапами такого развития стали накопление данных, управление данными (включая хранение и извлечение, а также выполнения транзакций), анализ данных.

Источник данных, анализ которых необходимо выполнить, может быть совершенно любой природы. Для специалиста в области защиты информации выполнение анализа данных, источником которых служат информационные системы и их компоненты, в частности сети, обеспечивающие передачу данных между объектами информационной системы, особенно актуально.

Термин «интеллектуальный анализ данных» (ИАД) происходит от англоязычного Data Mining (DM) – раскопка, разработка данных, который был введён В.Г. Пятецким–Шапиро в 1989 году на одном из семинаров, посвященных технологиям поиска знаний в базах данных (Knowledge Discovery in Databases, KDD), проводимых в рамках Международной конференции по искусственному интеллекту (International Joint Conference on Artificial Intelligence) IJCAI–89[[2]](#footnote-2).

Изначально DM определялось как направление в ИТ, включающее «методы обнаружения в данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах деятельности».

В настоящее время KDD и DM являются взаимозаменяемыми понятиями. С 2011 года появилось понятие – Data Science (наука о данных), которые тоже используется при описании различных технологий поиска знаний в данных.

Ставить полное соответствие между понятиями «интеллектуальный анализ данных» и Data Mining неверно, поскольку ИАД охватывает всю область проблем, связанных с извлечением знаний из данных, в то время как Data Mining больше связывают с аналитикой и интеллектуальным моделированием.

Применение современных практических методов анализа данных и распознавания востребовано в технических и гуманитарных областях, в науке и производстве, бизнесе и финансах.

**Цель работы:** изучение процессов построения и обучения деревьев решений в аналитическом приложении Deductor Studio Academic. Исследование процедур применения деревьев решений в задачах классификации.

**Задачами** работы являются:

1. Изучить методы интеллектуального анализа данных.
2. Изучить метод анализа данных с помощью ДПР.
3. Загрузить данные в аналитическую платформу Deductor Studio
4. Построить и обучить дерево решений в аналитической платформе Deductor Studio.
5. Классифицировать данные, использую деревья решений.
6. Проанализировать результаты обучения.

# **Теоретический материал по теме**

## Интеллектуальный анализ данных.

Анализ данных (АД) – это система подходов и методов, ориентированная на

выявление механизма порождения представленных данных в рамках имеющейся априорной модели этого механизма[[3]](#footnote-3).

Современные технологии анализа данных – новая парадигма процесса исследования данных, основанная на принципах, предложенных Джоном Тьюки:

* Анализ – это способ существования данных.
* Принцип многократного возвращения к одним и тем же данным.
* Принцип множественности возможных моделей.
* Принцип варьирования предпосылок с рассмотрением последствий такого варьирования.
* Принцип множественности результатов и выбора на основе неформальных процедур принятия решений.
* Принцип полного использования эндогенной информации и максимального учета информации экзогенной.

В ряде случаев АД строится и реализуется в соответствии с технологиями искусственного интеллекта.

Изначально инструментом для извлечения знаний был анализ данных с использованием методов прикладной статистики. Но так как не все пользователи обладали достаточным образованием и уровнем навыков для применения данных алгоритмов и методов, необходимо было решить две задачи для упрощения использования:

* максимально автоматизировать процесс извлечения из данных скрытых зависимостей, закономерностей и структур;
* разработать методы, которые были бы просты в использовании и понятны людям, не имеющим специального образования, а результаты легко интерпретируемы и осмысливаемы[[4]](#footnote-4).

Для решения первой задачи стали использовать элементы машинного обучения (МО) (machine learning) обширного раздела искусственного интеллекта (ИИ), использующего алгоритмы и методы математической статистики, теории информации, теории множеств, численной оптимизации и дискретного анализа для решения базовых задач анализа данных: классификации, кластеризации, регрессии и ассоциации.

В рамках второй задачи был сделан упор на использование эвристических моделей, алгоритмы и методы построения которых, хотя и математически не обоснованы, позволяют получить приемлемое решение в большинстве практически значимых случаев.

Эвристические методы полезны для решений задач анализа больших объемов данных, т.к. вместо полного перебора, используются более быстрый алгоритм, хоть и не вполне математически верный. Так же метод используется в получении результата в тех областях, где общего решения задачи не существует (распознавание образов). Поэтому ИАД широко применяет такие методы, как нейронные сети, деревья решений, карты Кохонена, ассоциативные правила и др.

Интеллектуальный анализ данных (ИАД) – исследование данных, использующее методы искусственного интеллекта и ориентированное на придание системе свойств искусственного интеллекта[[5]](#footnote-5).

Понятие «интеллектуальный анализ данных» (ИАД) представляет собой трансформацию англоязычного термина «Data Mining» — раскопка, разработка данных. Считается, что впервые термин «Data Mining» был введен Г. Пятецким-Шапиро в 1989 г. на одном из семинаров, посвященных технологиям поиска знаний в базах данных (Knowledge Discovery in Databases — KDD), проводимых в рамках Международной конференции по искусственному интеллекту (International Joint Conference on Artificial Intelligence) IJCAI-89 в Детройте[[6]](#footnote-6).

В интеллектуальном анализе данных применяется математический анализ для выявления закономерностей и тенденций, существующих в данных. Обычно такие закономерности нельзя обнаружить при традиционном просмотре данных, поскольку связи слишком сложны, или из-за чрезмерного объема данных.

Эти закономерности и тренды можно собрать вместе и определить как модель интеллектуального анализа данных. Модели интеллектуального анализа данных могут применяться к конкретным сценариям, а именно:

Прогнозирование: оценка продаж, прогнозирование нагрузки сервера или времени простоя сервера.

Риск и вероятность: выбор наиболее подходящих заказчиков для целевой рассылки, определение точки равновесия для рискованных сценариев, назначение вероятностей диагнозам или другим результатам.

Рекомендации: определение продуктов, которые с высокой долей вероятности могут быть проданы вместе, создание рекомендаций.

Поиск последовательностей: анализ выбора заказчиков во время совершения покупок, прогнозирование следующего возможного события.

Группирование: разделение заказчиков или событий на кластеры связанных элементов, анализ и прогнозирование общих черт.

ИАД используется только со структурированными данными, т. е. данными, представленными в виде таблиц с типизированными полями, где каждое поле имеет уникальное имя. Если исходные данные не являются структурированными или являются слабоструктурированными, то для аналитической обработки в рамках ИАД они непригодны и требуют специальной процедуры структуризации.

