



НТО ЮНИОРЫ 2026/27 · СФЕРА

ИИтех

Быстрый старт: главное для подготовки к
отборочному этапу

5-7 класс

Автор-составитель

Артём Осинowski, методист и разработчик детского технопарка «Кванториум»

Детский технопарк «Кванториум», Новосибирск · 2026

ИИтех

Быстрый старт: главное для подготовки к отборочному этапу НТО Юниоры. 5–7 класс

Автор-составитель – **Артём Осинowski**, методист и разработчик детского технопарка «Кванториум».

Нашли ошибку или неточность? Напишите на osinovsky.aa@kvantorium54.ru (рабочая почта): сфера, версия книги (1.0) и номер страницы – исправим в следующей версии. Актуальная версия и список изменений – nto.kvantorium54.ru/knigi. Личная почта для других вопросов – artem081033@gmail.com.

Важно. Это не официальное пособие олимпиады, а книга методиста «Кванториума». Каких заданий ждать на отборе 2026/27, мы не знаем: организаторы их заранее не показывают, а задания разных лет не повторяются. Книга готовит к темам из программы сферы и приёмам решения, а не к конкретным задачам.

Книга помогает школьнику самостоятельно подготовиться к отборочному этапу сферы «ИИтех» Национальной технологической олимпиады «НТО Юниоры»: теория с нуля, подробные разборы задач и задачи для самопроверки с ответами.

Большинство задач составлены автором по образовательной программе сферы, часть – по мотивам открытых заданий прошлых лет; источник указан у каждой задачи. Уровни сложности – шкала этой книги, а не баллы олимпиады. Правила, сроки и формат отбора – на сайте junior.ntcontest.ru.

Это «Быстрый старт» – сжатая версия книги: по каждой главе главное, один разбор и четыре задачи с ответами. Для подготовки к отбору рекомендуем полную версию – все разборы, задачи и полный справочник, главы те же: nto.kvantorium54.ru/knigi.

Быстрый старт · версия 1.0 от 02.10.2026, сборка 02.10.2026. © Артём Осинowski, 2026.

Содержание

Как устроена эта книга	3
Для кого эта книга	3
Как устроен отбор	3
Что пригодится на отборе и где это в книге	4
Как пользоваться книгой	5
Как вводить ответы на отборе	5
Что понадобится	6
План на месяц до отбора	33
Родителям и наставникам	34
Честно об этой книге	34
Глава 1 Что такое искусственный интеллект	35
Глава 2 Множества и логика	40
Глава 3 Комбинаторика и вероятность	44
Глава 4 Закономерности, последовательности, уравнения	49
Глава 5 Данные: собираем и наводим порядок	54
Глава 6 Графики и анализ данных	59
Глава 7 Нейросети своими руками	64
Глава 8 Python, диалоги и промпты	68
Ответы и решения	73
Справочник	76
Шпаргалка: всё «Запомни» из глав	76

Как устроена эта книга

Эта книга – тренажёр для отборочного этапа сферы «ИИтех» олимпиады «НТО Юниоры». Здесь есть всё, чтобы готовиться самостоятельно: теория с нуля, подробные разборы задач и похожие задачи для самопроверки с ответами в конце книги. Мощный компьютер и знакомый программист не нужны: половина задач решается на бумаге, остальное – в браузере и в бесплатных программах сферы.

Для кого эта книга

- Для школьников **5–7 классов**, которые выбрали (или думают выбрать) сферу «ИИтех».
- Для тех, кто **ни разу не программировал** и не знает, как устроена нейросеть. Каждая тема начинается с самого начала, а первый код мы разбираем построчно.
- Для **наставников и родителей**, которые хотят помочь. Для них есть отдельный раздел в конце введения.

Как устроен отбор

Таблица В.1

Когда	Что происходит
до 9 ноября	регистрация на олимпиаду (наш совет – успеть до 20 октября, тогда будут обе попытки)
20–30 октября	отбор, попытка 1 (3 часа)
31 октября – 10 ноября	отбор, попытка 2 (засчитывают лучшую из двух)
7–17 декабря	финалы в регионах

- Отбор проходит **дома, онлайн, с компьютера**. Задания открываются в личном кабинете на my.ntcontest.ru и решаются на платформе «Орбита» (orbita.education). Каждый решает сам за себя.
- Задания – это задачи с ответом: ввести число, выбрать верные варианты, сопоставить, расставить шаги по порядку, прочитать график или таблицу и сделать вывод.
- Разработчик сферы на ознакомительном вебинаре 2026 года сказала: в образовательном курсе изложен весь материал, который ждёт на отборе и в финале, и ничего нового на отборе не будет. Поэтому книга построена по программе сферы «Основы искусственного интеллекта и анализа данных».
- На вебинаре для наставников в сентябре 2026 года разработчики описали путь, по которому идут и курс, и отбор, и финал: понять задачу, записать её на языке математики, получить данные, исследовать их, построить модель и оценить, насколько хорошо она работает. Главы книги идут в том же порядке.
- В курсе, на отборе и в финале сферы используются инструменты, где программу собирают **из блоков**, без набора кода. Писать на Python с нуля от тебя не потребуют. Но

Python прямо назван на странице сферы в списке того, что пригодится, а в блочном инструменте курса за каждым блоком стоит строчка на Python. Поэтому в книге есть глава про Python.

- За тест бесплатного курса своей сферы дают **+5 баллов**. Курс – на academy.junior.online, вход через «Талант». Бонус нужно включить самому в личном кабинете («Мои привилегии») до конца отбора. Учти: после этого сферу уже не поменять.
- Финал – **командный** и очный: команда из двух-трёх человек около трёх часов решает задачи на компьютерах, на той же «Орбите». Команду можно собрать заранее, а можно прямо на месте.

Совет. Сроки и правила могут уточняться. Главный источник – сайт олимпиады junior.ntcontest.ru и памятка для Новосибирской области nto.kvantorium54.ru.

Что пригодится на отборе и где это в книге

На странице сферы перечислено, что пригодится на отборе. Вот где это в книге:

Таблица В.2

Что пригодится на отборе	Глава
Базовые понятия ИИ на уровне определений	1. Что такое искусственный интеллект
Математика: множества и логика	2. Множества и логика
Математика: комбинаторика (и вероятность – из программы сферы)	3. Комбинаторика и вероятность
Математика: уравнения, выражения с переменными; поиск закономерностей	4. Закономерности, последовательности, уравнения
Информатика: информация и информационные процессы	5. Данные: собираем и наводим порядок
Визуализация данных, графики по таблицам, умение читать графики и делать выводы	6. Графики и анализ данных
Нейросети: классификация, регрессия, точность модели (из программы сферы)	7. Нейросети своими руками
Основы программирования (Python); обработка текстовой информации	8. Python, диалоги и промпты

Главы 1–4 можно читать в любом порядке, для них хватит тетради и карандаша. Главы 5, 6 и 7 лучше читать подряд: в пятой мы собираем данные и наводим в них порядок, в шестой смотрим на них глазами, в седьмой учим на них нейросеть. Главу 8 можно начать когда угодно. Если ты ни разу не писал программ, загляни сначала в мини-шпаргалку Python в конце книги: код встречается уже в главах 5 и 6.

Как пользоваться книгой

В каждой главе теория и разборы идут вперемешку: за каждым разбором сразу следуют две похожие задачи «Реши сам» – чтобы закрепить приём, пока он свежий в голове, не долистывая до конца главы.

1. **Теория.** Коротко и по делу, с рисунками. Каждая новая тема начинается с большой картинки «на пальцах», а самое важное собрано в рамках «Запомни». Все «Запомни» из всех глав повторяются в конце книги, в шпаргалке.
2. **Разбираем задачи.** Задачи такие же, как на отборе, и к каждой – решение по шагам: что дано, что делаем и почему, где обычно ошибаются, какой ответ получился и похож ли он на правду.
3. **Реши сам.** Сразу после разбора – две похожие задачи для самостоятельного решения, тем же приёмом, но с другими числами.
4. **Ответы** – в конце книги, в разделе «Ответы и решения». Там только ответ, иногда – короткая подсказка, если задача с подвохом; полного решения нет специально: если подсказки не хватило, вернись к разбору перед задачей.

Не готов читать всю главу целиком? Есть отдельная короткая книжка «Быстрый старт» (около 50 страниц): из каждой главы главное, один разбор и несколько задач «Реши сам». Её удобно пройти за неделю, а эту книгу держать под рукой.

У каждой задачи есть уровень – как в обычном задачнике: у трудных задач звёздочка рядом с номером («задача 3*»), лёгкие подписаны словом «разминка». Отдельных значков и таблицы уровней в книге нет.

Звёздочка и «разминка» – это уровни нашей книги, а не баллы олимпиады. Сколько задач будет на отборе и сколько баллов стоит каждая, заранее не объявляют.

Совет. Разбор читай с листком бумаги. Закрой решение, попробуй решить сам, а потом сверяйся по шагам. Так ты запомнишь в три раза больше, чем если просто прочитаешь.

У каждой задачи внизу написано, откуда она взялась: «по мотивам задачи отборочного этапа» – значит, похожая задача была на отборе прошлых лет (мы пересказали её своими словами и поменяли числа), «своя задача по программе сферы» – значит, мы придумали её по теме из программы 2026/27 года. Программа сферы сама предлагает такой приём: решить знакомую задачу с другими входными данными. Если ты справился с задачей из разбора, а потом с обеими задачами «Реши сам», – тему ты понял.

Как вводить ответы на отборе

Так было на отборочных этапах прошлых лет. На «Орбите» перед каждой задачей читай инструкцию: там сказано, в каком виде нужен ответ.

- **Только число**, без единиц измерения и без знака процента, если не сказано иначе.
- **Дробную часть** на сайте обычно отделяют **точкой**: 0.85 , а не $0,85$. В книге мы пишем по-русски, через запятую.

- **Доля или проценты?** Точность модели 85 % можно записать как 85 и как 0.85 . Смотри, что просит условие.
- **Округляй так, как сказано в условии.** «До сотых» – два знака после точки. Промежуточные результаты не округляй – только ответ.
- **«Сколько способов»** – всегда целое число. Если получилось 12,5 варианта, ошибка где-то раньше.
- **У каждого могут быть свои числа.** В некоторых задачах условие или набор данных генерируется для каждого участника отдельно, так что списать ответ не выйдет.
- **Попытки ограничены.** Сначала реши на бумаге или проверь программу в онлайн-Python, и только потом вводи ответ.

Что понадобится

- Тетрадь в клетку, карандаш, линейка – для глав 1–4, кругов и таблиц.
- Калькулятор. Хватит обычного на телефоне: проценты, дроби, умножение.
- Компьютер с интернетом и браузером – для глав 5–8. Планшета мало: программы курса ставятся на компьютер (в программе сферы указан Windows 10 или новее). GraphNet и Датавиз-конструктор выпускают только для Windows.
- **Python** – любой онлайн-интерпретатор, ничего ставить не нужно. Для глав про данные и графики удобнее **Google Colab** (colab.research.google.com): там уже есть библиотеки для таблиц и графиков. Полистать готовые примеры можно и так, а чтобы запустить код, нужен аккаунт Google.
- **Сервисы из программы сферы**, по мере того как до них дойдёт курс:
 - **GraphNet** – конструктор нейросетей из блоков;
 - **Датавиз-конструктор** – графики, дашборды и «вычислительные эссе»;
 - **Teachable Machine** и **Machine Learning for Kids** – модели, которые учатся на твоих картинках и звуках без программирования;
 - **CodeChef** (codechef.ru) – блочная среда, в которой в курсе сферы собирают программу сбора данных: блоки HTTP, Pandas, списки, переменные; проект сохраняется файлом «.chef»;
 - **Scratch** – дополнительный материал кружка, не часть курса: в нём можно встроить обученную модель в свою программу;
 - ИИ-чаты **GigaChat** и **Алиса** – для главы про промпты (со взрослым, см. ниже).

Где скачать GraphNet и Датавиз-конструктор и как их поставить, рассказывают в курсе сферы – ссылки собраны в справочнике в конце книги.

CodeChef: программа из блоков

CodeChef (codechef.ru) – бесплатный конструктор программ: их не печатают, а собирают из цветных блоков, как из деталей. Справа тут же виден настоящий код на Python, который получился. В курсе сферы в CodeChef собирают программу сбора данных (модуль 5 «Данные», шаг 5 и задачи 4–5). Ниже – как начать и четыре задачи: от «привет» до сбора таблицы по страницам. Рамка с номером на картинке – куда смотреть и куда нажимать.

Шаг 1. Открыть. Зайди на codechef.ru: страница открывается сразу, входить и регистрироваться не нужно. Python работает прямо в твоём браузере (мы проверяли в Chromium – на

нём построены Chrome, Яндекс Браузер и Edge). Он загружается при открытии страницы, это несколько секунд; во время запуска вверху справа горит «Выполняется».

Шаг 2. Окно программы.

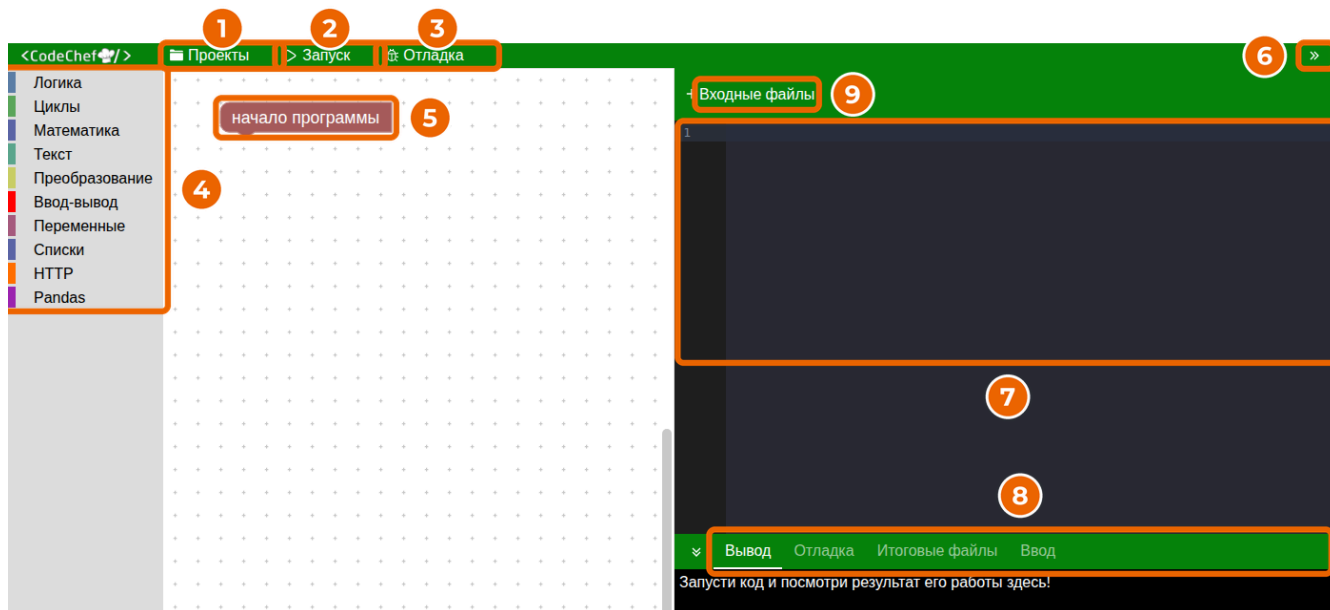


Рисунок В.1 – Окно CodeChef

1. **«Проекты»** – меню: «Открыть», «Создать проект», «Сохранить» (шаг 6).
2. **«Запуск»** – запустить программу. Что она напечатала, видно на вкладке «Вывод» (8).
3. **«Отладка»** – запуск по шагам: видно, как меняются переменные (задача П.3).
4. **Палитра блоков**, десять групп: «Логика», «Циклы», «Математика», «Текст», «Преобразование», «Ввод-вывод», «Переменные», «Списки», «HTTP», «Pandas». Щёлкни по названию – рядом откроются блоки группы.
5. **Рабочее поле**. На нём уже лежит блок «начало программы» – с него всё начинается. Блок, который ни к чему не прицеплен, в код не попадает.
6. **Двойная стрелка** – спрятать правую часть окна (код и вывод) или показать её снова. Когда блоки не помещаются в ширину, спрячь её: поле станет во всё окно.
7. **Код на Python**. Он сам меняется вместе с блоками; печатать в нём нельзя.
8. **Вкладки**: «Вывод» – что напечатала программа; «Отладка» – переменные при пошаговом запуске; «Итоговые файлы» – файлы, которые создала программа; «Ввод» – ответы на вопросы программы.
9. **«Входные файлы»** – добавить с компьютера файл `.txt`, `.json` или `.csv` для программы. В задачах ниже не нужен.

Шаг 3. Как ставить блоки. Открой группу, например «Ввод-вывод» (1 на левой половине), и перетаски блок мышью – берись за его левый край, а не за поле внутри. Блок-команду (с выемкой сверху и выступом снизу) подноси к нижнему краю предыдущего: он защёлкивается. Блок-значение (с выступом слева – число, текст, переменная, вычисление) вставляют в белое гнездо другого блока.

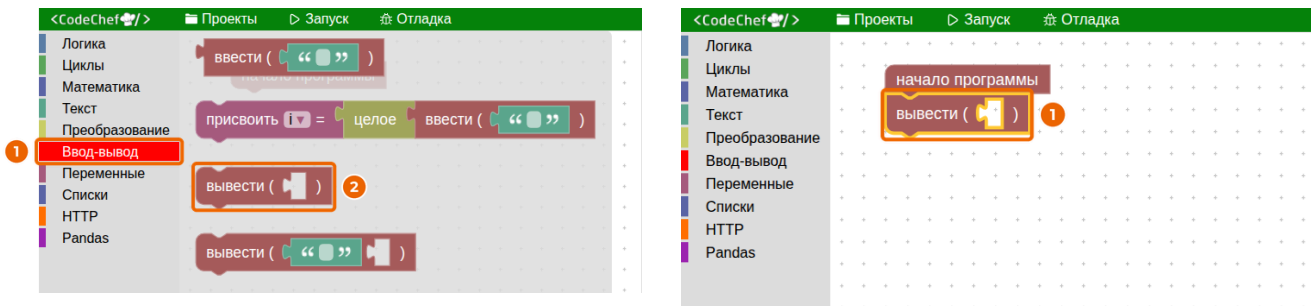


Рисунок В.2 – Слева – группа «Ввод-вывод» и блок «вывести» (2), справа – блок защёлкнутся под «началом программы» (1)

- Число или слово в блоке меняют щелчком по полю: щёлкни, впиши, нажми Enter. Стрелка ▼ открывает список – выбери нужное.
- Блок не защёлкнут – он чуть отъезжает и висит сам по себе. Перетащи его ещё раз.
- Блок тянет за собой всё, что под ним. Лишний блок: щёлкни по нему и нажми Delete.
- Поле двигают мышью за пустое место. Если в углу всплыла английская подсказка «Use → to navigate inside of blocks» – закрой её крестиком, на программу она не влияет.