ИАД включает несколько относительно независимых поднаправлений,

связанных с подготовкой данных к аналитической обработке, организации их хранения и визуализации результатов. Подготовка данных (или, как ее чаще называют, предобработка) включает два этапа — трансформацию и очистку данных[[7]](#footnote-7).

Трансформация — это набор процедур преобразования данных к виду, наиболее подходящему с точки зрения используемых методов анализа. К числу таких преобразований относятся: преобразование типов, нормализация, квантование и категоризация значений, разбор составных значений (адреса, ФИО), сортировка, группировка и др. Следует отметить, что трансформация данных должна только изменять их представление, но не внутреннее содержание, чтобы не искажать скрытые в данных зависимости и закономерности, обнаружение которых и является целью их анализа.

Очистка данных — это набор процедур, связанных с исключением из данных различных факторов, мешающих их корректной обработке. Наиболее критичными факторами, снижающими качество данных, являются пропуски (пустые ячейки таблиц), аномальные значения, дубликаты и противоречия, шумы, избыточность или неполнота данных. В большинстве случаев аналитикам приходится работать именно с «грязными», «сырыми» данными, в которых представлен весь спектр проблем, снижающих их качество. Это связано с тем, что разработчики ИС предприятий часто не уделяют достаточно внимания обеспечению и поддержке качества данных, циркулирующих в системе, и не учитывают перспектив их аналитической обработки. Поэтому проблема «грязных» данных является ключевой в ИАД, и ее решение может потребовать в несколько раз больше усилий и времени, чем собственно аналитическая обработка. Тем не менее использовать «грязные» данные нельзя, поскольку на их основе могут быть получены некорректные результаты анализа, которые могут привести к принятию неверных управленческих решений.

Отчасти, проблему качества данных удается решить с помощью специализированных баз данных, называемых хранилищами данных (data warehouse)[[8]](#footnote-8), которые оптимизированы для работы именно с аналитическими приложениями.

Процесс внедрения ИАД в практическую деятельность предприятий и организаций для решения конкретных задач повышения эффективности управления в большинстве случаев достаточно затратный и трудоемкий. Основными проблемами являются отсутствие формальной постановки задачи и стратегии поиска знаний, эвристический характер большинства интеллектуальных моделей, высокая размерность и низкое качество исходных данных[[9]](#footnote-9).

Тем не менее рынок программных средств, реализующих технологии ИАД, стремительно развивался и рос, и к концу первого десятилетия XXI в. достиг объема 7,8 млрд долл. США (с ежегодным ростом 12,1%).

Параллельно с ростом числа доступных ИАД-инструментов росла их сложность для большинства потенциальных пользователей. Поэтому были предприняты попытки разработать единые подходы к реализации ИАД-процессов, представлению данных и интерфейсов для моделирования.

Можно выделить 9 групп программных средств интеллектуального анализа данных.

1. DM-сьюты (DMST — Data Mining Suite Tools) — изначально ориентированы на задачи DM, включают большое число различных методов и алгоритмов анализа и моделирования, поддерживают работу с многомерными структурированными и неструктурированными данными.

2. Пакеты бизнес-аналитики (DMBT — Data Mining Business Tools) — не создавались изначально для решения задач DM, но включают функции интеллектуального анализа, особенно статистические методы в бизнес-приложениях.

3. Математические пакеты (DMМP — Data Mining Mat Package) — системы компьютерной математики, напрямую не разрабатывавшиеся для DM, но содержащие большое разнообразие алгоритмов и средств визуализации, которые позволяют реализовывать функциональность DM. Они могут работать не только с таблицами признаков и временными рядами, но и изображениями, видео- и звуковыми файлами.

4. Интеграционные пакеты (IDMT — Integration Data Mining Tool) — наборы алгоритмов, образующих либо отдельные программные средства, такие как KNIME, WEKA, KEEL, TANAGRA, либо пакеты расширения Gait-CAD, PRTools, RWEKA. Данные средства ориентированы не на практический анализ, а на разработку новых методов и алгоритмов, их сравнение и оценку эффективности.

5. Пакеты расширения (DMEP — Data Mining Extend Package) – модули подключения к программным средствам, таким, как Excel, Matlab, R, реализующие определенную (как правило, узкую) функциональность DM. Примерами являются Matlab Neural Networks Toolbox, Forecaster XL и XLMiner. Интерактивность в таких средствах обеспечивается базовым приложением.

6. DM-библиотека (DMLT — Data Mining Library Tools) — наборы функций DM, которые могут быть внедрены в другие приложения с помощью API. Графический интерфейс отсутствует, поэтому используются они в основном разработчиками. Примерами являются WEKA, JAVA Data Mining Package, LibSVM, Neurofusion, XELOPES.

7. Средства, ориентированные на использование какого-либо одного семейства алгоритмов или методов DM (SDMT — Specialties Data Mining Tools) — нейронных сетей, деревьев решений, ассоциативных правил и т.д. Примерами являются CART, C5.0, WizRule, Rule Discovery System для деревьев решений, Bayesian Lab для байесовских сетей, MagnumOpus для ассоциативного анализа, JavaNNS, Neuroshell, NeuroProject для нейронных сетей.

8. Исследовательские средства DM (RDMT — Research Data Mining Tools) — экспериментальное ПО, включающее новые, экспериментальные алгоритмы и разработки. Многие популярные аналитические инструменты, такие как Rapid Miner и WEKA, зародились как RDMT-приложения и в процессе эволюции перешли в DMS-класс.

9. Средства, ориентированные на определенную прикладную область (DMFT — Data Mining Field Tools), такие, как Text Mining, обработка и анализ изображений (ITK, ImageJ).

## ****Классификация данных с использованием деревьев решений****

**Деревья решений наряду с нейронными сетями относятся к числу самых популярных и мощных инструментов Data Mining, широко используемой для решения задач численного предсказания и классификации. Так же, как и нейронные сети, деревья решений являются моделями, строящимися на основе машинного обучения с учителем, иногда называемого индуктивным обучением. В этих случаях все объекты тренировочного набора данных заранее отнесены к одному из предопределенных классов.**

**Деревья решений представляют собой древовидные иерархические структуры, состоящие из решающих правил вида «Если…,то…»**[[10]](#footnote-10)**. В основе работы деревьев решений лежит процесс рекурсивного разбиения исходного множества наблюдений или объектов на подмножества, ассоциированные с классами. Разбиение производится с помощью решающих правил, в которых осуществляется проверка значений атрибутов по заданному условию. Рекурсивными называются алгоритмы, которые на каждом последующем шаге используют результаты, полученные на предыдущем.**

**Деревья решений состоят из объектов 2-х типов – узлов и листьев. В узлах располагаются правила, а в листьях – результат, т.е. наблюдения исходного набора данных, имеющих одинаковую переменную класса. Таким образом, каждый лист в дереве связан (или, как часто говорят в литературе, – ассоциирован) с определенным классом.**

**Дерево, представленное на рис. 1, решает задачу классификации объектов по четырем атрибутам на три класса. Визуально узлы и листья в дереве хорошо различимы: в узлах указываются правила, разбивающие содержащиеся в нем наблюдения, и производится дальнейшее ветвление. В листьях правил нет, они помечаются меткой класса, объекты которого попали в данный лист. Ветвление в листьях не производится, и они заканчивают собой ветвь дерева (поэтому их иногда называют терминальными узлами**[[11]](#footnote-11)**).**

Изображение выглядит как текст, диаграмма, линия, Шрифт

Автоматически созданное описание

**Рис. 1. Структура дерева решений.**

**Обратите внимание, что ДР является линейным классификатором, т.е. производит разбиение объектов в многомерном пространстве признаков плоскостями. Для двумерного случая это поясняется рис. 2.**

Изображение выглядит как Прямоугольник, снимок экрана, прямоугольный, линия

Автоматически созданное описание

**Рис. 2. Дерево решений как линейный классификатор.**

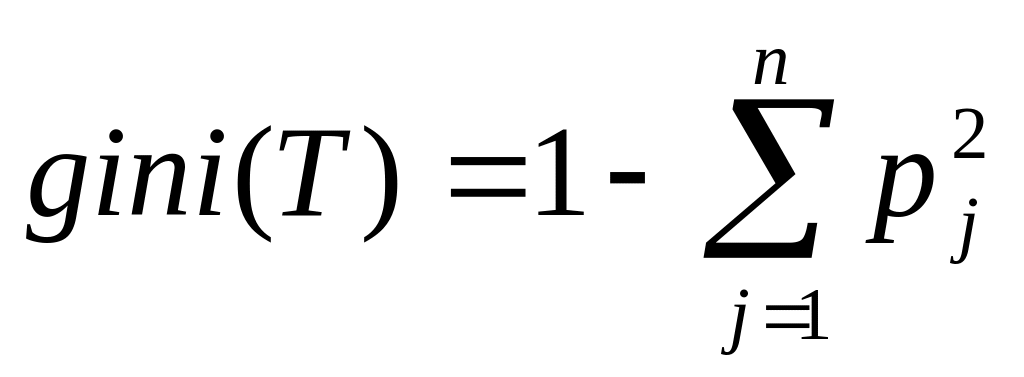
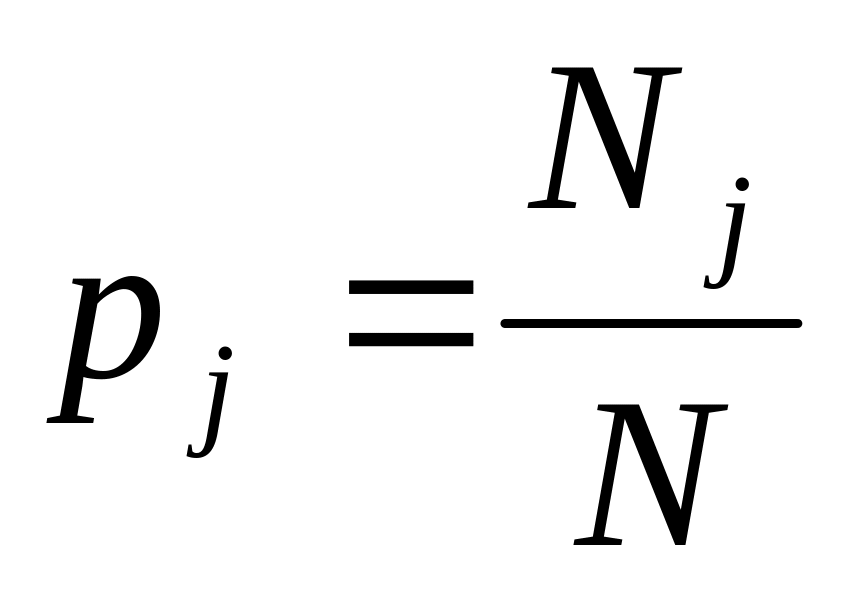
**Пусть квадраты представляют объекты класса 1, круги — класса 2, а треугольники — класса 3. Расположение объектов на рисунке качественно отражает их расположение в пространстве признаков. Пространство признаков разделено линиями на три подмножества, ассоциированных с классами. Эти же подмножества будут соответствовать и трем возможным исходам классификации. Обратите внимание, что во всех трех подмножествах имеются нераспознанные примеры, т.е. примеры, попавшие в подмножества, ассоциированные с другим классом.**

**Теоретически алгоритм может генерировать новые разбиения до тех пор, все примеры не будут распознаны правильно, т.е. пока подмножества, ассоциированные с листьями, не станут однородными по классовому составу. Однако это приводит к усложнению дерева: большое число ветвлений усложняет его структуру и ухудшает интерпретируемость пользователем. Действительно, при использовании НС аналитика интересует только реакция модели на определенное входное воздействие, а сама структура сети и значения ее весов никакой смысловой нагрузки не несут. В то же время правила в ДР формируются практически на естественном языке, что делает его объясняющую способность очень высокой. Иными словами, большой интерес представляют не только сами деревья как модели, делающие предсказания, но и правила, автоматически формируемые при их построении.**

**Алгоритмы конструирования деревьев решений состоят из этапов "построение" или "создание" дерева (tree building) и "сокращение" дерева (tree pruning). В ходе создания дерева решаются вопросы выбора критерия расщепления и остановки обучения (если это предусмотрено алгоритмом). В ходе этапа сокращения дерева решается вопрос отсечения некоторых его ветвей.**