Шаг 4. Переменные. Переменная – это коробочка с именем, в которую программа кладёт число, слово или список. В группе «Переменные» нажми «Создать переменную...», впиши имя и нажми Enter.

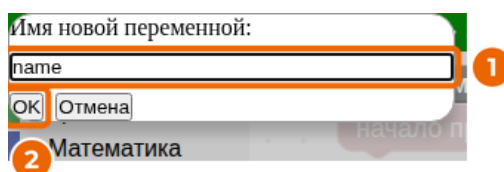


Рисунок В.3 – Окно создания переменной: имя (1), OK (2)

Имя пиши латиницей: `name`, `total`, `marks`. Русские буквы CodeChef превращает в код вида `_D0_BE_D1_86...`: программа работает, но читать её и смотреть переменные при отладке невозможно. В группе три блока: «присвоить ... =» (положить значение), «увеличить ... на» и блок с самим именем (взять значение).

Шаг 5. Ввод и запуск. Ответы на вопросы программа берёт не из окна, а с вкладки «Ввод»: впиши их заранее, по одному в строке (1 – вкладка, 2 – поле). Потом нажми «Запуск» вверху и смотри результат на вкладке «Вывод».

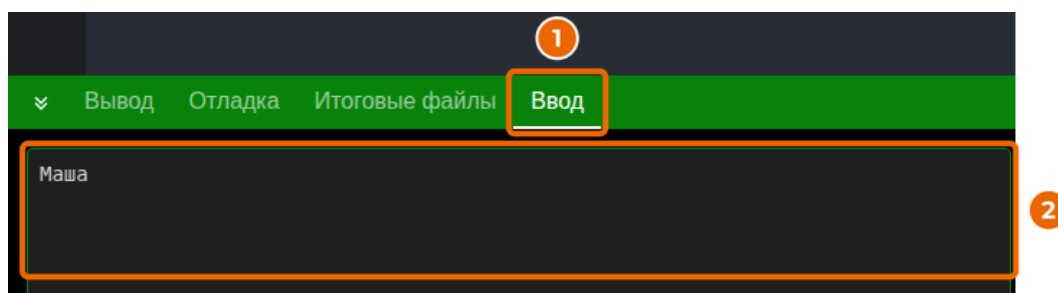


Рисунок В.4 – Вкладка «Ввод»

Если ответов меньше, чем вопросов, программа остановится с сообщением `E0FError: EOF when reading a line` – допиши строки. Если там, где нужно число, стоит слово, будет

`ValueError: invalid literal for int()` – впиши цифры. Зависшую программу (например, бесконечный цикл) останавливает крестик вверху справа, рядом с надписью «Выполняется».

Шаг 6. Сохранить и открыть. Всё, что ты собрал, живёт только на этой странице: обновил её или закрыл вкладку – поле пустое. Поэтому сохраняйся.

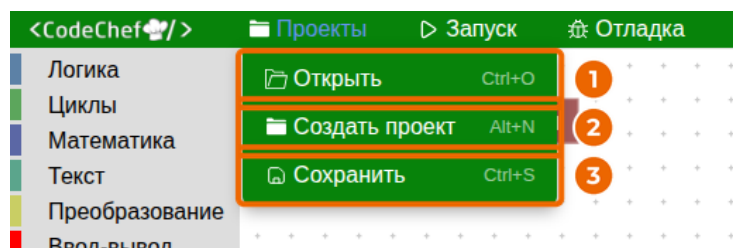


Рисунок В.5 – Меню «Проекты»: открыть (1), создать (2), сохранить (3)

«Сохранить» (3) скачивает файл `project.chef` : переименуй его, чтобы не перепутать с другими. «Открыть» (1) просит выбрать такой файл – так открывают и проект, который дают в курсе. «Создать проект» (2) очищает поле и сначала переспрашивает: «Все несохранённые изменения будут потеряны».

Задача П.1. Привет. Программа спрашивает имя и здороваются. Создай переменную `name` . Из «Переменных» возьми «присвоить ... =», поставь под «начало программы» и выбери в списке `name` . Из «Ввод-вывод» возьми верхний блок «ввести ()» и вставь его в гнездо справа от «=»; в кавычках впиши вопрос `Как тебя зовут?` (пробел в конце оставь, чтобы ответ не слился с вопросом). Под ним поставь блок «вывести» с текстом и пустым гнездом, впиши `Привет,` , а в пустое гнездо вставь блок `name` из «Переменных».

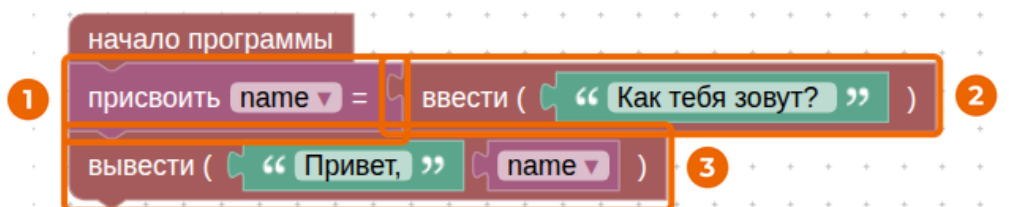


Рисунок В.6 – Программа П.1: переменная (1), вопрос (2), ответ на экран (3)

Открой вкладку «Ввод», впиши любое имя и нажми «Запуск».

```

1 name = None
2
3
4 name = input('Как тебя зовут? ')
5 print('Привет, ', name)

```

Вывод: Как тебя зовут? Маша
Привет, Маша

Рисунок В.7 – Результат: справа в коде – те же команды (1), в выводе – вопрос с твоим ответом и приветствие (2)

Что должно получиться: в «Вывод» две строки – Как тебя зовут? Маша (твое имя после вопроса) и Привет, Маша. В коде справа – `name = input('Как тебя зовут? ')` и `print('Привет, ', name)`. Впиши другое имя и запусти ещё раз: вторая строка изменится.

Задача П.2. Оценки: сколько, сумма, среднее. Программа просит три оценки, складывает их в список и считает. Создай переменные `marks` (список), `total` (сумма) и `x` (очередная оценка). Собирай по кадру сверху вниз.

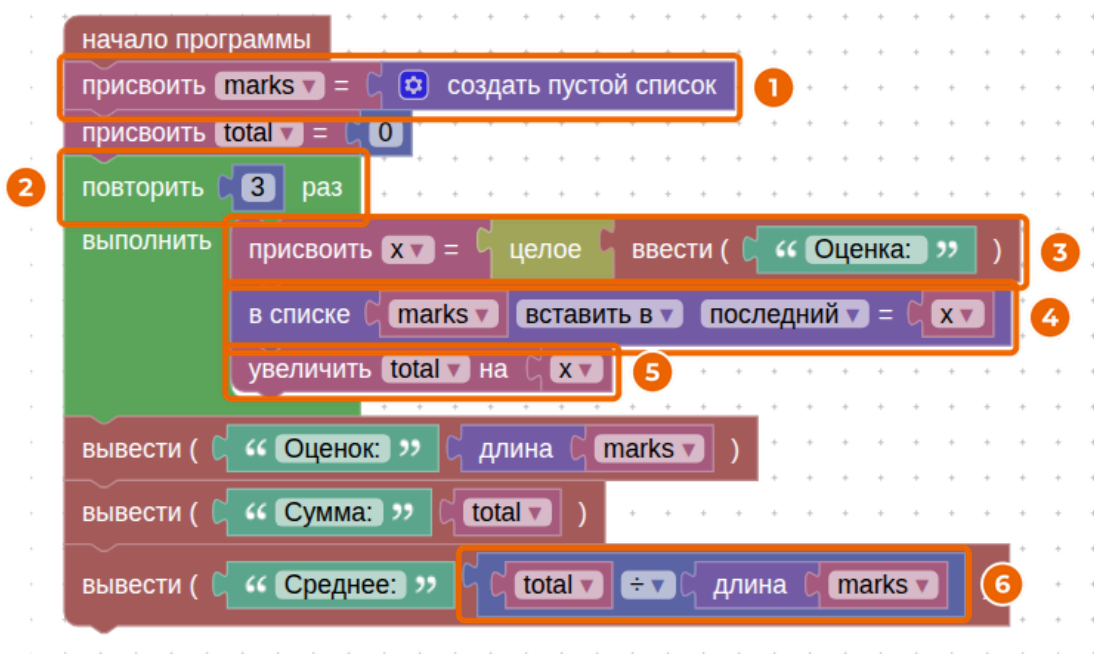


Рисунок В.8 – Программа П.2

1. «присвоить `marks` =» и блок «создать список из» из группы «Списки». Нужен пустой список, поэтому нажми на шестерёнку и потяни верхний «элемент» в правой части

влево, в серое поле: вместе с ним уйдут и нижние, а надпись станет «создать пустой список».

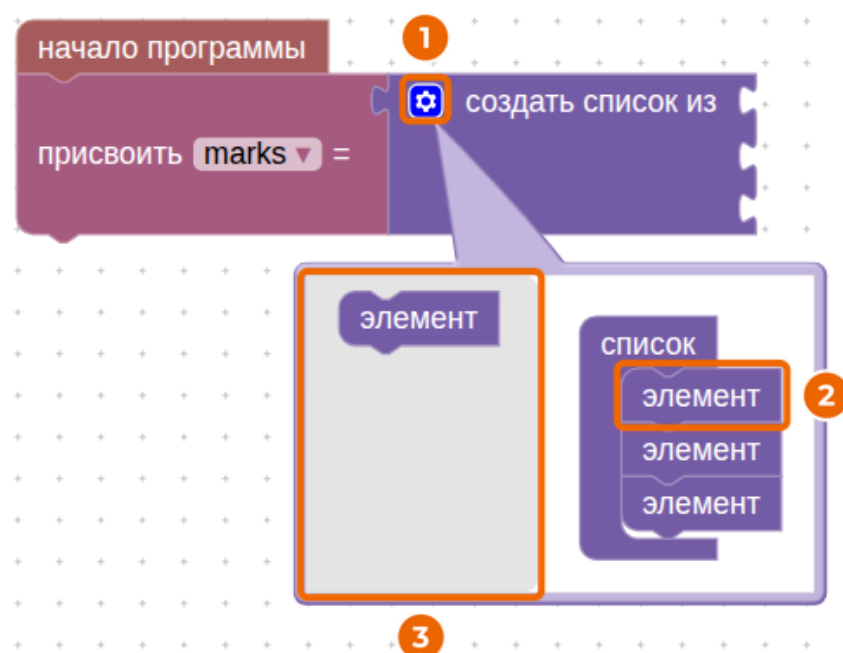


Рисунок В.9 – Шестерёнка: сама шестерёнка (1), «элемент», который тянут (2), серое поле, куда его тянут (3)

Под ним – «присвоить `total` =» с числом 0 из «Математики». (2) Из «Циклов» – «повторить ... раз» и число 3 (первый блок «Математики» – впиши в него 3). Внутри, в «выполнить», ставь по очереди три блока. (3) Из «Ввод-вывод» – второй блок, «присвоить ... = целое ввести ()»: выбери в нём `x` и впиши вопрос `Оценка: .` Слово «целое» превращает ответ в число. (4) Из «Списков» – «в списке ... присвоить № ... =»: выбери «вставить в» и «последний», слева поставь `marks`, справа `x` – оценка ляжет в конец списка. (5) «увеличить `total` на»: выбери `total`, а число 1 замени блоком `x`.

После цикла – три блока «вывести». Первый: текст `Оценок:` и блок «длина» из «Списков» со списком `marks`. Второй: `Сумма:` и `total`. Третий (6): `Среднее:` и блок «+» из «Математики» – выбери в нём `÷`, слева поставь `total`, справа «длину» списка `marks`.

На вкладке «Ввод» впиши три строки: `5`, `4`, `3` – и нажми «Запуск».

```

1  from numbers import Number
2
3  marks = None
4  total = None
5  x = None
6
7
8  marks = []
9  total = 0
10 for count in range(3):
11     x = int(input('Оценка: '))
12     marks.append(x)
13     total = (total if isinstance(total, Number) else 0) + x
14     print('Оценок:', len(marks))
15     print('Сумма:', total)
16     print('Среднее:', total / len(marks))

```

Вывод:

```

Оценка: 5
Оценка: 4
Оценка: 3
Оценок: 3
Сумма: 12
Среднее: 4.0

```

Рисунок В.10 – Результат П.2: вывод (1) и тот же код на Python (2)

Что должно получиться: Оценок: 3, Сумма: 12, Среднее: 4.0. Попробуй 5, 5, 4: сумма 14, а среднее – 4.666666666666667. Длинный хвост – это нормально, округлять здесь нечем. Проверь себя: $5 + 4 + 3 = 12$, $12 : 3 = 4$.

Задача П.3. Копилка. В копилке 0 рублей, каждый день в неё кладут 15. Через сколько дней накопится хотя бы 100? Создай `money` и `days`, обеим дай значение 0 («присвоить ... =», число 0). Из «Циклов» возьми «повторять, пока» и вставь в его гнездо блок сравнения из «Логики»: выбери в нём знак `<`, слева `money`, справа число 100. Внутри цикла поставь два блока «увеличить»: `money` на 15 (число 1 замени на 15) и `days` на 1. После цикла выведи Дней: с `days` и В копилке: с `money`.

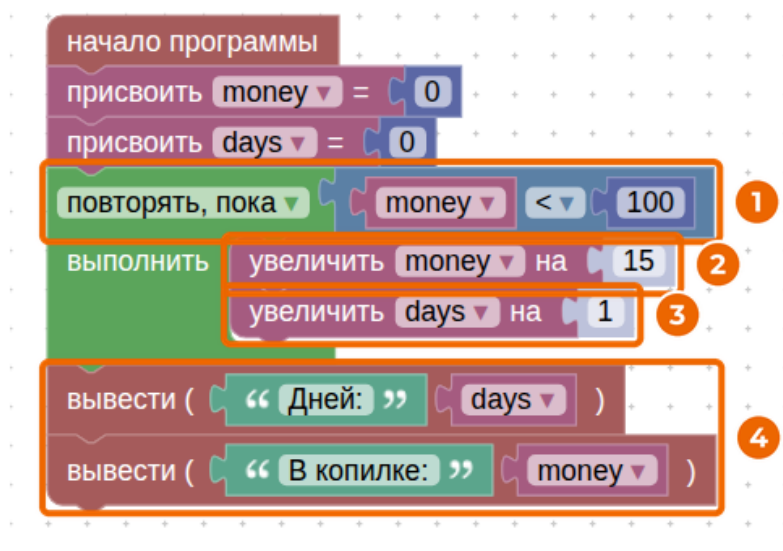


Рисунок В.11 – Программа П.3: условие (1), копим деньги (2), считаем дни (3), ответ (4)

Что должно получиться: Дней: 7 и В копилке: 105. Проверь пальцем: за 6 дней накопится 90 – это ещё меньше 100, значит нужен седьмой. Если вместо 15 по ошибке оставить 0, копилка не растёт и цикл не кончится никогда – останови его крестиком сверху справа.

Теперь посмотри, как цикл крутится. В узкой колонке слева от кода, правее номеров строк, щёлкни напротив строки `days = ...` – появится красная точка (1): здесь программа остановится. Нажми «Отладка» вверху и открой вкладку «Отладка»: в таблице будут `money` и `days` (2). «Продолжить» (3) пускает программу до следующей остановки, «Шаг вперед» – на одну строку, «Завершить» – конец. Строка `Number` в таблице служебная, её не читай. Крестик (4) тоже останавливает программу.

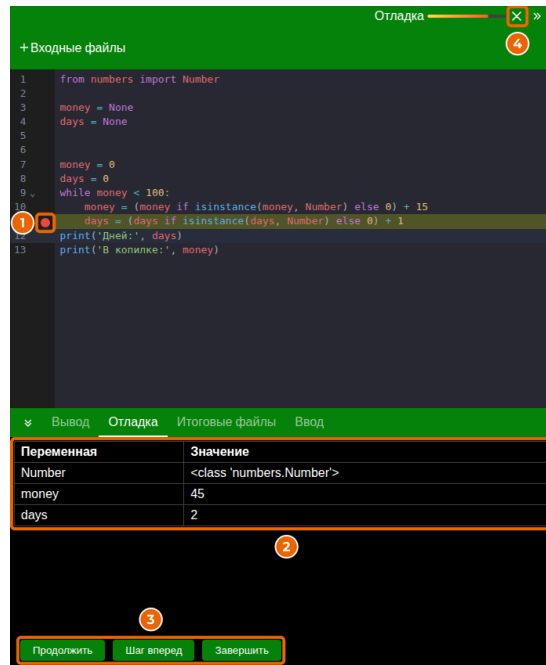


Рисунок В.12 – Отладка: точка останова (1), таблица переменных (2), кнопки (3), крестик (4)

Что должно получиться: при первой остановке `money` равно 15, а `days` – 0 (до `days` программа ещё не дошла). После двух «Продолжить» – `money` 45 и `days` 2: деньги растут по 15 за круг, дни – по 1.

Задача П.4. Таблица по страницам. Сервер отдаёт 200 записей страницами по 50. Программа обойдёт все четыре страницы, склеит их в одну таблицу и сохранит файл. Так же устроена программа курса, только там страниц больше и другой адрес. Для тренировки возьмём открытый учебный сервер с адресом <https://jsonplaceholder.typicode.com/todos>: он отдаёт 200 дел, у каждого четыре поля – `userId`, `id`, `title` (заголовок) и `completed` («сделано»). Параметр `_page` – номер страницы, `_limit` – сколько записей на странице. Нужен интернет. Заранее создай переменные `df_list`, `df`, `data` и `i`.

Главный блок задачи – «HTTP GET» из группы «HTTP»: он ходит на сервер за данными.