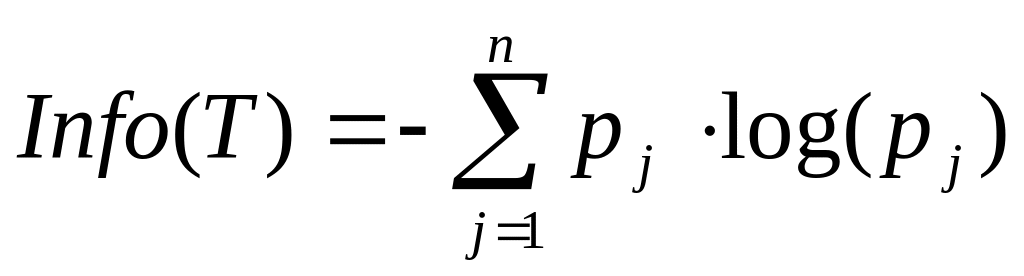
**Процесс создания дерева происходит сверху вниз, т.е. является нисходящим. В ходе процесса алгоритм должен найти такой критерий расщепления, иногда также называемый критерием разбиения, чтобы разбить множество на подмножества, которые бы ассоциировались с данным узлом проверки. Каждый узел проверки должен быть помечен определенным атрибутом. Существует правило выбора атрибута: он должен разбивать исходное множество данных таким образом, чтобы объекты подмножеств, получаемых в результате этого разбиения, являлись представителями одного класса или же были максимально приближены к такому разбиению. Последняя фраза означает, что количество объектов из других классов, так называемых "примесей", в каждом классе должно стремиться к минимуму.**

**Существуют различные критерии расщепления. Наиболее известные – мера энтропии и индекс Gini. Если дано множество *T*, включающее примеры из *n* классов, индекс Gini, определяется по формуле:**

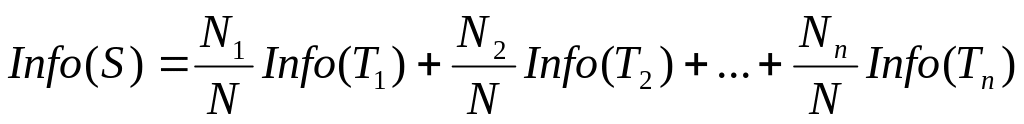
 ,    – доля класса ***j*** в узле ***T***.

**где *T* – текущий узел, *pj* – вероятность класса *j* в узле *T*, *n* – количество классов, *N* – количество объектов в узле.**

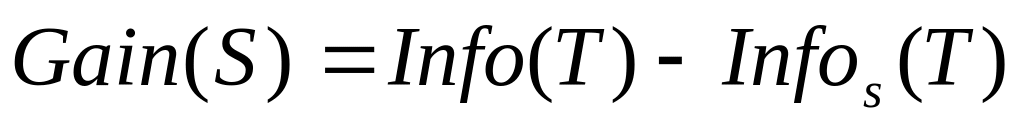
**Мера энтропии при построении деревьев решений – это мера разнообразия классов в узле. В результате разбиения должны образовываться узлы с меньшим разнообразием состояний входной переменной. Следовательно, энтропия падает, а количество внутренней информации в узле растет. Формально энтропия определенного узла *Т* дерева решений определяется:**



**Энтропия всего разбиения – сумма энтропий всех узлов, умноженная на долю записей каждого узла в общем числе записей:**



**Для выбора атрибута расщепления используется критерий, называемый приростом информации или уменьшением энтропии:**



**В качестве наилучшего атрибута для использования в разбиении *S* выбирается тот, который обеспечивает наибольший прирост информации *Gain(S)*.**

**Большое дерево не означает, что оно "подходящее". Чем больше частных случаев описано в дереве решений, тем меньшее количество объектов попадает в каждый частный случай. Такие деревья называют "ветвистыми" или "кустистыми", они состоят из неоправданно большого числа узлов и ветвей, исходное множество разбивается на большое число подмножеств, состоящих из очень малого числа объектов. В результате "переполнения" таких деревьев их способность к обобщению уменьшается, и построенные модели не могут давать верные ответы.**

**В процессе построения дерева, чтобы его размеры не стали чрезмерно большими, используют специальные процедуры, которые позволяют создавать оптимальные деревья, так называемые деревья "подходящих размеров".**

**Какой размер дерева может считаться оптимальным? Дерево должно быть достаточно сложным, чтобы учитывать информацию из исследуемого набора данных, но одновременно оно должно быть достаточно простым. Другими словами, дерево должно использовать информацию, улучшающую качество модели, и игнорировать ту информацию, которая ее не улучшает.**

**Тут существует две возможные стратегии. Первая состоит в наращивании дерева до определенного размера в соответствии с параметрами, заданными пользователем. Определение этих параметров может основываться на опыте и интуиции аналитика, а также на некоторых "диагностических сообщениях" системы, конструирующей дерево решений. Вторая стратегия состоит в использовании набора процедур, определяющих "подходящий размер" дерева, они разработаны Бриманом, Куилендом и др. в 1984 году. Однако, как отмечают авторы, нельзя сказать, что эти процедуры доступны начинающему пользователю.**

**Процедуры, которые используют для предотвращения создания чрезмерно больших деревьев, включают: сокращение дерева путем отсечения ветвей (стрижка)**[[12]](#footnote-12)**; использование правил остановки обучения. Следует отметить, что не все алгоритмы при конструировании дерева работают по одной схеме. Некоторые алгоритмы включают два отдельных последовательных этапа: построение дерева и его сокращение; другие чередуют эти этапы в процессе своей работы для предотвращения наращивания внутренних узлов.**

**Остановка построения дерева. Рассмотрим правило остановки. Оно должно определить, является ли рассматриваемый узел внутренним узлом, при этом он будет разбиваться дальше, или же он является конечным узлом, т.е. узлом решением. Остановка – такой момент в процессе построения дерева, когда следует прекратить дальнейшие ветвления.**

**Один из вариантов правил остановки - "ранняя остановка" (prepruning), в конечных узлах дерева обычно оказывается мало примеров. Правила в таких узлах имеют низкую значимость и объясняющую способность. Поэтому, задав минимально допустимое количество примеров в узле дерева, можно ограничить его рост, запретив создавать новые узлы. Преимущество использования такого варианта – уменьшение времени на обучение модели. Однако здесь возникает риск снижения точности классификации.**

**Второй вариант остановки обучения – ограничение глубины дерева. В этом случае построение заканчивается, если достигнута заданная глубина.**

**Еще один вариант остановки – задание минимального количества примеров, которые будут содержаться в конечных узлах дерева. При этом варианте ветвления продолжаются до того момента, пока все конечные узлы дерева не будут чистыми или будут содержать не более чем заданное число объектов.**

**Сокращение дерева или отсечение ветвей. Решением проблемы слишком ветвистого дерева является его сокращение путем отсечения (pruning) некоторых ветвей. Сначала строится полное дерево, распознающее все примеры. Как правило, оно получается слишком сложным и трудно интерпретируемым. Затем производится автоматическое удаление узлов и листьев, имеющих малую значимость, т. е. тех, в которые попало лишь небольшое количество примеров. Понятно, что решающие правила, относящиеся к очень узкому кругу наблюдений, практически не имеют ценности. Наоборот, правила, позволяющие классифицировать большое число наблюдений, являются наиболее значимыми. В качестве параметра отсечения ветвей можно использовать показатель, называемый достоверностью[[13]](#footnote-13).**

**Качество классификационной модели, построенной при помощи дерева решений, характеризуется двумя основными признаками: точностью распознавания и ошибкой. Точность распознавания рассчитывается как отношение объектов, правильно классифицированных в процессе обучения, к общему количеству объектов набора данных, которые принимали участие в обучении. Ошибка рассчитывается как отношение объектов, неправильно классифицированных в процессе обучения, к общему количеству объектов набора данных, которые принимали участие в обучении.**

**Отсечение ветвей или замену некоторых ветвей поддеревом следует проводить там, где эта процедура не приводит к возрастанию ошибки. Процесс проходит снизу-вверх, т.е. является восходящим. Это более популярная процедура, чем использование правил остановки. Деревья, получаемые после отсечения некоторых ветвей, называют усеченными. Если такое усеченное дерево все еще не является интуитивным и сложно для понимания, используют извлечение правил, которые объединяют в наборы для описания классов. Каждый путь от корня дерева до его вершины или листа дает одно правило. Условиями правила являются проверки на внутренних узлах дерева.**

**Таблица сопряженности. Для быстрой и эффективной оценки надежности классификации с помощью ДР используется таблица сопряженности, которая является аналогом диаграммы рассеяния для задачи численного предсказания. Пример таблицы сопряженности представлен на рис. 3.**

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

**Рис. 3. Таблица сопряженности.**

**Ячейки, расположенные на главной диагонали таблицы и стоящие на пересечении одноименных строк и столбцов, содержат число примеров, правильно распознанных для данного класса. Любое число в ячейке, расположенной вне главной диагонали, т. е. на пересечении разноименных строки и столбца, показывает число неправильно распознанных примеров для данного класса.**

**Управление процессом обучения ДР. Большинство приложений, реализующих ДР, предоставляют пользователю возможность управления процессом обучения, позволяя выбирать параметры алгоритма. Основными параметрами являются настройки ограничения размера дерева, позволяющие избежать его переобучения, снизить сложность и улучшить интерпретируемость. Как было отмечено выше, есть два способа ограничения — ранняя остановка и отсечение ветвей. Если выбрана ранняя остановка, то необходимо ограничить минимально допустимое количество примеров в узле.**

**Если выбрано отсечение ветвей, то сначала строится полное дерево, а затем узлы, начиная с нижнего уровня дерева, последовательно преобразовываются в листья. Очевидно, что при отсечении каждого узла (т.е. фактически правила) общая ошибка дерева возрастает, поэтому процесс усечения дерева имеет смысл производить, пока ошибка не возрастет до некоторого разумного значения, которое обеспечит баланс между сложностью и точностью дерева. Выбор способа упрощения дерева и его параметров может производиться только на основе предыдущего опыта, если подобная задача уже решалась, либо подбираться экспериментально.**

**В процессе работы алгоритма пользователь получает информацию о числе примеров, распределенных по узлам дерева, количестве распознанных и не распознанных примеров на обучающем и тестовом множествах, а также количестве узлов и правил в дереве (количество узлов всегда будет больше, чем количество правил, поскольку листья тоже учитываются). В теории ДР пример считается распознанным, если метка класса, присвоенная деревом, соответствует фактическому классу объекта. Процент распознанных примеров является хорошим критерием для оценки качества обучения: если распознано больше 70%, то качество можно считать хорошим, если 50–70% — удовлетворительным, если ниже 50%, то плохим. В последнем случае практическое использование построенной модели не имеет смысла и ее следует перестроить, скорректировав обучающий набор данных или параметры обучения.**

**Программные средства, реализующие деревья решений. Так же как и в случае нейронных сетей, деревья решений реализуются как специализированными программными средствами, так и входят в состав аналитических пакетов. К последним относятся IBM Data Modeler, SAS Enterprise Miner, Rapid Miner, Deductor и др. Специализированными инструментами, разработанными специально для моделирования на основе ДР, являются SPSS Answer Tree, See5, С4.5 и др**[[14]](#footnote-14)**.**

# **Загрузка данных в аналитическое приложение**

В любой процедуре аналитической обработки данных первым этапом стоит загрузка данных в аналитическое приложение. Т.к. данные могут хранится в разных по типу источниках, а также могут быть представлены в различных формах, – это задача не такая простая. При это данные должны быть структурированными. Поля, содержащие записи об описываемом объекте или явлении, должны быть типизированными, т.е. содержать значения только одного тип (числовые, строковые, логические, дата/время и т.д.). Если условия не соблюдаются, то при загрузке данных возникнут ошибки и пропуски.

Данные, необходимые для анализа, могут содержаться в самых разнообразных типах файлов и форматах:

* **Данные, хранящиеся в отдельных (локальных) файлах.** Текстовые файлы с разделителями, документы Word, Excel и т. д. Любой файл, данные в котором организованы в виде столбцов и записей.
* **Базы данных (БД) различных СУБД.** Oracle, SQL Server, Firebird, dBase, FoxPro, Access и т.д.
* **Специализированные хранилища данных (ХД).** Являются наиболее оптимальными, т.к. их структура и функционирование специально оптимизируются для работы с аналитической платформой.
* **Файлы в TXT-формате.** Более уязви с точки зрения нарушения структуры, т.к. имеют произвольную структуру данных, несмотря на то, что эти файлы очень просты в создании (Блокнот от Windows).
* **Текстовой файл с разделителями.** Состоит из вертикальных столбцов и горизонтальных строк, из которых будут сформированы поля и записи соответственно. Для разделения столбцов должны использоваться однотипные символы-разделители, например, пробел, табуляция, точка с запятой и др. Можно использовать как один символ, так и несколько идущих подряд символов.