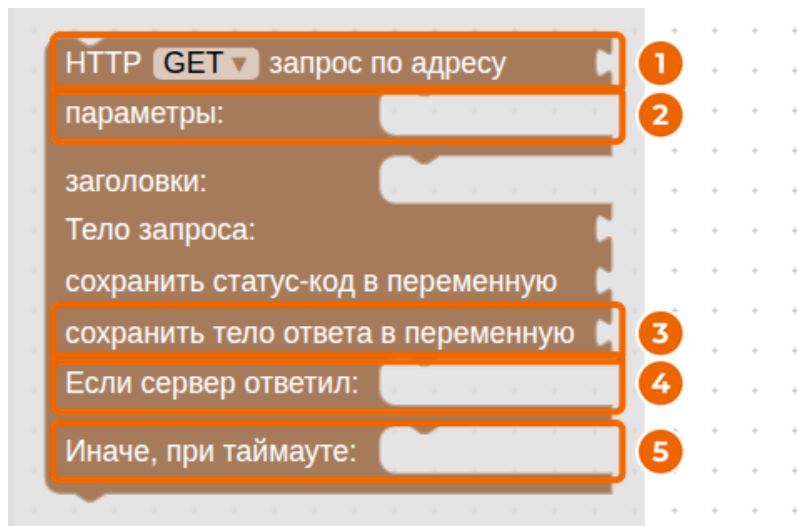


Рисунок В.13 – Блок «HTTP GET» в палитре: адрес (1), параметры (2), тело ответа (3), «Если сервер ответил» (4), «Иначе» (5)

Начало. Из «HTTP» – «подключить библиотеку requests», из «Pandas» – «подключить библиотеку pandas как pd» (1 на кадре ниже). Первая ходит за данными в интернет, вторая умеет работать с таблицами. Потом «присвоить `df_list =`» и пустой список, как в П.2 (2).

Цикл. Из «Циклов» – «цикл по ... от ... до ... шаг ...»: выбери переменную `i` (в новом блоке стоит другая буква), впиши числа от 1, до 5, шаг 1 (3). Последнее число в перебор не входит, поэтому для четырёх страниц пишем 5.

Запрос. Внутри цикла – «HTTP GET». В адрес (4) вставь текстовый блок с адресом. В «параметры» (5) поставь два блока «HTTP-параметр» друг под другом: ключ `_limit` со значением 50 (оба – пустые кавычки из группы «Текст») и ключ `_page`, а значение – переменная `i`. В «сохранить тело ответа в переменную» (6) вставь `data`. Остальные гнёзда оставь пустыми.

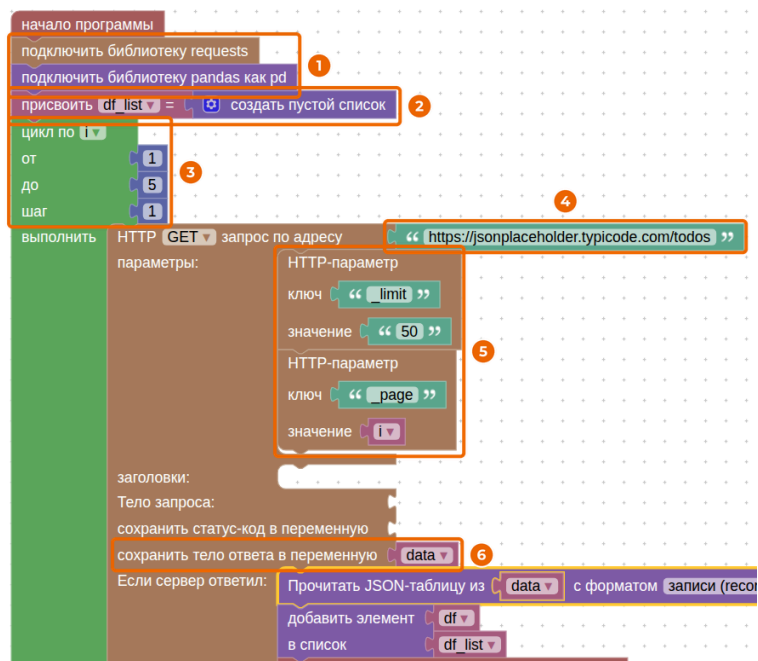


Рисунок В.14 – Начало программы П.4

Ответ сервера. В ветке «Если сервер ответил»: «Прочитать JSON-таблицу из `data` с форматом записи (records) в переменную `df`» из «Pandas» (7), потом «добавить элемент `df` в спи-

сок `df_list` » (8) из «Pandas» и «вывести» с текстом Готова страница и `i` (9). В ветке «Иначе, при таймауте» – «вывести» Ошибка (10). Ветки лежат близко: если блок прыгает не в ту, подними или опусти его чуть выше или ниже.

После цикла – не внутри него. «присвоить `df` => «объединить список таблиц `df_list` в одну таблицу» (11); «вывести» с блоком «первые 3 строк таблицы `df` » из «Pandas» (12); «показать информацию о таблице `df` » (13); «сохранить таблицу `df` в CSV-файл с именем» `todos.csv` (14).

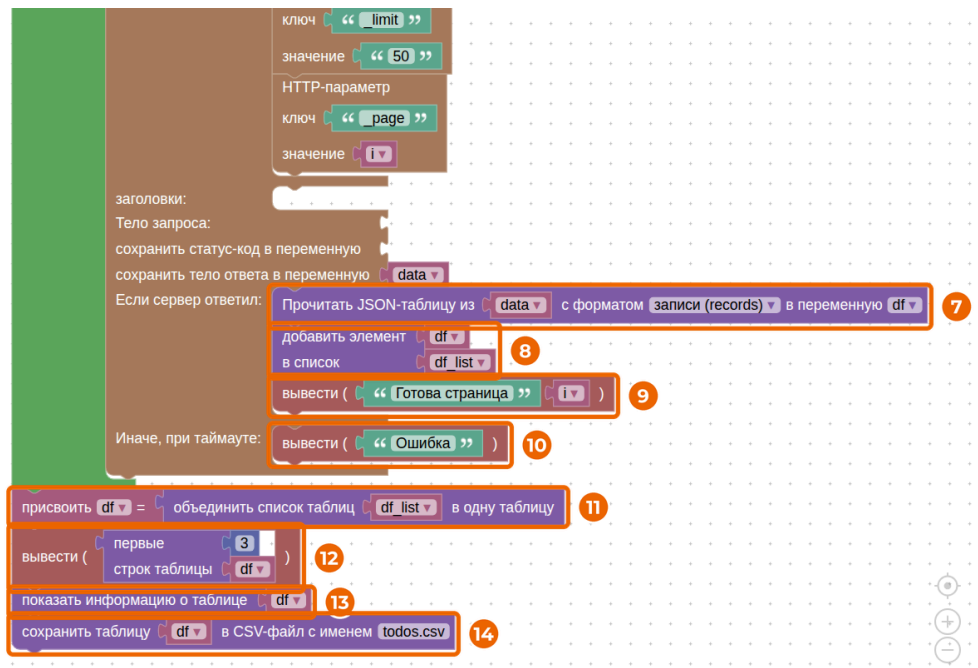


Рисунок В.15 – Конец программы П.4

Нажми «Запуск». Несколько секунд программа ходит за страницами, потом печатает результат.

```

Вывод
Готова страница 1
Готова страница 2
Готова страница 3
Готова страница 4
 1
 2
 3
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558

```

Датавиз-конструктор: таблица, статистика, диаграммы

Датавиз-конструктор – программа для исследования данных без кода: загружаешь таблицу, смотришь статистику, строишь диаграммы, собираешь дашборды и «вычислительные эссе» – диаграммы с пояснениями. В курсе сферы ИИтех в нём делают задания на наборе пассажиров космического лайнера Spaceship Titanic. Кадры сняты в версии 2.0.0 для Windows.

Шаг 1. Поставить. Инструкция разработчиков – teletype.in/@irk.digitalwave/dataviz-install, ссылка на неё есть и в курсе сферы (модуль «Исследовательский анализ данных»). Программа работает только под Windows и ставится без установщика: распаковал архив – запускаешь `DatavizConstructor.exe`.

Шаг 2. Окно программы.

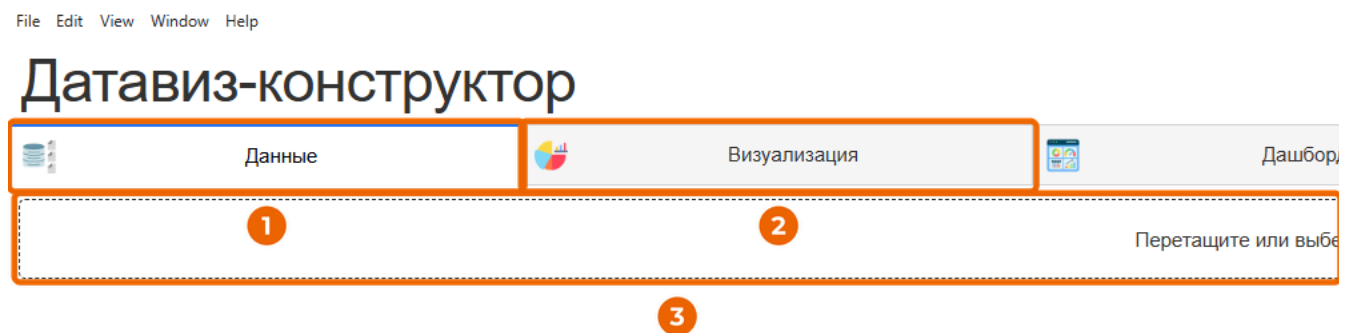


Рисунок В.17 – Окно Датавиз-конструктора, левая часть: вкладки и зона загрузки

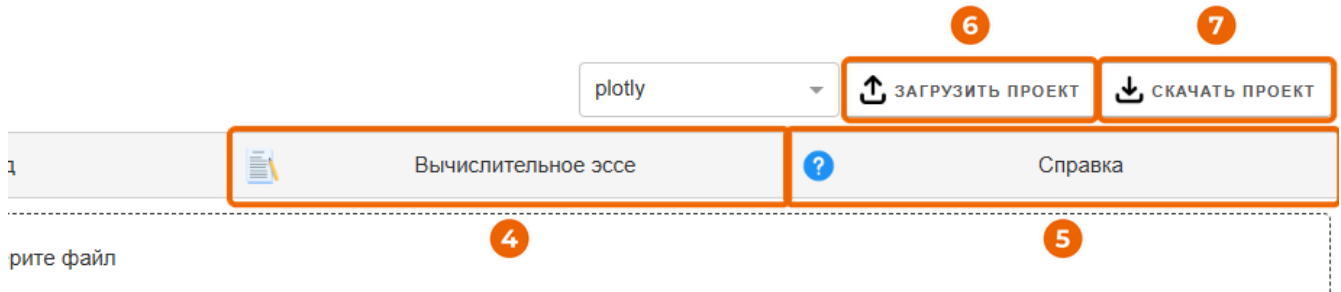


Рисунок В.18 – Окно Датавиз-конструктора, правая часть: вкладки и кнопки проекта

1. **Данные** – загрузить таблицу, отфильтровать, посмотреть статистику, скачать.
2. **Визуализация** – листы с диаграммами: «+» добавляет лист.
3. **Зона загрузки** – перетащи сюда файл `.csv` или щёлкни и выбери его.
4. **Вычислительное эссе** – диаграммы и текст одной лентой, как отчёт.
5. **Справка** – подсказки по программе.
6. **Загрузить проект** – открыть сохранённую работу (файл `.dviz`).
7. **Скачать проект** – сохранить всю работу в файл `.dviz`. Сохраняй почаще.

Между вкладками «Визуализация» и «Вычислительное эссе» есть ещё «Дашборд» – несколько диаграмм на одном экране.

Задача П.1. Сколько пассажиров без возраста? Скачай в курсе файл `train.csv` и загрузи его во вкладке «Данные».

СКРЫТЬ/ПОКАЗАТЬ ПОЛЯ

	NA	PassengerId	HomePlanet	CryoSleep	Cabin	Destination	Age	VIP
	filter data.							
x	0	101	Europa	0	B/0/P	TRAPPIST-1e	39	
x	0	201	Earth	0	F/0/S	TRAPPIST-1e	24	
x	0	301	Europa	0	A/0/S	TRAPPIST-1e	58	
x	0	302	Europa	0	A/0/S	TRAPPIST-1e	33	
x	0	401	Earth	0	F/1/S	TRAPPIST-1e	16	
x	0	501	Earth	0	F/0/P	PSO J318.5-22	44	
x	0	601	Earth	0	F/2/S	TRAPPIST-1e	26	
x	1	602	Earth	1	G/0/S	TRAPPIST-1e	28	
x	0	701	Earth	0	F/3/S	TRAPPIST-1e	35	
x	0	801	Europa	1	B/1/P	55 Cancri e	14	
x	1	802	Europa	1	B/1/P	TRAPPIST-1e	34	
x	0	803	Europa	0	B/1/P	55 Cancri e	45	
x	0	901	Mars	0	F/1/P	TRAPPIST-1e	32	
x	0	1001	Earth	0	G/1/S	TRAPPIST-1e	48	
x	0	1101	Earth	0	F/2/P	TRAPPIST-1e	28	

Рисунок В.19 – Таблица загружена: служебный столбец NA

Программа сама добавила первым столбец NA (1): в нём 1, если в строке есть хоть один пропуск, и 0, если пропусков нет. Сами пустые ячейки подсвечены красным.

483	0	291	0	Sandie Hi
1539	3	0	0	Billex Jaci
0	0	0		Candra Jaci
785	17	216	0	Andon
0	0	0	0	Erraia
0		0	0	Altard
7295	589	110	124	Wezen

Рисунок В.20 – Пустые ячейки подсвечены красным

Над таблицей – кнопки: «Статистика» (1), поле «Название файла» (2) и «Скачать» (3). Нажми «Статистика».

ДАННЫЕ

1 СТАТИСТИКА

2 Название файла

ЭНКОДЕР

3 СКАЧАТЬ

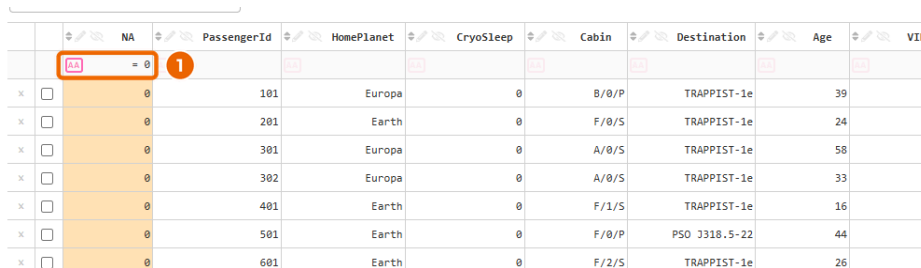
Рисунок В.21 – Кнопки над таблицей

index	Age	RoomService
count	8514	8512
mean	28.8279304675	224.6876174812
std	14.4890214239	666.7176629281
min	0	0
25%	19	0
50%	27	0
75%	38	47
max	79	14327

Рисунок В.22 – Статистика: count у Age (1)

Что должно получиться: строк в таблице 8693, а у Age count (1) – 8514. **count считает только непустые значения**, поэтому возраст не указан у 8693 – 8514 = 179 пассажиров. Посмотри count у других столбцов – пропуски есть почти везде.

Задача П.2. Чистая таблица. Вернись к таблице кнопкой «Данные». В строке фильтра под заголовком NA (1) впиши = 0 и нажми Enter – останутся строки без пропусков. Порядок строк и столбцы не трогай.



The screenshot shows a table with columns: NA, PassengerId, HomePlanet, CryoSleeper, Cabin, Destination, Age, and VIP. The NA column has a filter dropdown set to 0. The table contains 7 rows of data, all of which have a value of 0 in the NA column.

	NA	PassengerId	HomePlanet	CryoSleep	Cabin	Destination	Age	VIP
x	0	101	Europa	0	B/0/P	TRAPPIST-1e	39	
x	0	201	Earth	0	F/0/S	TRAPPIST-1e	24	
x	0	301	Europa	0	A/0/S	TRAPPIST-1e	58	
x	0	302	Europa	0	A/0/S	TRAPPIST-1e	33	
x	0	401	Earth	0	F/1/S	TRAPPIST-1e	16	
x	0	501	Earth	0	F/0/P	PSO 3318.5-22	44	
x	0	601	Earth	0	F/2/S	TRAPPIST-1e	26	

Рисунок В.23 – Фильтр NA = 0

Впиши название файла (например, train_clean) и нажми «Скачать»: Windows спросит, куда сохранить, – имя (1) и «Сохранить» (2).

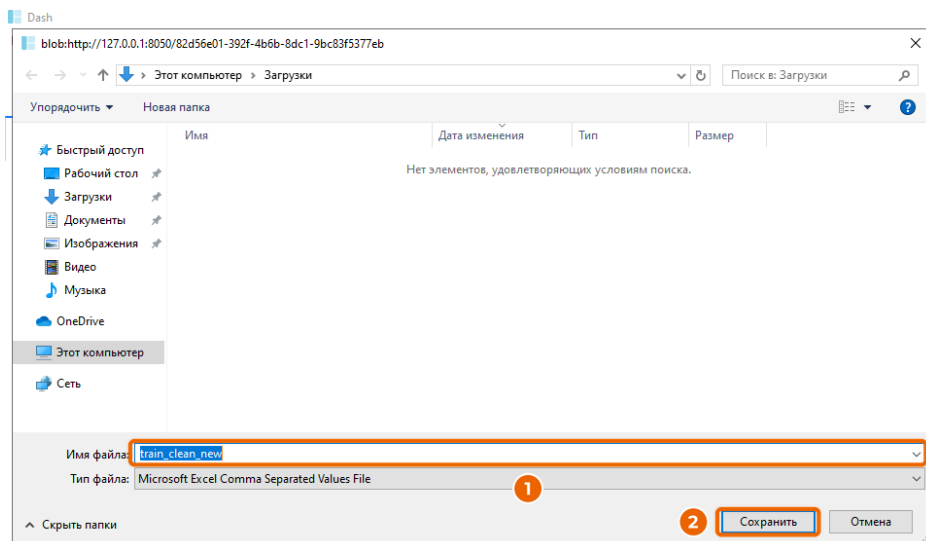
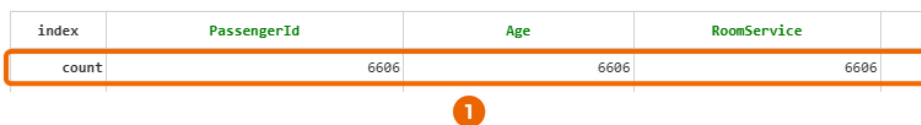


Рисунок В.24 – Сохраняем отфильтрованную таблицу

Загрузи сохранённый файл снова и открой «Статистику».