Версия Academic аналитической платформы Deductor поддерживает работу только с текстовыми файлами, поэтому с помощью редактора Блокнот создадим текстовый файл с разделителями. В качестве данных будем использовать результаты агрохимического исследования почвы (приложение А).

1. **В редакторе Блокнот создаем файл с разделителями. В первой строке вводим заголовки полей.**
2. **Запускаем аналитическое приложение Deductor Studio.**
3. **Запускаем Мастер импорта и выбираем тип файла для загрузки, в нашем случае пункт Text (direct) (рис. 4).**

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

**Рис. 4. Запуск Мастера импорта и выбор источника данных**

1. **Указываем путь и имя загружаемого файла, проверяем содержание файла (рис. 5).**

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

**Рис. 5. Выбор файла-источника данных и предварительный просмотр**

1. **На следующем шаге Мастера импорта устанавливаем параметры загружаемого текстового файла (рис. 6).**

Изображение выглядит как текст, снимок экрана, дисплей, число

Автоматически созданное описание

**Рис. 6. Настройка параметров загрузки**

1. **Выполняем настройку полей загружаемого файла (рис. 7).**

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

**Рис. 7. Настройка полей загружаемого файла**

1. **Запускаем процесс загрузки посредством нажатия кнопки *«Пуск»*. Процесс загрузки прошел успешно (рис. 8).**

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

**Рис. 8. Запуск процесса загрузки файла**

1. **Выбираем способ визуализации загруженных файлов (рис. 9). Устанавливаем флажки *«Таблица»* и *«Статистика»*.**

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

**Рис. 9. Выбор параметров визуализации данных**

1. **Просматриваем результат загрузки данных (рис. 10, 11).**

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рис. 10. Метод визуализации «Таблица».

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рис. 11. Метод визуализации «Статистика».

**За время выполнения работы мы ознакомились с процессом загрузки данных в аналитическое приложение Deductor Studio Academic. Изучили, что структурированными данными называют упорядоченные в вертикальные столбцы (поля) и горизонтальные строки (записи) данные, обязательно типизированные в полях, т.е. содержащие значения одного типа. Типичным способом представления структурированных данных является таблица. Обычный текст не является структурированными данными и непригоден к непосредственному анализу, но возможно его преобразовать методами структуризации. Так же узнали, что в одном поле одновременно может содержаться только один тип данных (**вещественный, логический, строковый или целый). При создании текстового файла с разделителями узнали, что он должен состоять из вертикальных столбцов и горизонтальных строк, для разделения столбцов должны использоваться однотипные символы-разделители, можно использовать как один символ, так и несколько идущих подряд символов.

# **Построение и обучение деревьев решений**

Для построения и обучения деревьев решений будем использовать загруженные нами данные агрохимического исследования почвы.

1. Запускаем аналитическое приложение Deductor Studio и загружаем выборку «Урожайность»
2. Открываем «Мастер обработки», в разделе Data Mining в списке доступных методов обработки выбираем пункт «Дерево решений» (рис. 12)Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

   Автоматически созданное описание

Рис. 12. Обработка «Дерево решений».

1. Устанавливаем назначения атрибутов – входные, выходные и неиспользуемые (рис. 13).

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 13. Настройка назначений атрибутов.

1. Обозначаем размеры обучающего и тестового множеств 90 % и 10 % соответственно, способ отбора – случайный (рис. 14).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рис. 14. Настройка размеров обучающего и тестового множеств.

1. Настраиваем параметры обучения дерева решений:

**параметры ранней установки:** минимальное количество примеров в узле, при котором будет создан новый узел, – 2, флажок «Строить дерево с более достоверными правилами в ущерб компактности» – установить;

**отсечение узлов дерева** – отключить, сбросив соответствующий флажок в разделе Параметры отсечения (рис. 15).

Таким образом, мы настроили алгоритм обучения на построение «полного» дерева, содержащего максимальное число правил.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рис. 15. Параметры обучения дерева решений.

1. На следующем шаге нужно запустить процесс обучения и проконтролировать его результаты (рис. 16).

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис.16. Запуск процесса обучения и контроль его результатов.

1. Выбираем способы визуализации: ***«Дерево решений», «Правила», «Таблица сопряженности»***.
2. Оцениваем визуально сложность и интерпретируемость дерева решений. Для просмотра содержимого узла выделить его и щелкнуть по кнопке , в результате чего внизу откроется список попавших в данный узел объектов. Правило в узле является тем более эффективным, чем больше примеров будет распределено в данный узел.

# **Анализ результатов обучения ДР.**

Анализируем результаты обучения за два запуска в аналитической платформе Deductor Studio.

В первом случае процент распознанных примеров на обучающем множестве: 98%. Процент распознанных примеров на тестовом множестве: 50%. Число правил в построенном дереве: 4. Число узлов в построенном дереве: 7.

По внешнему виду дерева решений можем увидеть, что в первом запуске участвуют критерии «кислотность» и «фосфор», сложности в интерпретации нет (рис. 17).

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 17. «Дерево решений» первый пример.

По правилам (рис. 18) делаем вывод, что при кислотности >= 5.35 и фосфору >= 18.61 – урожайность высокая, если выполняется условие по кислотности почвы, но содержание фосфора меньше 18.61, то урожайность средняя, если почва по кислотности меньше 5.35 – урожайность низкая.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 18. «Правила» первый пример.

В таблице сопряженности ячейки с числом правильно распознанных примеров отображаются в зеленых ячейках, а неправильно распознанных – в красных. В первый запуск модель допустила 4 ошибки классификации (рис. 19).

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рис. 19. «Таблица сопряженности» первый пример.

Во втором случае процент распознанных примеров на обучающем множестве: 100%. Процент распознанных примеров на тестовом множестве: 100%. Число правил в построенном дереве: 5. Число узлов в построенном дереве: 9 (рис. 20).

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рис. 20. Результаты процесса обучения.

По внешнему виду дерева решений видим, что в этот раз используются критерии «кислотность» и «калий», длина дерева решений увеличена по сравнению с предыдущим запуском (рис. 21).

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рис. 21. «Дерево решений» второй пример.