The screenshot shows a table with columns: index, PassengerId, Age, and RoomService. The table contains 4 rows of data. The 'count' row is highlighted with a red box and labeled with a red circle 1.

index	PassengerId	Age	RoomService
count	6606	6606	6606

Рисунок В.25 – В чистой таблице count одинаковый у всех столбцов

Что должно получиться: count у всех столбцов – **6606** (1). Ровно столько строк должно остаться, так проверяет себя и курс. Не совпало – в курсе есть готовый файл data_new.csv, работай с ним. Важно: фильтр таблицы не действует на диаграммы – поэтому чистый файл и загружают заново.

Задача П.3. Кого «унесло» чаще? В столбце Transported 1 – пассажира переместило, 0 – нет; в CryoSleeper 1 – он был в криосне. Вкладка «Визуализация» → «Выберите тип диаграммы» →

«Столбчатая» (1).

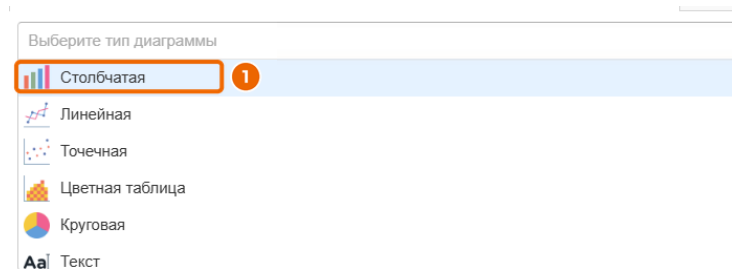


Рисунок В.26 – Тип диаграммы

Ось X – CryoSleep (1), ось Y – Transported (2), агрегация (3) пока «Сумма».

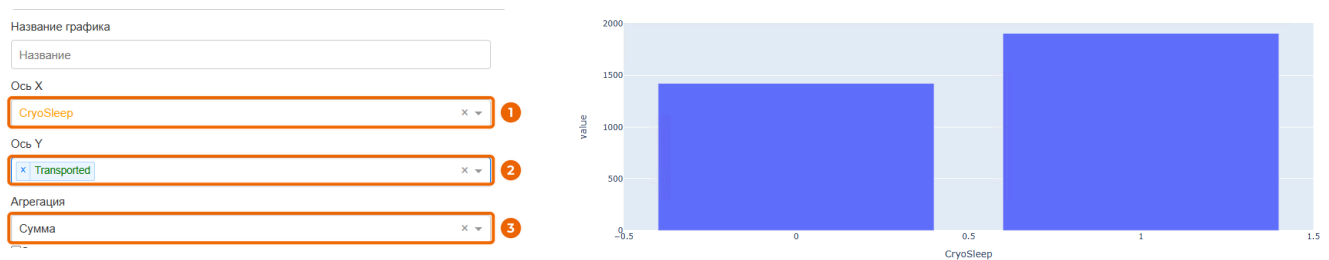


Рисунок В.27 – Слева – настройки диаграммы, справа – агрегация «Сумма»: сколько пассажиров переместило

Теперь смени агрегацию на «Среднее» (1).

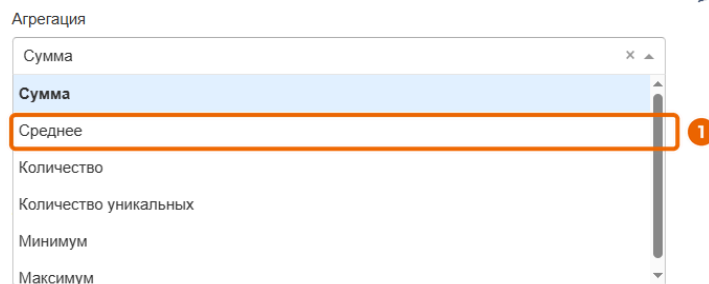


Рисунок В.28 – Агрегация «Среднее»

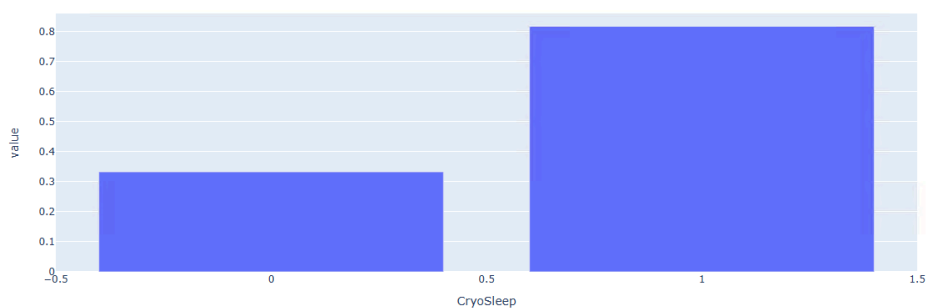


Рисунок В.29 – Среднее Transported: доля перемещённых

Что должно получиться: с «Суммой» столбцы показывают, **сколько** пассажиров переместило – около 1420 без криосна и около 1900 в криосне. Со «Средним» – **какая доля**: 0,33 и 0,82. Среднее столбца из нулей и единиц – это и есть доля единиц. Вывод: из спавших в криосне перемещены 82 % пассажиров, из бодрствовавших – только треть. Сравнить группы разного размера честнее по доле, а не по сумме.

GraphNet: собираем нейросеть из блоков

GraphNet – конструктор нейросетей: сеть собирают из блоков и соединяют их линиями, а учит её библиотека Keras. На сфере ИИтех с него начинают модуль про нейросети. Кадры сняты в версии 0.1.0 для Windows. В курсе модуль GraphNet открывается позже (4 октября), поэтому здесь – первая сеть из раздела 7.3: линейная регрессия $y = w \cdot x + b$.

Шаг 1. Поставить. Инструкция автора – teletype.in/@yupest/graphnet: там ссылка на файл GraphNet_v010.exe (около 650 МБ). Установка не нужна – скачал и запустил. Windows может предупредить о неизвестном издателе: «Подробнее» → «Выполнить в любом случае» (вместе со взрослым). Первый запуск долгий – программа распаковывается 15–40 секунд. Процессор Pentium или Celeron не подойдёт.

Шаг 2. Окно. Слева – список блоков (1), на поле уже стоит блок Input (2) – вход сети. Внизу кнопка «Собрать модель»: она соединяет блоки, обучает сеть и выполняет всё, что нарисовано.

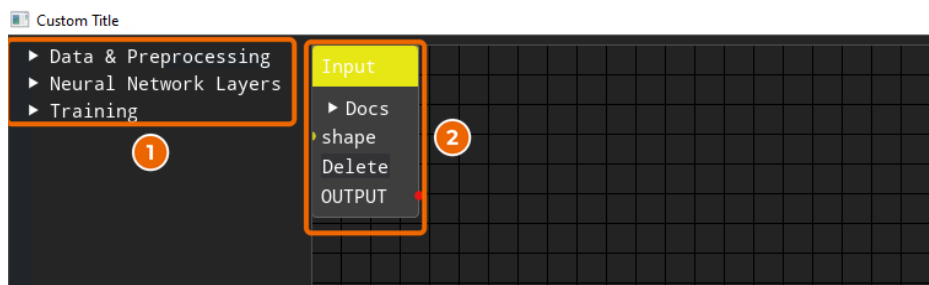


Рисунок В.30 – Окно GraphNet: список блоков (1) и блок Input (2)

Разверни разделы списка треугольниками – увидишь все блоки. Блок ставят на поле, **перетаскивая мышью** (щелчок по названию ничего не добавляет). Удалить – кнопка Delete в самом блоке.

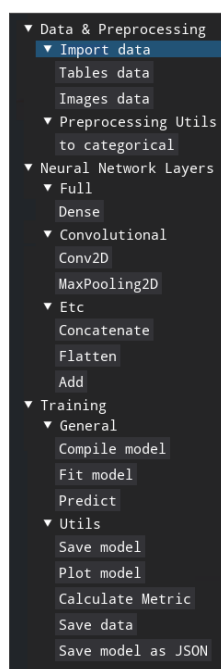


Рисунок В.31 – Все блоки GraphNet

- **Tables data** – загрузить таблицу из файла; **Images data** – папку с картинками.

- **Dense** – полносвязный слой нейронов; **Conv2D**, **MaxPooling2D** – слои для картинок.
- **Compile model** – выбрать, как учить: оптимизатор и функцию ошибки.
- **Fit model** – обучить: что подавать на вход (x), какие правильные ответы (y), сколько эпох.
- **Predict** – прогноз обученной сети на новых данных; **Calculate Metric** – посчитать точность.
- **Save model**, **Save data** – сохранить модель и результаты в файл.

Важно. Схему блоков GraphNet пока не сохраняет: закрыл программу – схему придётся собирать заново. Сделай снимок экрана, а обученную модель сохрани блоком Save model.

Задача П.1. Сколько вёдер на грядки. Дачник поливает грядки: на каждую уходит 3 ведра и ещё 1 ведро на дорожку – $y = 3x + 1$. Пусть сеть сама найдёт эти числа. Создай в Блокноте три файла, по одному числу в строке, без заголовка: `x.csv` – числа от 1 до 10 (грядки); `y.csv` – 4, 7, 10 и так до 31 (ведра); `new.csv` – 12 и 25 (новые грядки). Собери схему:

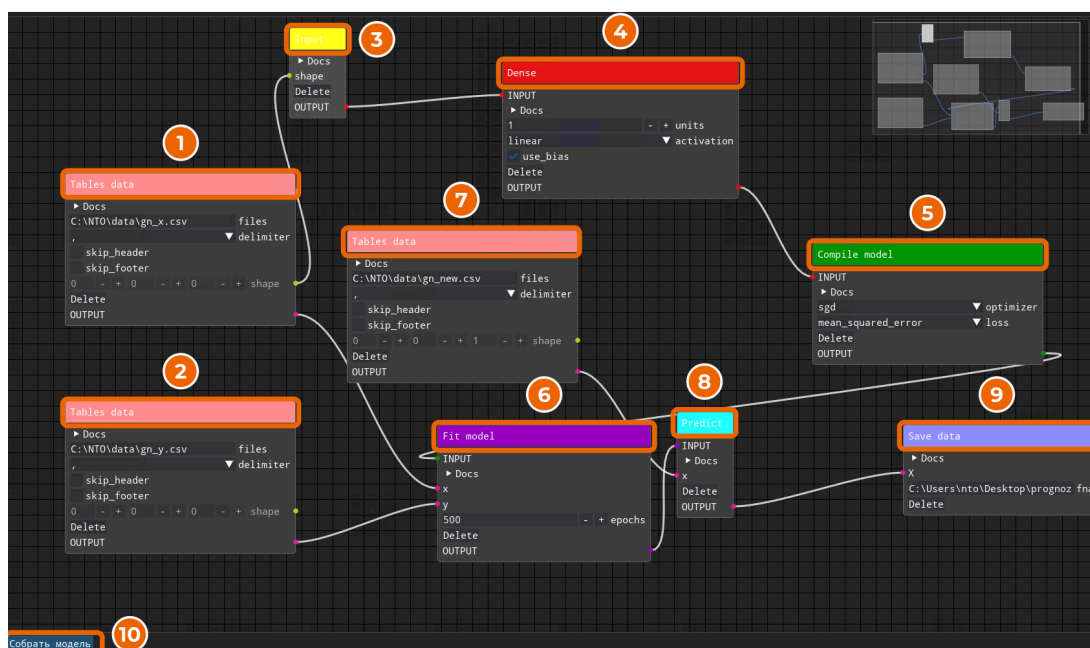


Рисунок В.32 – Собранная сеть: порядок блоков – как номера

1. **Tables data** с файлом `x.csv` – в поле `files` впиши полный путь, например `C:\Users\Ученик\Desktop\x.csv` (1 на рисунке ниже). Её выход **shape** (2) соедини со входом **shape** блока **Input** (3) – так сеть узнает размер входа.
2. **Tables data** с файлом `y.csv` – правильные ответы.
3. **Input** – уже стоит на поле.
4. **Dense**: `units` – 1 (один нейрон), `activation` – `linear`, галочка **use_bias** – чтобы у нейрона было смещение b . Соедини **OUTPUT** блока **Input** со входом **INPUT** блока **Dense**.
5. **Compile model**: `optimizer` – `sgd`, `loss` – `mean_squared_error` (средний квадрат ошибки). Вход – от **OUTPUT** **Dense**.
6. **Fit model**: вход **INPUT** – от **Compile model**, `x` – от таблицы `x.csv`, `y` – от таблицы `y.csv`, `epochs` – 500.
7. **Tables data** с файлом `new.csv` – новые грядки для прогноза.
8. **Predict**: **INPUT** – от **Fit model**, `x` – от таблицы `new.csv`.
9. **Save data**: `X` – от **Predict**, в поле имени файла – полный путь, например `C:\Users\Ученик\Desktop\prognoz.txt`.

10. Нажми «Собрать модель».

Линию тянут мышью от цветной точки-выхода одного блока к точке-входу другого.

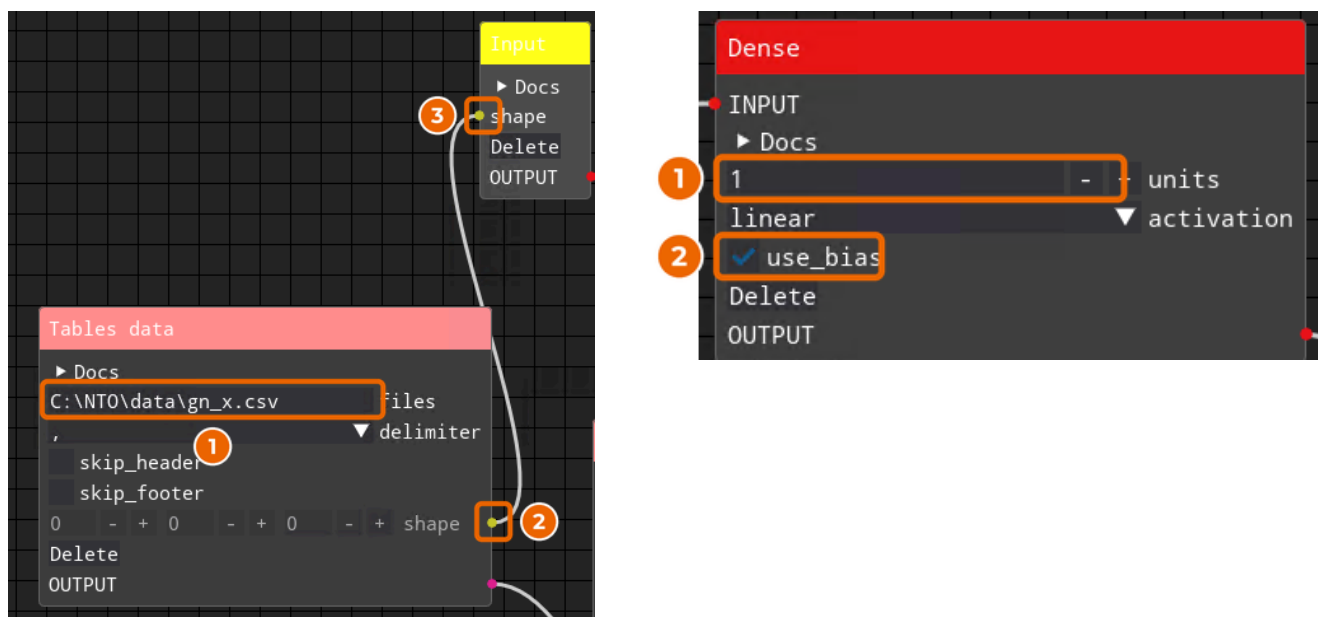


Рисунок В.33 – Слева – таблица и вход: путь к файлу (1), выход shape (2), вход shape у Input (3); справа – Dense: units (1), use_bias (2)

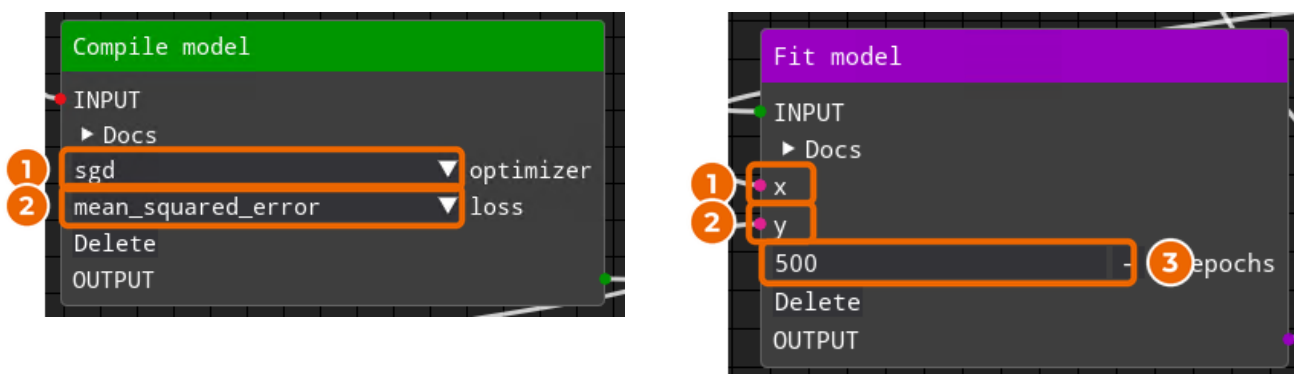


Рисунок В.34 – Слева – Compile model: оптимизатор (1) и ошибка (2); справа – Fit model: x (1), y (2), эпохи (3)

Что должно получиться: на несколько секунд появится окошко «Обучение», потом в файле `prognoz.txt` окажутся два числа – около 37 и около 76. У нас вышло 37,06 и 76,21: правильные ответы $3 \cdot 12 + 1 = 37$ и $3 \cdot 25 + 1 = 76$. Значит, сеть нашла $w \approx 3$ и $b \approx 1$ сама, по десяти примерам. Числа в файле записаны так: `3.705e+01` – это $3,705 \cdot 10^1 = 37,05$.

Задача П.2. Оптимизатор решает. В `Compile model` смени `sgd` на `adam` (1), эпохи оставь 500, снова «Собери модель» и открой `prognoz.txt`.

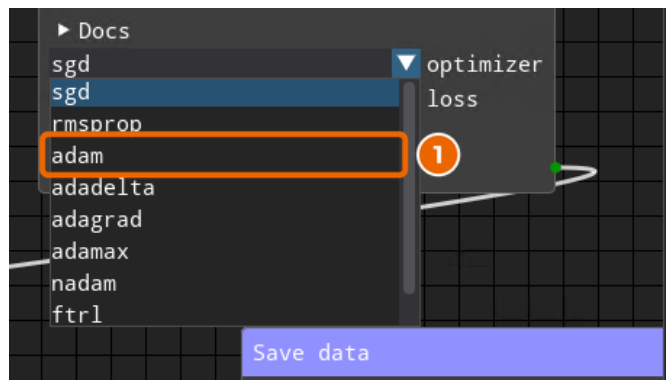


Рисунок В.35 – Список оптимизаторов

Что должно получиться: прогноз намного меньше правильного (у нас вышло 7,8 и 15,8 вместо 37 и 76; у тебя будут другие числа – веса сети в начале случайные). Adam с настройками по умолчанию делает слишком маленькие шаги и за 500 эпох не успевает дойти до $w = 3$. Поставь 5000 эпох – прогноз заметно приблизится (у нас 34,8 и 69,6), но учиться сеть будет несколько минут. Вывод для отбора: обученность сети зависит и от оптимизатора, и от числа эпох (раздел 7.3).

Teachable Machine: научи компьютер отличать круг от квадрата

Teachable Machine (teachablemachine.withgoogle.com) – бесплатный сервис Google: модель-классификатор в нём собирают мышкой, без единой строчки кода. Работает в браузере, регистрироваться не нужно. В курсе сферы 2026/27 Teachable Machine не используется – это инструмент кружка, чтобы руками понять, как работает классификация. Ниже весь путь по шагам и четыре задачи. Рамка с номером на картинке – куда нажимать.

Надписи в сервисе – на языке твоего браузера. Если они английские, внизу страницы есть выпадающий список языков – выбери «русский». Главные кнопки по-английски: «Get Started» – «Начать», «Image Project» – «Проект с изображениями», «Train Model» – «Обучить модель», «Advanced» – «Дополнительные», «Under the hood» – «Дополнительные сведения», «Preview» – «Посмотреть», «Input» – «Ввод», «Output» – «Выходные данные».

Шаг 0. Подготовка картинки. Веб-камера не нужна: примеры мы загрузим файлами. Нарисуй их сам в Paint или в любом редакторе, где можно сохранить PNG или JPG.

- Холст квадратный, например 400 × 400, фон белый: сервис всё равно обрежет картинку до квадрата.
- На холсте одна закрашенная фигура. Инструмент «Овал» с зажатой клавишей Shift рисует круг, «Прямоугольник» с Shift – квадрат. Овал без Shift тяни так, чтобы одна сторона была в полтора-два раза длиннее другой.
- У каждой картинки свои размер, место на холсте и цвет (красный, синий, зелёный, жёлтый, фиолетовый). Цвета и размеры у кругов и квадратов должны быть одинаково разными, чтобы по ним нельзя было угадать фигуру.
- Нужно четыре папки: «круг» – 30 картинок, «квадрат» – 30, «овал» – 30 и «проверка» – по 5 новых кругов, квадратов и овалов, которые в классы мы не положим (назови их «новый-круг-1», «новый-квадрат-1» и так далее).

Рисовать 105 картинок в одиночку долго, поэтому разделите работу в кружке. Если рисовать лень, можно фотографировать: крышки и монеты – круги, стикеры – квадраты, овалы вырежи

из бумаги. Снимай на светлом ровном фоне.

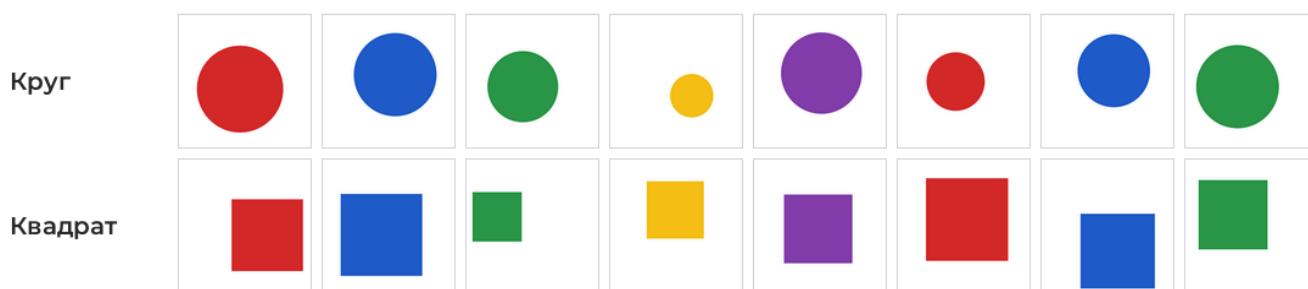


Рисунок В.36 – Образец выборки: по восемь картинок из папок «круг» и «квадрат»

Шаг 1. Открыть сайт. Зайди на teachablemachine.withgoogle.com и нажми «Начать» (1).



Рисунок В.37 – Главная страница Teachable Machine: кнопка «Начать»

Шаг 2. Выбрать проект. На странице «Новый проект» нажми «Проект с изображениями» (1 на левой половине рисунка). В окне выбери «Стандартную модель изображения» (1 на правой половине): она работает с цветными картинками и нужна нам. «Встроенная модель» – для микроконтроллеров, её не трогай.

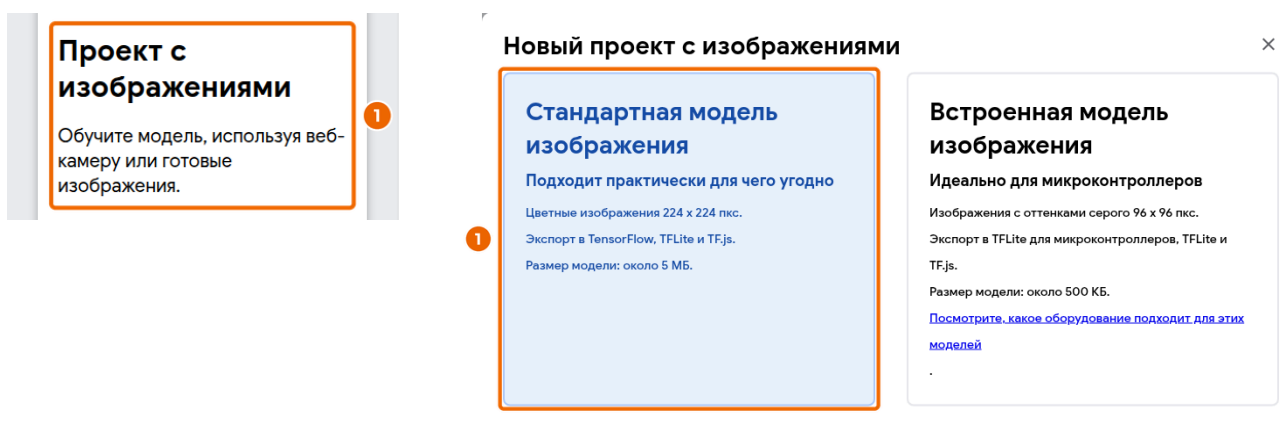


Рисунок В.38 – Новый проект: слева – карточка «Проект с изображениями», справа – выбор модели

Шаг 3. Окно редактора. Что где лежит, видно на рисунке. Сверху слева может всплыть синяя подсказка с видео – закрой её крестиком.

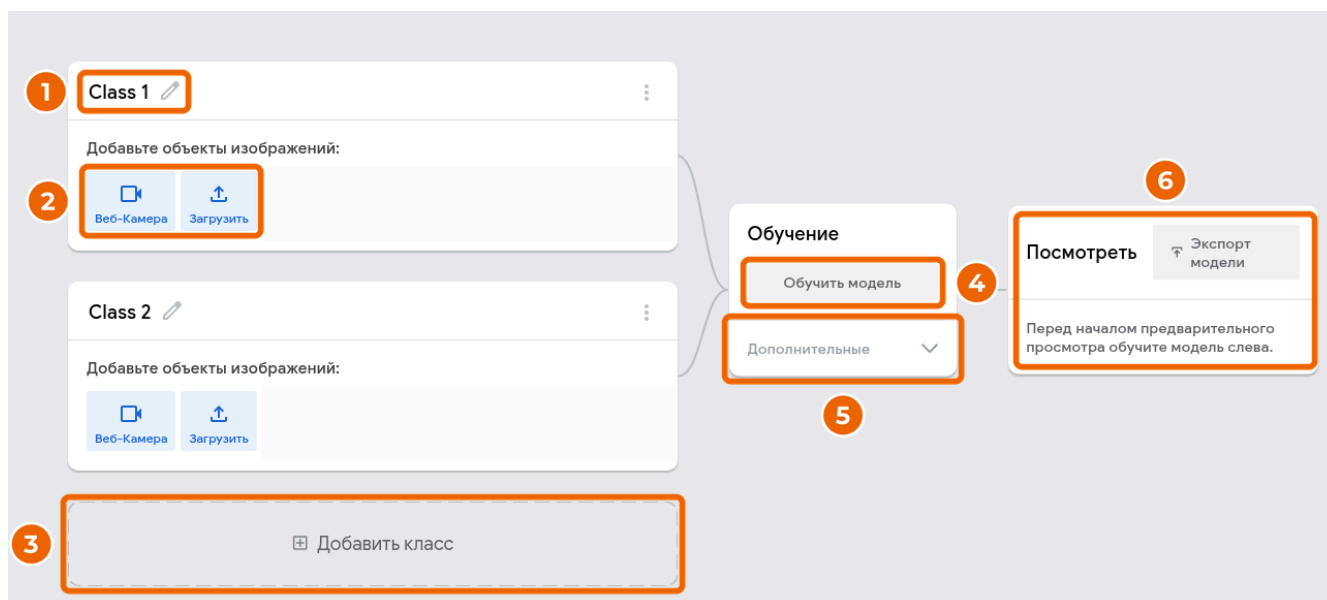


Рисунок В.39 – Редактор Teachable Machine

1. **Название класса.** Класс – это коробка для картинок одного вида. Название меняется карандашом.
2. **«Веб-камера» и «Загрузить»** – два способа положить в класс примеры. Нам нужен «Загрузить».
3. **«Добавить класс»** – ещё одна коробка. Сразу есть два класса, «Class 1» и «Class 2».
4. **«Обучить модель»** – запуск обучения.
5. **«Дополнительные»** – настройки обучения (эпохи, размер пакета, скорость обучения; они знакомы тебе из главы 7) и кнопка «Дополнительные сведения» с графиками.
6. **«Посмотреть»** – сюда после обучения показывают новые картинки и смотрят, что ответит модель.

Шаг 4. Назвать классы. Нажми на карандаш рядом с названием класса, сотри «Class 1», впиши «Круг» и нажми Enter. Второй класс назови «Квадрат»: так выглядит поле для названия, пока ты печатаешь (1 на левой половине рисунка ниже). Названия на панели «Посмотреть» обрезаются многоточием («Квад...») – это только подпись, на модель она не влияет.

Шаг 5. Загрузить примеры. В классе «Круг» нажми «Загрузить». Слева откроется панель: нажми «Выберите изображения из файлов» (1 на правой половине рисунка) и в окне выбора файлов найди папку «круг». Нажми Ctrl+A, чтобы выделить все 30 картинок, потом «Открыть». Сервис предупреждает (2 на правой половине), что обрежет картинки до квадрата: наши и так квадратные.

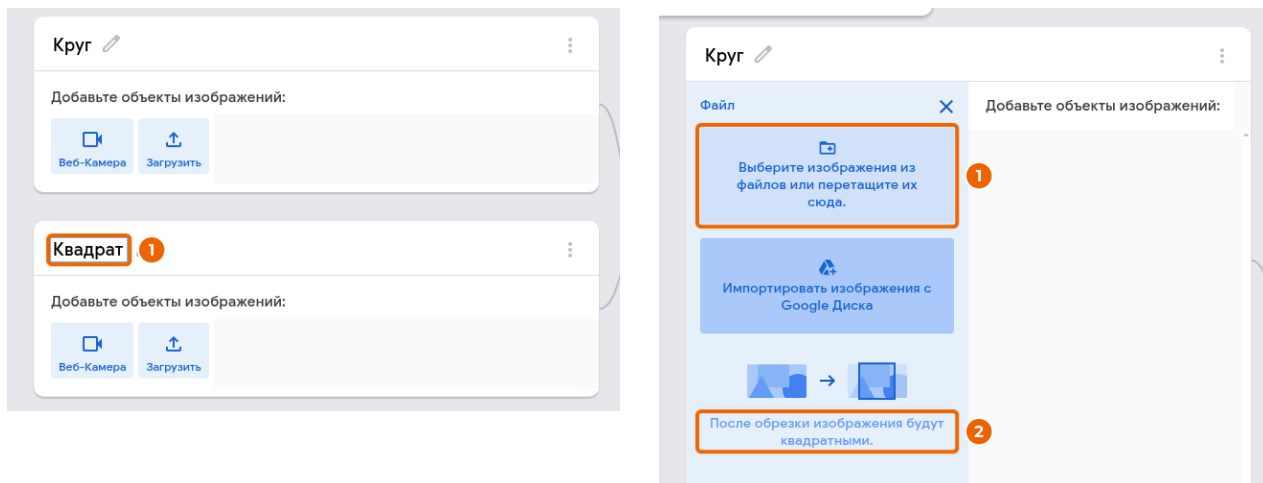


Рисунок В.40 – Слева – переименование класса, справа – панель «Загрузить»

Закрой панель крестиком, то же самое сделай со вторым классом и папкой «квадрат». Проверь, что в каждом классе написано «Образцов изображений: 30» (1 и 2). Если число не такое, значит, в классе лишние или пропавшие картинки – добавь или найди их.

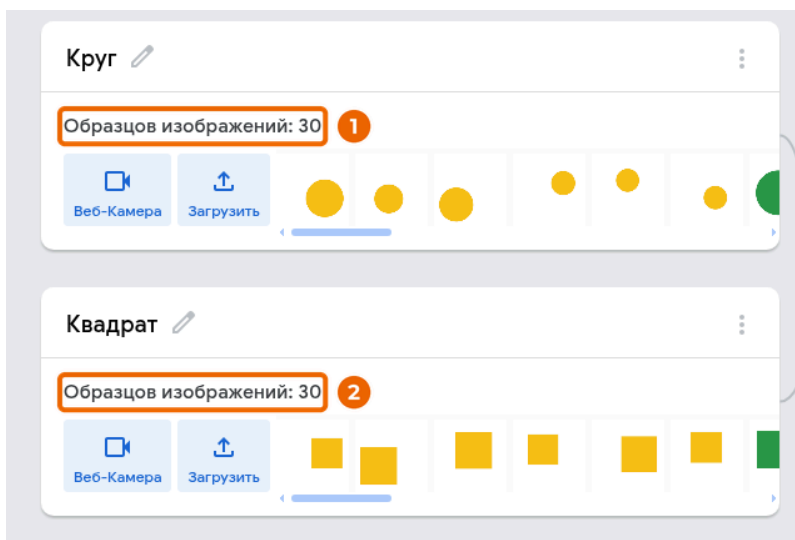


Рисунок В.41 – Оба класса готовы: по 30 образцов

Шаг 6. Обучить. Нажми «Обучить модель» (на рисунке шага 3 – кнопка 4). Кнопка превратится в полоску «Обучение», а под ней побежит счётчик (1): время и номер эпохи из 50. Сверху всплывёт сообщение (2): не переключайся на другие вкладки и не закрывай эту, пока идёт обучение. Оно занимает от нескольких секунд до минуты. Когда на кнопке появится «Модель обучена», модель готова.

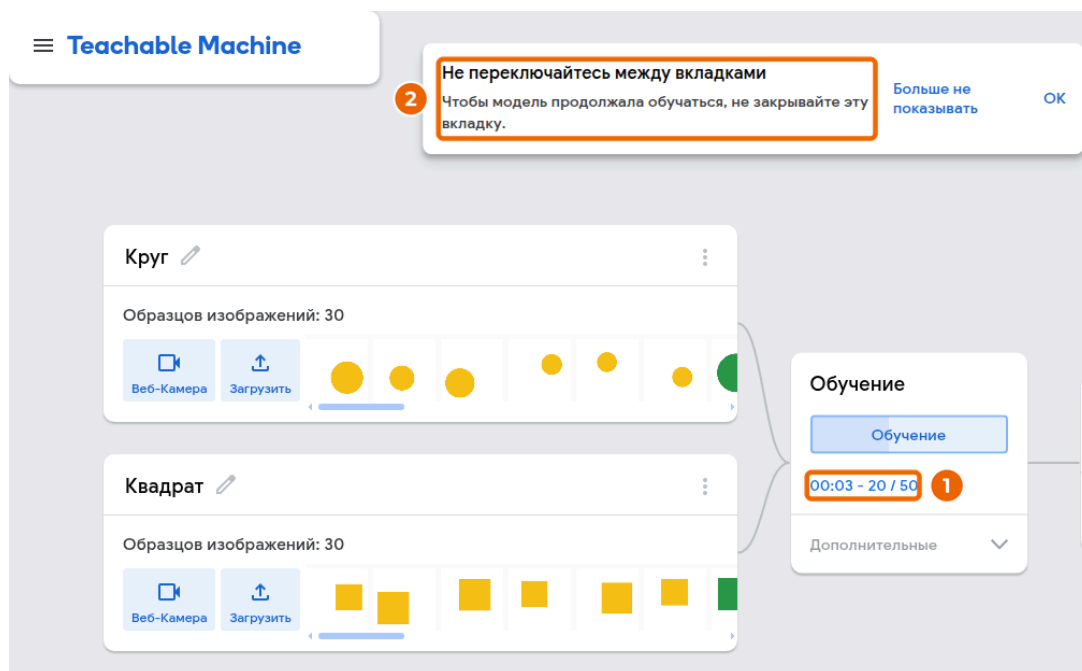


Рисунок В.42 – Обучение идёт: 20-я эпоха из 50

Шаг 7. Проверить на новой картинке. По умолчанию панель «Посмотреть» ждёт веб-камеру: браузер попросит к ней доступ или сервис напишет «Не удалось включить веб-камеру». Нам камера не нужна. Открой выпадающий список справа от слова «Ввод» (1) и выбери «Файл». Нажми «Выберите изображения из файлов» (2) и возьми любую картинку из папки «проверка». Под ней в «Выходных данных» (3) появятся полоски: для каждого класса – процент уверенности. Проценты всех классов в сумме дают 100, а ответ модели – класс с самой длинной полоской.