По правилам (рис. 22) делаем вывод, что при кислотности >= 5.35 и содержанию калия >= 18.14 – урожайность высокая, если выполняется условие по кислотности почвы, но содержание фосфора меньше 11.575, то урожайность средняя, если почва по кислотности меньше 5.35 – урожайность низкая.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рис. 22. «Правила» второй пример.

В таблице сопряженности мы видим, что модель не допустила ошибок классификации (рис. 23).

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рис. 23. «Таблица сопряженности» второй пример.

# **Заключение**

Курсовое проектирование посвящено исследованию процедур применения деревьев решений в задачах классификации. В результате было построено и обучено дерево решений в аналитическом приложении Deductor Studio Academic.

Во время выполнения работы были успешно завершены следующие задачи:

1. Изучены методы интеллектуального анализа данных.
2. Изучен метод анализа данных с помощью ДПР.
3. Загружены данные в аналитическое приложение.
4. Построено и обучено НМ в аналитической платформе Deductor Studio.
5. Классифицированы данные с использованием деревьев решений.
6. Проанализированы результаты обучения.

Для загрузки данных в аналитическое приложение был использовать файл формата «Text» с разделителями. Данными послужили результаты агрохимического исследования почвы. По окончанию загрузки данных были выбраны способы визуализации «***«Таблица»* и *«Статистика».***

Для классификации данных был применен метод обработки больших данных – дерево решений. Были проведены два этапа обучения, по завершению обучения выбраны способы визуализации ***«Дерево решений», «Правила», «Таблица сопряженности»***. По результатам обучения проведен анализ.

# **Список использованных источников**

1. **Алексеев, Д. С.** Технологии интеллектуального анализа данных: учебник для вузов / Д. С. Алексеев, О. В. Щекочихин. – Санкт-Петербург: Лань, 2022. – 176с.
2. **Васильев, Е. П.** Интеллектуальный анализ данных в технологиях принятия решений: учеб. пособие / Е. П. Васильев, В. И. Орешков. – Рязань: Рязан. гос. радиотехн. ун-т., 2023. – 179 с.
3. **Кафтанников, И. Л.** Особенности применения деревьев решений в задачах классификации / И. Л. Кафтанников, А. В. Парасич // Вестник Южно-Уральского государственного университета. – 2015. – № 15. – С. 26-32.
4. **Корячко, В. П.** Интеллектуальные системы и нечеткая логика: учебник / В. П. Корячко, М. А. Бакулева, В. И. Орешков. – Москва: КУРС, 2023. – 352 с.
5. **Мусаев, А. А.** Интеллектуальный анализ данных: учебное пособие / А. А. Мусаев. – СПб: СПбГТИ(ТУ), 2018. – 56с.
6. **Орешков, В. И.** Интеллектуальный анализ данных как современный инструмент поддержки управленческих решений [Текст] / В.И. Орешков// Вестник Рязанского гос. агротехнологического университета имени П.А. Костычева. Рязань: РГАТУ. — 2011. — № 4. — С. 55–59.
7. **Паклин, Н. Б.** Бизнес-аналитика: от данных к знаниям: учеб. пособие / Н. Б. Паклин, В.И. Орешков. — 2-е изд., испр. — СПб.: Питер, 2013. — 704 с.: ил.
8. **Прокопенко, Н. Ю.** Системы поддержки принятия решений: учеб. пособие / Н. Ю. Прокопенко. – Н. Новгород: Нижегор. гос. архитектур.-строит. ун-т., 2017. – 188 с.

# **Приложение А**

№ поля Площадь, га Кислотность, pH Азот, мг/100г Калий, мг/100г Фосфор, мг/100г Юг, % Ср. угол Урожайность, ц/га Урожайность (класс) Показатель