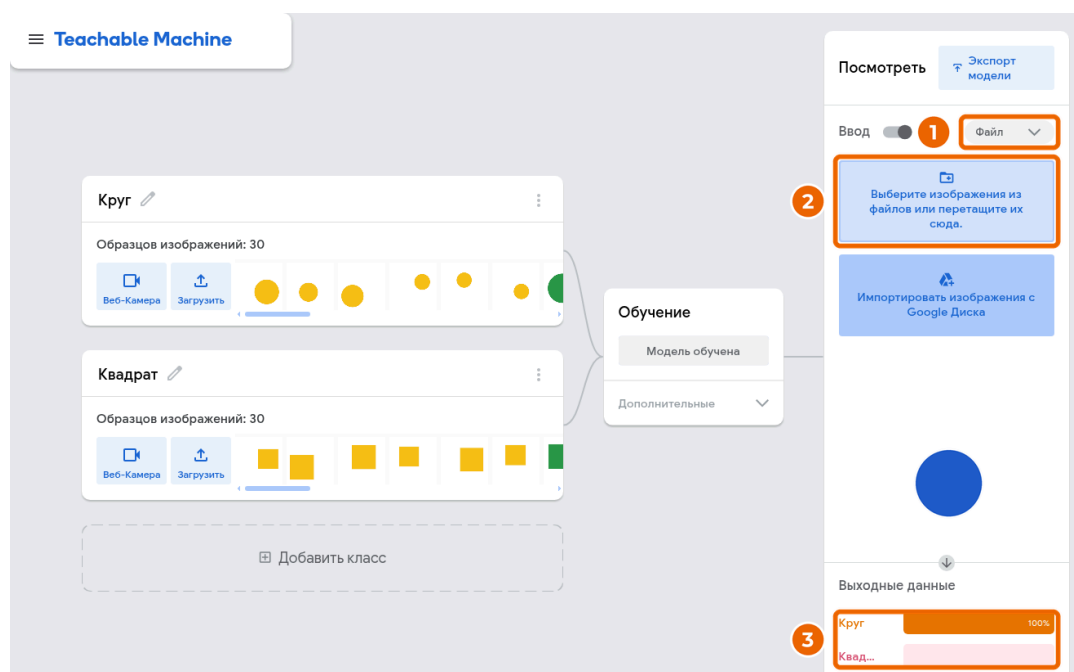


Рисунок В.43 – Проверка на файле: новый синий круг, ответ «Круг» со 100 %

Не потерять работу. Закроешь вкладку – проект пропадёт. Чтобы сохранить, нажми три полоски слева вверху и выбери «Скачать проект как файл»: на компьютер придёт файл `project.tm`. Открывается он на странице выбора проекта кнопкой «Открыть существующий

проект из файла». Входить в аккаунт Google для этого не нужно. Пункты про Google Диск пропусти.

Практика: четыре опыта.

Опыты идут по порядку: первые три – в одном проекте, для четвёртого начни новый. Числа в твоём проекте будут немного другими: при каждом обучении сеть стартует с других случайных весов. Важно, куда числа сдвигаются, а не какими они получились у нас.

Задача П.1. Круг или квадрат? Сделай по шагам 1–7 модель из двух классов «Круг» и «Квадрат» по 30 картинок. Проверь четыре картинки из папки «проверка»: два новых круга и два новых квадрата разных цветов. Для каждой запиши, какой класс выбрала модель и сколько процентов получил этот класс.

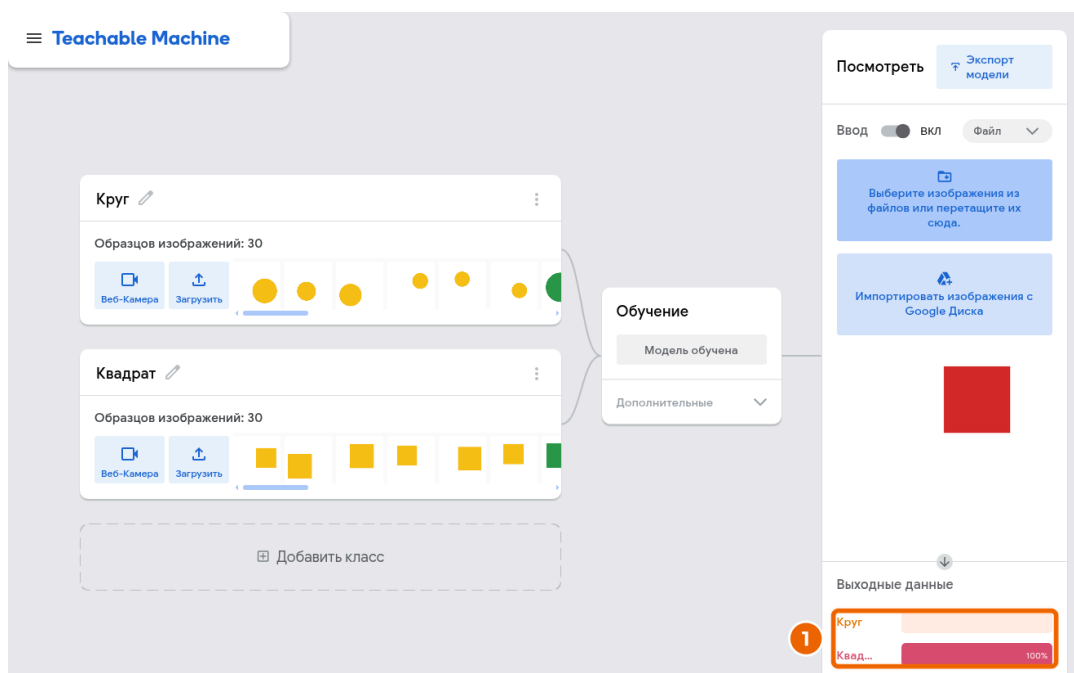


Рисунок В.44 – Новый красный квадрат: ответ «Квадрат» со 100 %

Что должно получиться: на всех четырёх картинках правильный класс набрал больше 90 %. У нас получилось 100 % и 0 %: фигуры такие разные, что модель не сомневалась. Связь с книгой: 60 картинок, которые ты положил в классы, – обучающая выборка, а картинки из «проверки» – проверка на новых данных: их модель до этого не видела (глава 1 и раздел 7.5).

Задача П.2. Заглянуть под капот. Не закрывая проект, раскрой в блоке «Обучение» список «Дополнительные» и нажми «Дополнительные сведения» (1). Справа выедет панель. Нажми «Вычислить точность на каждый класс» (2) и «Вычислить матрицу ошибок» (3). Потом прокрути панель вниз к графикам «Точность на каждую эпоху» и «Потеря на каждую эпоху». Ответь: сколько картинок каждого класса сервис отложил для проверки? Какая получилась точность по классам? Что стоит в клетках матрицы вне главной диагонали?

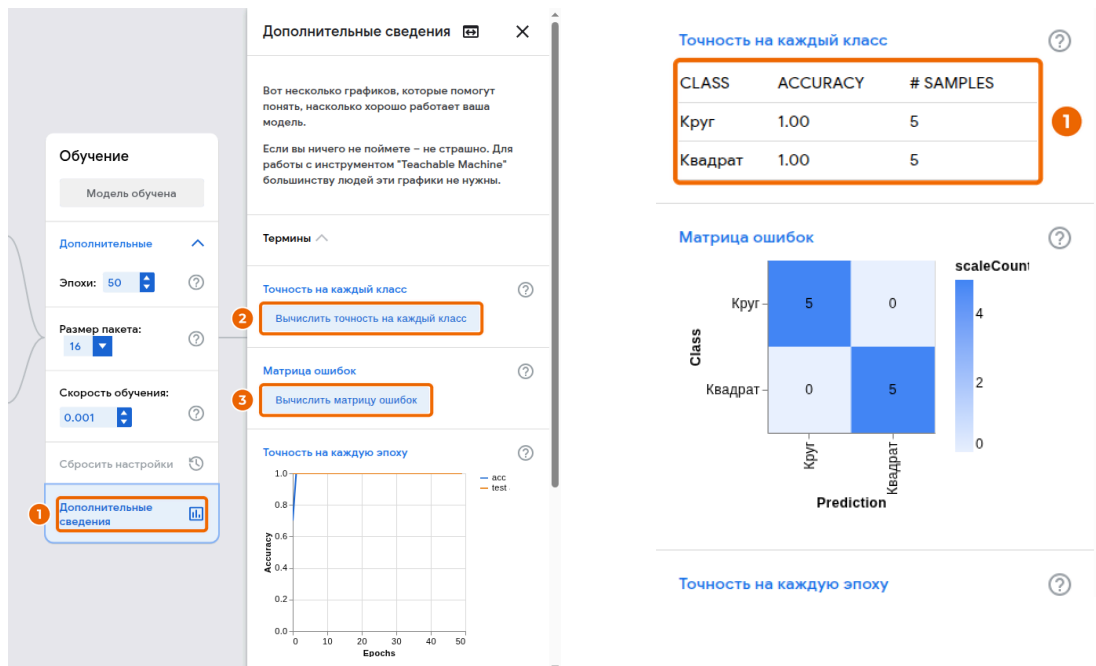


Рисунок В.45 – Слева – кнопки панели «Дополнительные сведения», справа – таблица точности и матрица ошибок хорошей модели

Что должно получиться:

- Сервис сам отложил 15 % картинок и не учил на них модель: по 5 из каждой 30 (столбец «# Samples»). В Teachable Machine они называются тестовыми, а в книге – проверочными.
- Точность у обоих классов 1.00: из 5 отложенных картинок верно названы все 5.
- В матрице ошибок строка – настоящий класс, столбец – ответ модели, число в клетке – сколько картинок. У нас по 5 на главной диагонали и нули вне её: модель не ошиблась ни разу.
- На графике точности две линии: «acc» – точность на обучающих картинках, «test» – на отложенных. У хорошей модели обе почти сразу упираются в 1.0.
- График потерь показывает ошибку (функцию потерь): она падает почти до нуля за несколько эпох.

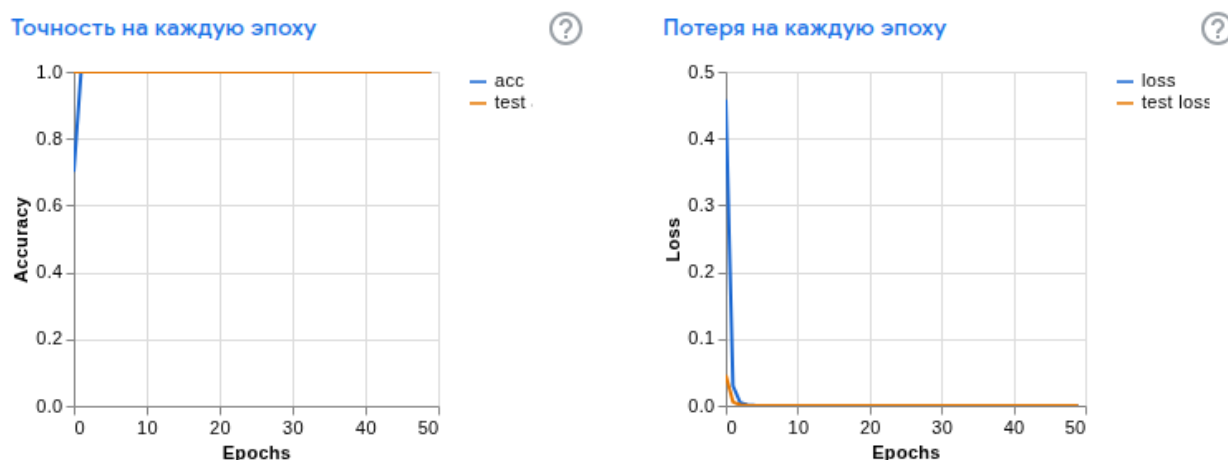


Рисунок В.46 – Хорошая модель: слева точность по эпохам («acc» – обучающие, «test» – отложенные), справа потеря

Задача П.3. Перепутанные подписи. В классе «Круг» нажми «Загрузить» и добавь ещё 10 картинок, но квадратов: любые десять из папки «квадрат». Теперь в «Круге» 40 образцов, и 10 из них подписаны неправильно – будто в кружке кто-то перепутал папки. Нажми «Обучить модель» ещё раз: после любого изменения в классах кнопка снова становится активной. Когда модель обучится, в списке «Ввод» может опять появиться сообщение о камере: переключи список на «Веб-камера» и обратно на «Файл». Потом: а) проверь на новых картинках те же четыре; б) открой «Дополнительные сведения», вычисли точность по классам и матрицу ошибок; в) посмотри оба графика и сравни их с графиками из П.2.

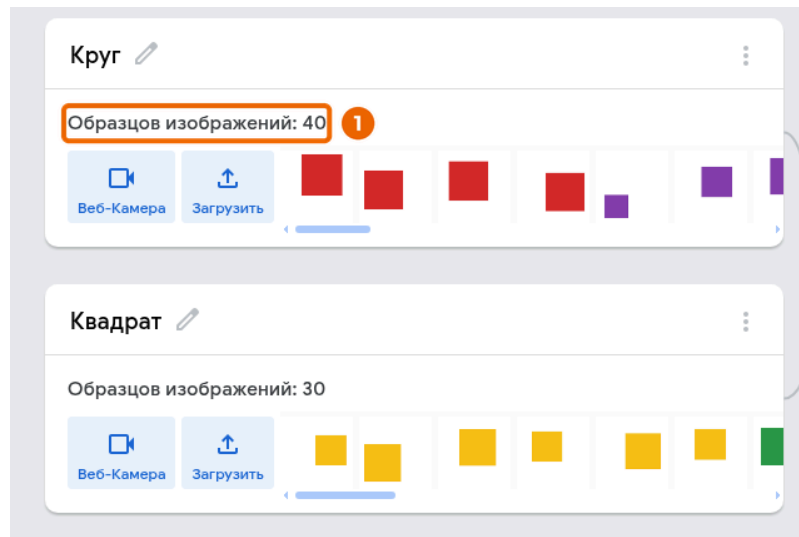


Рисунок В.47 – В классе «Круг» теперь 40 образцов, среди них красные и фиолетовые квадраты

Что должно получиться:

- На четырёх новых картинках модель ещё часто права: у нас круги получили 100 %, квадраты от 81 до 100 %. По одной-двум картинкам порчи почти не видно.
- В таблице точность по классам упала: у нас 0.50 для круга и 0.60 для квадрата, в других запусках выходило от 0.20 до 0.80. Одинаковых чисел не жди, но 1.00 уже не будет. В матрице вне главной диагонали больше нет одних нулей.
- На графике точности линии разошлись: «асс» держится около 0.9, «test» падает ниже, у нас примерно до 0.55.
- На графике потерь «loss» всё ещё снижается, а «test loss» растёт.

Модель подгоняется под обучающие картинки вместе с их ошибками и всё хуже справляется с отложенными. Это то самое переобучение из раздела 7.4. Значит, ошибки в подписях не видны на одной-двух проверочных картинках: их находят по точности по классам и по расхождению графиков.



Рисунок В.48 – Испорченная модель: слева таблица и матрица ошибок, справа график точности – «test» ушла вниз от «acc»

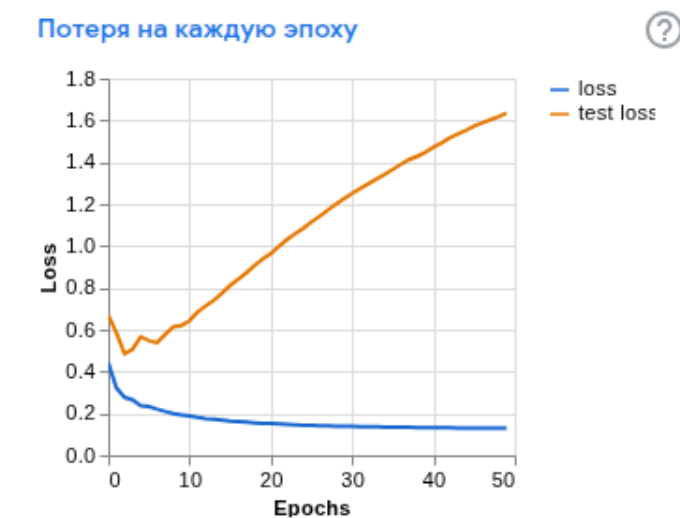


Рисунок В.49 – Потеря на каждую эпоху: «test loss» растёт, пока «loss» падает

Задача П.4. Мало примеров. Открой сайт заново в новой вкладке и начни новый проект (шаги 1 и 2). Назови классы «Круг» и «Овал». В «Круг» загрузи всего 3 любые картинки, в «Овал» – все 30. Обучи модель. Проверь пять новых кругов и один новый квадрат из папки «проверка», запиши ответы. Потом загрузи в «Круг» остальные 27 картинок, обучи модель ещё раз (список «Ввод» переключи так же, как в П.3) и проверь те же пять кругов. Сколько новых кругов модель назвала овалами до и после добора? Что она ответила про квадрат?

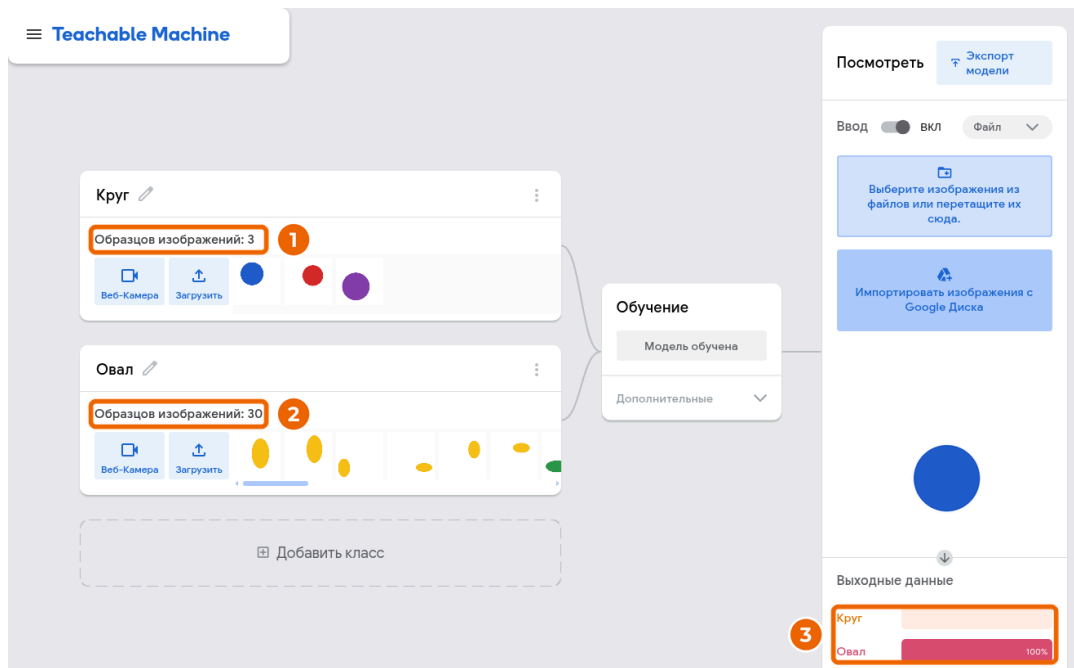


Рисунок В.50 – В «Круге» 3 образца, в «Овале» 30: новый синий круг назван «Овалом» на 100 %

Что должно получиться:

- Пока кругов три, модель называет «Овалом» почти всё: из пяти новых кругов у нас «Кругом» назван один, остальные четыре – «Овал» с уверенностью 95–100 %.
- В таблице точности у круга всего одна отложенная картинка, и по ней ничего не скажешь: у нас вышло 0.00, в другой раз было 1.00. Проверка на новых картинках показала больше любой таблицы.
- Квадрат модель тоже назвала «Овалом» на 100 %, хотя таких картинок никогда не видела. «Не знаю» она сказать не умеет и выбирает из тех классов, что есть. Уверенность – это не правильность.
- После добора до 30 кругов все пять новых кругов стали «Кругом» (97–100 %), овалы остались овалами.

Так работает перекося классы из раздела 7.5: три картинки против тридцати – это даже не один порядок.

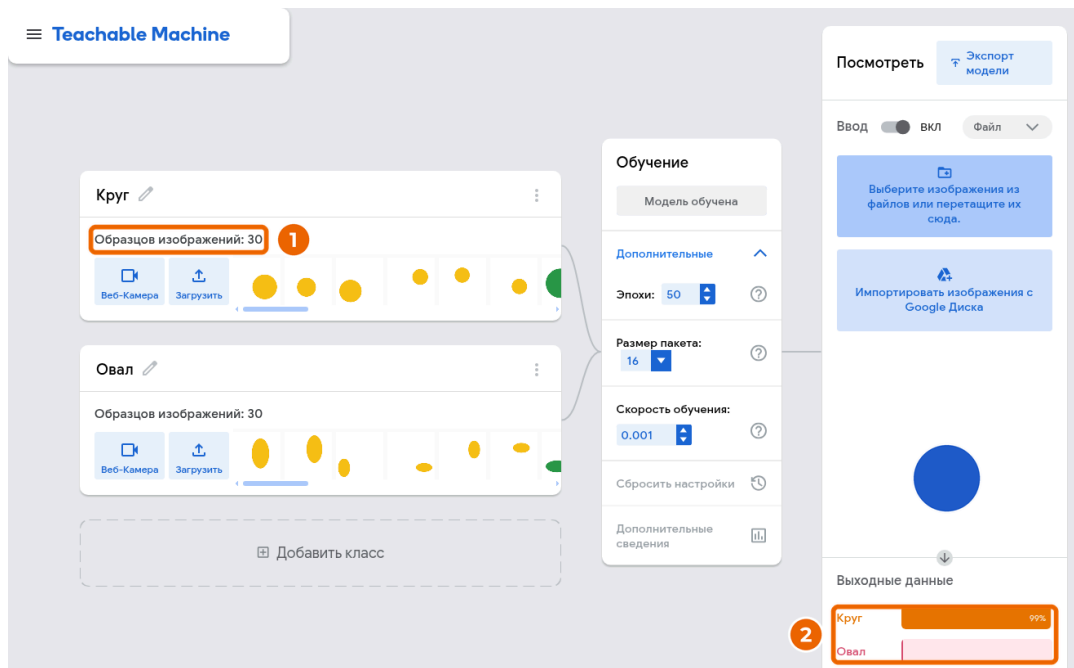


Рисунок В.51 – После добора: в «Круге» 30 образцов, новый синий круг назван «Кругом» на 99 %

Если взять вместо овалов квадраты, перекося почти незаметно: у нас три круга и тридцать квадратов модель различила без единой ошибки, фигуры слишком разные. Перекося ломает модель там, где классы похожи: круг и овал, синица и снегирь у кормушки.

План на месяц до отбора

Если начинаешь в конце сентября, вот спокойный план: примерно по часу в день.

Таблица В.3

Неделя	Что читаем	Что делаем
1	главы 1–3	каждый день 3–4 разбора с задачами «Реши сам»; зарегистрироваться на курс сферы
2	главы 4–5	открыть онлайн-Python или Colab, повторить код из разборов главы 5
3	главы 6–7	поставить GraphNet и Датавиз-конструктор по инструкциям курса, построить графики из разборов
4	глава 8 и шпаргалка	заново решить задачи, где были ошибки; пройти тест курса сферы (+5 баллов)

Отбор длится до 10 ноября, и попыток две. Если первая попытка прошла неудачно, вернись к главам, где потерял баллы, и иди на вторую.

Родителям и наставникам

- **Помогайте разбираться, но не решайте за ребёнка.** И не давайте решать нейросети: на командном финале сразу видно, кто разобрался сам, а за кого решили. Лучше пусть ребёнок ошибётся на отборе и научится, чем попадёт в финал и растеряется там.
- **Садитесь рядом на первых задачах.** Детей чаще всего останавливает не сложность, а «программа не ставится», «что-то зависло» или «непонятно, куда нажать».
- **ИИ-чаты – вместе со взрослым.** Прочитайте правила сервиса: GigaChat, Алиса и DeepSeek рассчитаны на взрослых, а детям до 18 лет разрешают пользоваться ими с согласия родителей и под их присмотром. Договоритесь, что в чат не пишут имя, адрес, школу и фотографии. И проверяйте ответы вместе: нейросеть умеет уверенно ошибаться.
- **Всё про регистрацию, бонусы и финал** для Новосибирской области – на nto.kvantorium54.ru (вкладки «Я наставник» и «Я участник или родитель»).
- **Региональный координатор НТО Юниоры в Новосибирской области** – детский технопарк «Кванториум», info@kvantorium54.ru.

Честно об этой книге

Главное, что нужно знать про эту книгу: заданий отбора 2026/27 мы не видели – организаторы их заранее не показывают. Угадывать их бесполезно: наставники, которые следят за олимпиадой не первый год, говорят, что задания из года в год не повторяются. Поэтому книга учит темам из программы сферы и приёмам решения, а не готовит к конкретным задачам. Когда организаторы выложат пробные задания, мы сверим с ними книгу и выпустим новую версию.

Это не официальное издание НТО. Книгу составил методист «Кванториума» по открытым материалам олимпиады: программе сферы «Основы искусственного интеллекта и анализа данных» 2026/27, записям вебинаров сферы и разборам задач отборочных этапов прошлых лет (сфера тогда называлась «Технологии и виртуальные помощники», а потом «Технологии и искусственный интеллект»). Задачи прошлых лет пересказаны своими словами, с новыми числами и новыми ответами.

Разработчики предупредили, что GraphNet и Датавиз-конструктор к этому сезону обновились. Поэтому кнопки и окна этих программ мы описываем в общих чертах, а смысл – подробно: смысл от версии к версии не меняется. Точную инструкцию к новой версии смотри в курсе сферы.

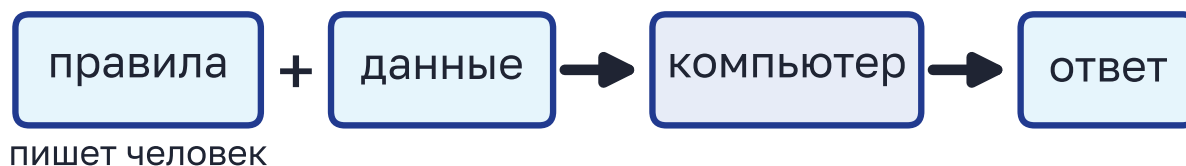
Каждое число в ответах мы пересчитали на компьютере, а каждую программу запустили. Если ты всё-таки нашёл ошибку или что-то осталось непонятным, напиши автору: osinovsky.aa@kvantorium54.ru. Следующие версии книги станут лучше благодаря тебе.

Что такое искусственный интеллект



С этой главы начинается вся сфера. Формул здесь почти нет, зато есть слова, без которых дальше не прочитать ни одного условия: «признак», «обучающая выборка», «классификация», «вес нейрона», «точность». На отборе их проверяют так: сопоставь термины, определи тип задачи по описанию, посчитай точность модели по таблице ответов или выход нейрона по его весам. Задачи несложные, а баллы на них теряют из-за одного – путают похожие слова. Поэтому у каждого слова в главе есть свой пример.

Обычная программа



Машинное обучение

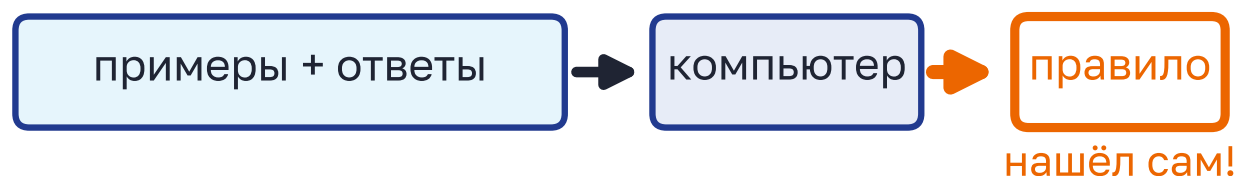


Рисунок 1.1 – Обычная программа и машинное обучение

В обычной программе правило придумал человек. В машинном обучении компьютер находит правило сам – по примерам с ответами.

Запомни. ИИ – программы для задач, где человеку нужен интеллект. Машинное обучение – та часть ИИ, которая учится на примерах, а нейросети – часть машинного обучения. Если программа только выполняет правила, записанные человеком (звонок, калькулятор, светофор по таймеру), – это не ИИ.

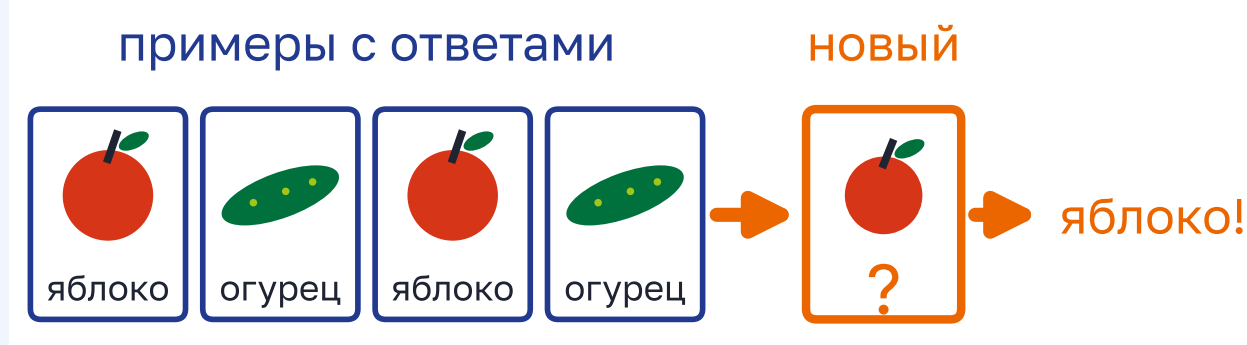


Рисунок 1.2 – Примеры с ответами и новый предмет

Никто не объяснял правило словами. Посмотрели на примеры с ответами – и новый предмет узнаётся сам.

Запомни. Строка таблицы – объект. Столбцы, по которым угадываем, – признаки, а столбец, который надо угадать, – ответ (целевая переменная). Номера и фамилии, которые лишь называют строку, – не признаки: перед обучением их удаляют вместе с ответом.

НА ПАЛЬЦАХ

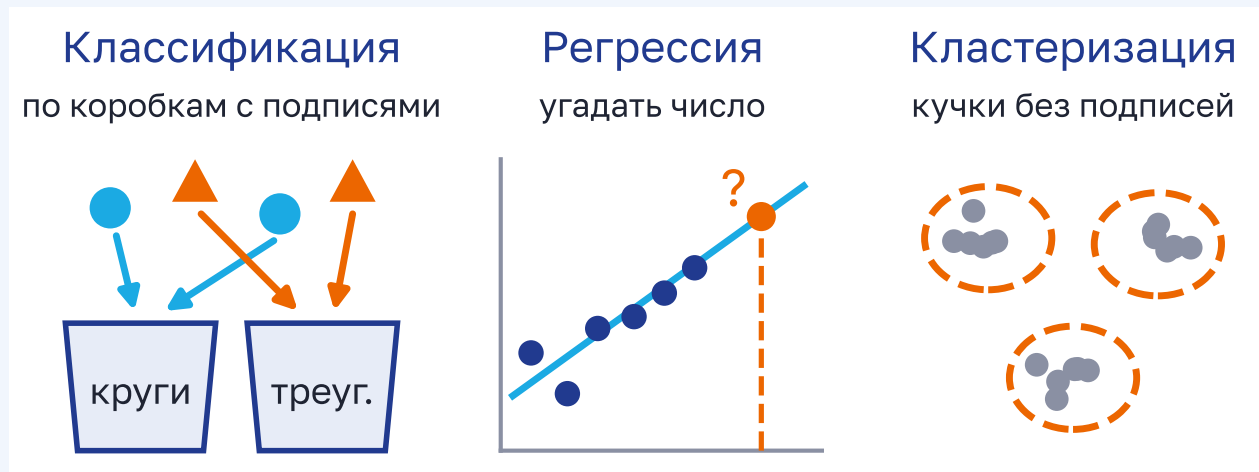


Рисунок 1.3 – Три типа задач машинного обучения

Классификация раскладывает по коробкам, на которых уже есть подписи. Регрессия угадывает число. Кластеризация сама находит кучки, и подписей у них нет.

Запомни. Готовые группы и ответы в примерах – классификация. Число – регрессия. Ответов нет, ищем группы сами – кластеризация. Первым делом смотри, **что должно получиться на выходе**.

Запомни. Обучающая выборка – учимся, проверочная – сдаём экзамен. Проверочные примеры модель при обучении не видит. Точность, которой можно верить, – только на проверочной выборке.

Запомни. Нейрон: умножь каждый вход на его вес, сложи, сравни сумму с порогом. $S \geq T$ – выход 1, иначе 0. Отрицательный вес работает «против». Обучение – это подбор весов.

Запомни. Точность = верных : всего · 100 %. Доля ошибок = 100 % – точность. Считаем на проверочной выборке. Для регрессии – средняя ошибка: насколько в среднем промахнулись, без учёта знака.

Разбор для старта

Задача 1.1 Какой это тип задачи

разминка · типы задач машинного обучения

Сопоставь каждую задачу с её типом: 1 – классификация, 2 – регрессия, 3 – кластеризация.

- А. По скорости и размеру астероида отнести его к одному из трёх классов опасности: 1, 2 или 3 (в обучающих примерах класс указан).
- Б. Предсказать, сколько сантиметров снега выпадет завтра в Новосибирске.
- В. Разбить 5000 снимков звёздного неба на группы похожих; какие будут группы, заранее неизвестно.
- Г. По письму определить, спам это или нет.
- Д. По температуре и заряду аккумулятора лунохода предсказать, сколько минут он проработает до разрядки.
- Е. Разделить покупателей магазина на группы по их покупкам, ничего не зная о покупателях заранее.

Ответ запиши как шесть цифр подряд – по порядку для А, Б, В, Г, Д, Е.

Дано: шесть описаний задач.

Найти: тип каждой.

Решение

Шаг 1. Правило. Для каждой задачи задаём два вопроса из раздела 1.3. Есть ли в примерах правильные ответы? Если есть – что получается на выходе: число или одна из готовых групп?

Шаг 2. А. Ответ – класс 1, 2 или 3, список классов известен, в примерах он указан. Цифры здесь – имена классов, а не величины: класса 2,5 не бывает. Классификация – **1**.

Шаг 3. Б. Ответ – число сантиметров, может быть и 12,5. Регрессия – **2**.

Шаг 4. В. Правильных ответов нет, группы надо найти самим. Кластеризация – **3**.

Шаг 5. Г. Два готовых класса: «спам» и «не спам». Классификация – **1**.

Шаг 6. Д. Ответ – число минут. Регрессия – **2**.

Шаг 7. Е. Ответов нет, группы неизвестны. Кластеризация – **3**.

Перепроверь кластеризацию. Только в В и Е прямо сказано, что групп заранее никто не знает. В остальных четырёх ответы в примерах есть, значит, там классификация или регрессия.

 **Ошибка номер один** – записать задачу А в регрессию, потому что ответ – цифра. Цифра – ещё не число-величина: номер класса, номер автобуса, номер отсека – это имена. А задачу В часто принимают за классификацию из-за слова «группы». Группы в классификации известны заранее и подписаны, а здесь их надо найти.

Ответ: 123123.

Своя задача по программе сферы, модуль 1

Проверь себя

1.1. Определи тип задачи (1 – классификация, 2 – регрессия, 3 – кластеризация): а) по росту и весу собаки предсказать её возраст в годах; б) по записи голоса птицы определить, кто поёт: синица, воробей или снегирь (в обучающих записях это подписано); в) разложить 300 снимков комет на группы похожих, не зная заранее, какие будут группы; г) предсказать, сколько человек придёт на новогоднюю ёлку в Академгородке. Ответ – четыре цифры подряд.

1.2. Какие из этих устройств и программ используют машинное обучение (учились на примерах)? 1) калькулятор в телефоне; 2) разблокировка телефона по лицу; 3) светофор, который переключается каждые 40 секунд; 4) голосовой помощник, который распознаёт твою речь; 5) электрочайник, который выключается при закипании; 6) лента видео, которая подбирает ролики по тому, что ты смотрел. Запиши номера в порядке возрастания.

1.3. Наставник кружка хочет по таблице предсказать, пройдёт ли ученик в финал олимпиады. Столбцы таблицы: «Фамилия», «Номер в журнале», «Класс», «Часов занятий в неделю», «Решено задач на пробном туре», «Прошёл в финал». Какие столбцы нужно удалить, чтобы остались только признаки? Какой это тип задачи?

1.4. Сопоставь термины и определения. Термины: 1) обучающая выборка; 2) проверочная выборка; 3) признак; 4) целевая переменная; 5) идентификатор. Определения: А) столбец, значение которого модель должна угадать; Б) примеры, которые модели не показывают во время обучения, чтобы потом оценить её; В) столбец, по которому различают строки, но который ничего не говорит об ответе; Г) примеры с ответами, на которых модель учится; Д) свойство объекта, по которому модель угадывает ответ. Ответ запиши как пять букв по порядку для терминов 1–5.