1 183,01 4,8 3 19,3 18,52 91,36 1,28 2,2 Низкая Азот

2 185,99 5,5 5 10,83 19,15 67,9 0,83 4,8 Средняя Азот

3 47,52 5,6 6 12,76 16,0 44,24 0,92 6,3 Средняя Азот

4 63,04 5,5 6 12,15 14,0 92,8 0,69 6,4 Средняя Азот

5 141,08 5,2 4 24,42 19,55 26,26 1,05 3,5 Низкая Азот

6 100,06 5,2 4 14,89 10,8 20,05 1,51 3,4 Низкая Азот

7 18,15 5 3 7,4 6 1,33 1,78 2,2 Низкая Азот

8 99,64 5,4 5 15,97 25,2 0,12 1,38 4,7 Средняя Азот

9 91,36 5,1 4 9,31 6,94 12,38 1,36 3,5 Низкая Азот

10 118,06 5,6 6 11,17 13,73 0,14 1,44 6,3 Средняя Азот

11 73,24 4,9 3 13,15 12,32 8,81 1,14 2,2 Низкая Азот

12 128,73 6,1 9 6,91 16,11 50,15 1,28 5,6 Средняя Калий

13 40,87 6,1 9 6,43 12,93 99,97 1,8 5,0 Средняя Калий

14 47,29 0 0 0 0 74,68 1,13 0,5 Низкая

15 191 6 9 17,78 19,74 14,34 1,21 11,2 Высокая Нет

16 35,48 4,9 3 9,3 7,25 0,02 1,41 2,2 Низкая Азот

17 17,7 4,4 1 6,8 10 46,39 2,32 0,4 Низкая Азот

18 79,67 4,9 3 14,23 14,21 0,9 1,07 2,2 Низкая Азот

19 58,26 6,1 9 14,42 23,28 43,6 1,01 11,6 Высокая Нет

20 84,62 5,4 5 8,64 9,33 0,06 1,01 4,8 Средняя Азот

21 31,77 4,9 3 13,5 12,32 23,25 0,86 2,2 Низкая Азот

22 32,3 5,3 4 6,73 6,33 18,57 0,99 3,5 Низкая Азот

23 26 5,9 8 18,58 20,5 0 1,46 9,7 Высокая Нет

24 39,8 4,6 2 8,98 19,44 26,52 1,33 1,2 Низкая Азот

25 202,8 5,4 5 16,78 11,13 88,09 0,71 4,8 Средняя Азот

26 30,85 6,1 9 17,4 25,07 99,75 1,37 10,9 Высокая Нет

27 97,8 5,7 7 6,63 14,11 33,2 0,64 5,2 Средняя Фосфор

28 37,01 5,7 7 6,63 14,11 43,92 1,21 5,1 Средняя Азот

29 71,99 5 3 10,68 8,22 16,23 1 2,1 Низкая Азот

30 81,95 5,1 4 19,59 17,56 33,02 0,7 3,6 Низкая Азот

31 69,32 5 3 10,68 8,22 29,9 0,91 2,2 Низкая Азот

32 10,85 5,1 4 19,59 17,56 0 1,05 3,5 Низкая Азот

33 82,05 4,8 3 10,59 9,75 2,41 0,81 2,3 Низкая Азот

34 14,85 5,1 4 15,59 15,71 62,63 0,82 3,5 Низкая Азот

35 17,24 4,8 3 10,59 9,75 3,51 0,55 2,1 Низкая Азот

36 118,96 5,1 4 15,59 15,71 18,05 0,81 3,4 Низкая Азот

37 21,65 4,7 2 8,55 13,88 7,52 0,91 1,2 Низкая Азот

38 118,96 5,1 4 15,59 15,71 18,05 0,81 3,5 Низкая Азот

39 21,65 4,7 2 8,55 13,88 7,52 0,91 1,1 Низкая Азот

40 60,13 5,0 3 10,25 19,83 5,63 1,03 2,2 Низкая Азот

41 25,36 5,2 4 9,77 17,33 13,97 1,0 3,5 Низкая Азот

42 16,45 5,1 4 12,07 9,33 91,35 0,9 3,5 Низкая Азот

43 15,4 5,8 7 20,9 21,5 12,29 1,03 8,0 Высокая Азот

44 222,61 6,0 9 6,13 13,91 40,65 0,76 4,7 Средняя Калий

45 70,93 5,4 5 17,7 17,48 3,31 1,58 4,8 Средняя Азот

46 115,87 6,1 9, 7,49 12,18 51,5 0,84 6,3 Средняя Калий

47 128,4 5,9 8 11,98 25,19 46,99 1,15 9,7 Высокая Нет

48 132,53 5,8 7 14,74 14,21 18,14 0,74 8,0 Высокая Азот

49 168,1 4,7 2 10,66 13,22 75,35 0,8 1,2 Низкая Азот

50 45,95 6,1 9 7,21 11,99 4,37 3,37 5,9 Средняя Азот

51 77,28 6,0 9 6,32 10,5 16,67 2,12 4,9 Средняя Калий

52 58,53 4,8 3 11,42 14,8 37,76 0,51 2,2 Низкая Азот

53 151,65 4,8 3 10,89 13,23 5,84 1,07 2,2 Низкая Азот

54 38,23 5,6 6 13,08 25,2 0,08 1,66 6,3 Средняя Азот

55 24,68 6,1 9 19,0 20,5 14,47 1,28 11,2 Высокая Нет

56 15,17 6,3 8 5,6 12,25 0,06 1,11 4,1 Средняя Калий

1. Алексеев, Д. С. Технологии интеллектуального анализа данных: учебник для вузов / Д. С. Алексеев, О. В. Щекочихин. – Санкт-Петербург: Лань, 2022. – С. 5. [↑](#footnote-ref-1)
2. Васильев, Е. П. Интеллектуальный анализ данных в технологиях принятия решений: учеб. пособие / Е. П. Васильев, В. И. Орешков. – Рязань: Рязан. гос. радиотехн. ун-т., 2023. – С. 34. [↑](#footnote-ref-2)
3. Мусаев, А. А. Интеллектуальный анализ данных: учебное пособие / А. А. Мусаев. – СПб: СПбГТИ(ТУ), 2018. – С.7. [↑](#footnote-ref-3)
4. Васильев, Е. П. Интеллектуальный анализ данных в технологиях принятия решений: учеб. пособие / Е. П. Васильев, В. И. Орешков. – Рязань: Рязан. гос. радиотехн. ун-т., 2023. – С. 35. [↑](#footnote-ref-4)
5. Мусаев, А. А. Интеллектуальный анализ данных: учебное пособие / А. А. Мусаев. – СПб: СПбГТИ(ТУ), 2018. – С.8. [↑](#footnote-ref-5)
6. Корячко, В. П. Интеллектуальные системы и нечеткая логика: учебник / В. П. Корячко, М. А. Бакулева, В. И. Орешков. – Москва: КУРС, 2023. – С. 65. [↑](#footnote-ref-6)
7. Паклин, Н. Б. Бизнес-аналитика: от данных к знаниям: учеб. пособие / Н. Б. Паклин, В.И. Орешков. — 2-е изд., испр. — СПб.: Питер, 2013. — С. 403. [↑](#footnote-ref-7)
8. Корячко, В. П. Интеллектуальные системы и нечеткая логика: учебник / В. П. Корячко, М. А. Бакулева, В. И. Орешков. – Москва: КУРС, 2023. – С. 68. [↑](#footnote-ref-8)
9. Орешков, В. И. Интеллектуальный анализ данных как современный инструмент поддержки управленческих решений [Текст] / В.И. Орешков// Вестник Рязанского гос. агротехнологического университета имени П.А. Костычева. Рязань: РГАТУ. — 2011. — № 4. — С. 55–59. [↑](#footnote-ref-9)
10. Корячко, В. П. Интеллектуальные системы и нечеткая логика: учебник / В. П. Корячко, М. А. Бакулева, В. И. Орешков. – Москва: КУРС, 2023. – С. 136. [↑](#footnote-ref-10)
11. Васильев, Е. П. Интеллектуальный анализ данных в технологиях принятия решений: учеб. пособие / Е. П. Васильев, В. И. Орешков. – Рязань: Рязан. гос. радиотехн. ун-т., 2023. – С. 66. [↑](#footnote-ref-11)
12. Корячко, В. П. Интеллектуальные системы и нечеткая логика: учебник / В. П. Корячко, М. А. Бакулева, В. И. Орешков. – Москва: КУРС, 2023. – С. 137. [↑](#footnote-ref-12)
13. Васильев, Е. П. Интеллектуальный анализ данных в технологиях принятия решений: учеб. пособие / Е. П. Васильев, В. И. Орешков. – Рязань: Рязан. гос. радиотехн. ун-т., 2023. – С. 68. [↑](#footnote-ref-13)
14. Корячко, В. П. Интеллектуальные системы и нечеткая логика: учебник / В. П. Корячко, М. А. Бакулева, В. И. Орешков. – Москва: КУРС, 2023. – С. 146. [↑](#footnote-ref-14)