Множества и логика



Когда ты выбираешь в интернет-магазине фильтры «синий» и «до 1000 рублей», сайт ищет пересечение двух множеств. Когда спам-фильтр решает, пустить письмо во «Входящие» или нет, он проверяет правило из связок И, ИЛИ, НЕ. В этой главе научимся считать элементы, когда группы пересекаются, рисовать круги Эйлера и читать логические правила. На отборе такие задачи обычно короткие: история про книги, кружки или поисковые запросы, а в ответ вводится одно число или выбирается вариант.

НА ПАЛЬЦАХ

Множество «игрушки в коробке»

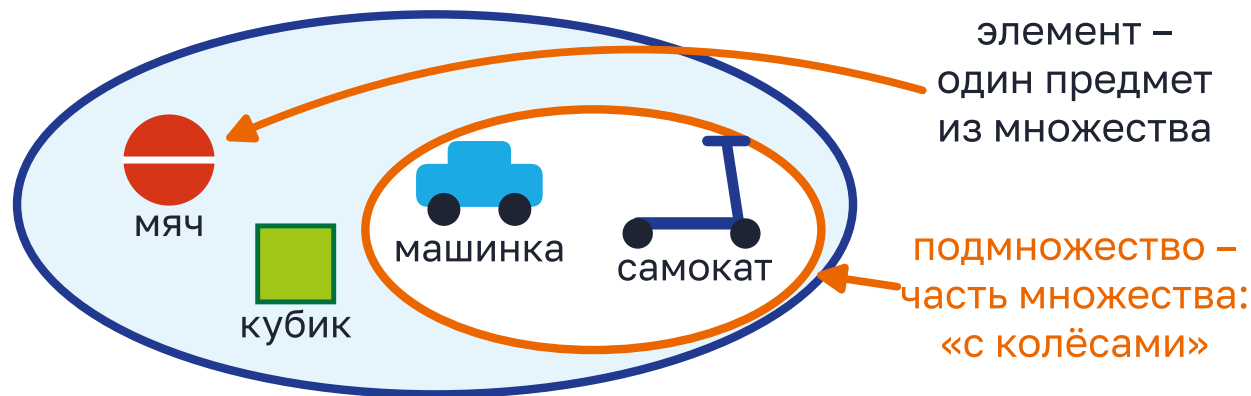


Рисунок 2.1 – Множество – как обруч, в который сложили предметы

Всё, что лежит внутри обруча, – элементы множества. Обруч поменьше внутри большого – подмножество.

Запомни. Множество – набор без повторов, порядок в нём не важен. $n(A)$ – сколько в нём элементов. Подмножество – часть, все элементы которой есть в большом множестве.

НА ПАЛЬЦАХ

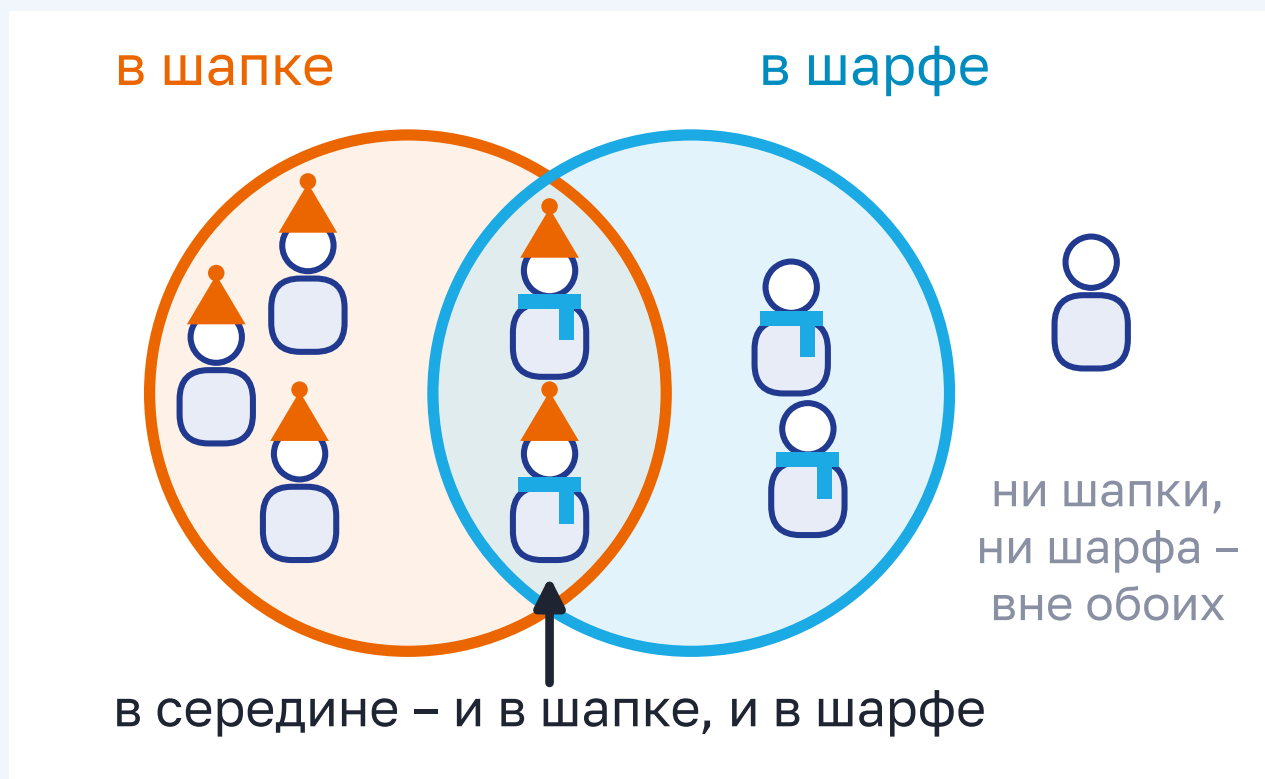


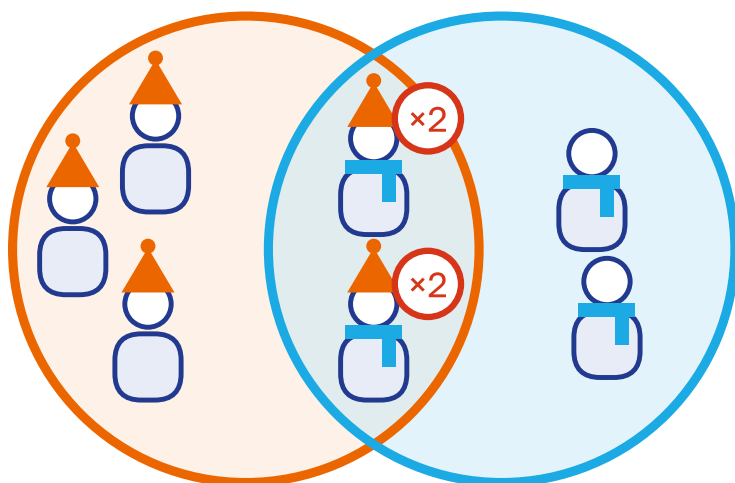
Рисунок 2.2 – Два обруча на полу: в левый встали те, кто в шапке, в правый – те, кто в шарфе

Тот, кто и в шапке, и в шарфе, должен стоять сразу в двух обручах. Поэтому обручи кладут внахлёст, а он встаёт в середину.

Запомни. $A \cap B$ – «и то, и другое», $A \cup B$ – «хотя бы одно», « A , но не B » – круг A без общей части, «не A » – всё за пределами круга A . Одно понятие может занимать и один кусочек картинки, и несколько.

в шапке: 5

в шарфе: 4



$$5 + 4 = 9?$$

но детей
в обручах – 7

двоих из середины
посчитали дважды:
 $9 - 2 = 7$

Рисунок 2.3 – Пятеро в шапке, четверо в шарфе. Но детей в обручах не девять, а семь

Двое из середины попали и в «шапки», и в «шарфы». Сложили – посчитали их дважды. Один лишний раз надо вычесть.

Запомни. Хотя бы в одном = $n(A) + n(B) - n(A \cap B)$. Ни в одном = все минус хотя бы в одном. Для трёх кругов заполняй кусочки от центра наружу, а в конце проверь сумму формулой.

Запомни. И – истина, только если истинны оба. ИЛИ – ложь, только если ложны оба. НЕ – наоборот. Порядок: скобки, НЕ, И, ИЛИ. Для k высказываний в таблице истинности 2^k строк: 4, 8, 16...

Запомни. & – И – пересечение: найдётся меньше. | – ИЛИ – объединение: найдётся больше. Пересечение по трём числам: $n(A \cap B) = n(A) + n(B) - n(A \cup B)$.

Разбор для старта

Задача 2.1 Кружки после уроков

разминка · объединение, формула включений и исключений

В 6 «В» 28 учеников. На робототехнику ходят 16 человек, на программирование – 11, на оба кружка – 5. а) Сколько учеников ходят хотя бы в один кружок? б) Сколько не ходят ни в один?

Дано: всего 28; $n(P) = 16$; $n(\Pi) = 11$; $n(P \cap \Pi) = 5$.

Найти: $n(P \cup \Pi)$ – ?; ни в один – ?

Решение

Шаг 1. Что происходит. Два множества: «ходят на робототехнику» и «ходят на программирование». Они пересекаются: 5 человек ходят туда и туда. Рисуем два круга внахлест внутри прямоугольника «весь класс».

Шаг 2. Хотя бы в один. По формуле включений и исключений: $16 + 11 - 5 = 22$ ученика.

Шаг 3. Ни в один. Из всего класса вычитаем тех, кто хотя бы в одном: $28 - 22 = 6$ учеников.

Проверка. Только робототехника: $16 - 5 = 11$, только программирование: $11 - 5 = 6$, оба: 5, ни одного: 6. Всего $11 + 6 + 5 + 6 = 28$ – как раз весь класс.

И прикинь на глаз: тех, кто ходит хотя бы в один кружок, не может быть меньше, чем в самом большом кружке (16), и больше, чем во всём классе (28). 22 – между ними.

⚠ Самая частая ошибка. Сложить $16 + 11 = 27$ и ответить «27 ходят хотя бы в один, 1 не ходит никуда». Пятерых, кто ходит в оба кружка, посчитали два раза.

Ответ: а) 22 ученика; б) 6 учеников.

Своя задача по программе сферы, модуль 2

Проверь себя

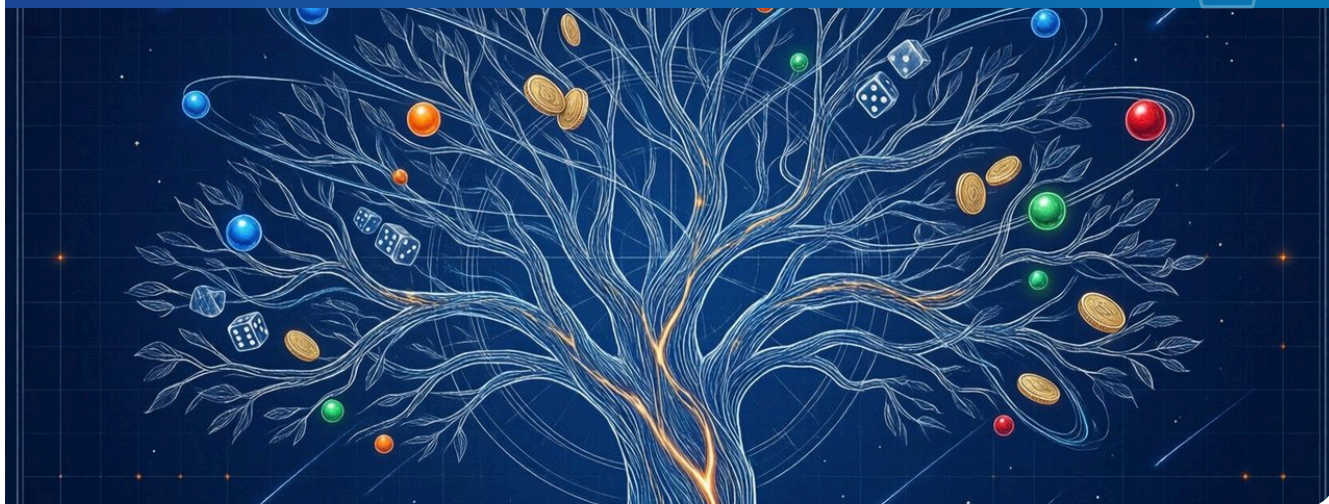
2.1. В классе 25 учеников. Математику любят 14, информатику – 12, а 4 ученика не любят ни то, ни другое. Сколько учеников любят оба предмета?

2.2.* В классе 30 учеников. 20 ходят на плавание, 17 – на футбол. Какое **наименьшее** и какое **наибольшее** число учеников может ходить и туда, и туда? Запиши два числа через точку с запятой: сначала наименьшее.

2.3. A – множество букв слова «МАТЕМАТИКА», B – множество букв слова «ИНФОРМАТИКА». Сколько элементов в $A \cup B$?

2.4. A – чётные числа от 1 до 20, B – числа от 1 до 20, которые делятся на 3. Найди $n(A \cap B)$ и $n(A \cup B)$.

Комбинаторика и вероятность



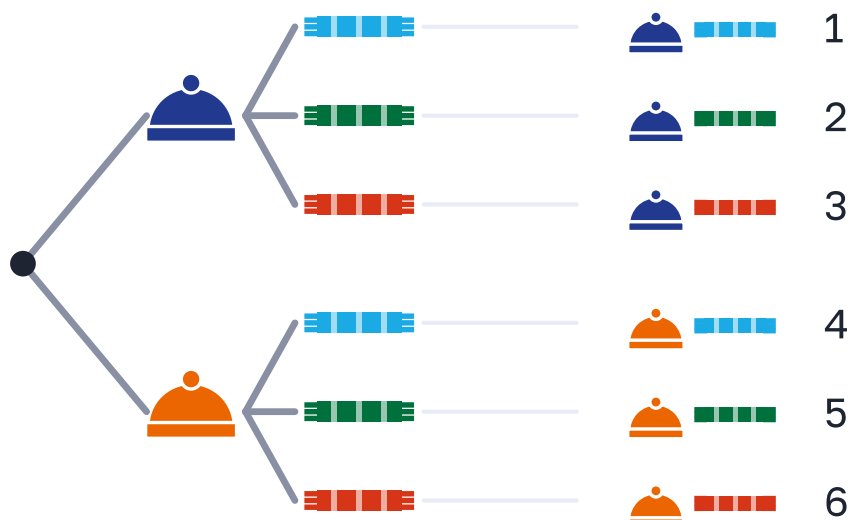
Сколько разных разговоров может получиться у чат-бота, если на каждом шаге человек выбирает одну из нескольких кнопок? Какова вероятность выбросить на двух кубиках 7? И что имеет в виду программа, когда пишет «кошка – 80 %»? Все три вопроса решаются одним приёмом: аккуратно пересчитать варианты и не потерять ни одного. На отборе прошлых лет была как раз такая задача – «дерево возможностей» чат-бота, ответ вводился одним числом. Арифметика здесь на уровне таблицы умножения. Баллы теряют на другом: пропускают ветку или считают одну и ту же дважды.

НА ПАЛЬЦАХ

шаг 1: шапка

шаг 2: шарф

что вышло



$$2 \cdot 3 = 6$$

нарядов

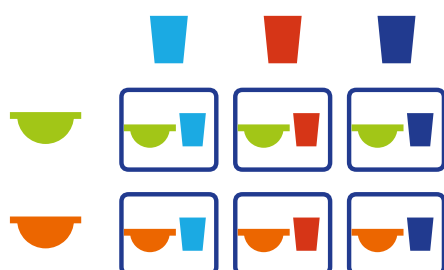
Рисунок 3.1 – Две шапки и три шарфа

Каждая шапка ветвится на три шарфа. Веток две, в каждой по три – всего $2 \cdot 3 = 6$ нарядов.

Запомни. Выбор по шагам – перемножаем, сколько вариантов на каждом шаге: $a \cdot b \cdot c$. Так можно, только если число вариантов на шаге не зависит от того, что выбрали раньше.

НА ПАЛЬЦАХ

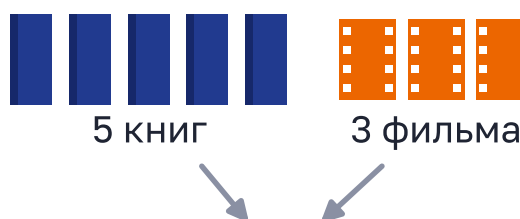
каша И напиток



$$2 \cdot 3 = 6$$

берём и то, и другое – умножаем

книга ИЛИ фильм



$$5 + 3 = 8$$

выбираем одно – складываем

Рисунок 3.2 – «И» – умножаем, «или» – складываем

Кашу и напиток берут вместе – каждая каша с каждым напитком, $2 \cdot 3 = 6$ пар. Книгу или фильм выбирают одно – все варианты встанут в одну очередь, $5 + 3 = 8$.

Запомни. «И» – берём по варианту на каждом шаге – умножаем. «Или» – берём одно из нескольких групп – складываем. Если у групп есть общие элементы, общее вычитаем один раз.

НА ПАЛЬЦАХ

$$\begin{array}{|c|} \hline 1\text{-е место} \\ \hline 3 \text{ варианта} \\ \hline \end{array} \cdot \begin{array}{|c|} \hline 2\text{-е место} \\ \hline 2 \text{ варианта} \\ \hline \end{array} \cdot \begin{array}{|c|} \hline 3\text{-е место} \\ \hline 1 \text{ вариант} \\ \hline \end{array} = 6$$

все очереди:



Рисунок 3.3 – Три робота встают в очередь на зарядку

На первое место – любой из трёх, на второе – любой из двух оставшихся, на третье – последний. $3 \cdot 2 \cdot 1 = 6$ очередей.

Запомни. Расставляем по местам: на каждое следующее место вариантов на один меньше. Все n предметов в ряд – $n! = n \cdot (n - 1) \cdot \dots \cdot 1$. Пары, где порядок не важен, – $\frac{n \cdot (n - 1)}{2}$.

Запомни. $P = \frac{\text{подходящие}}{\text{все}}$, и все исходы должны быть равновозможными. $0 \leq P \leq 1$. «Не случится» – $1 - P$. У двух монет 4 исхода, у двух кубиков – 36: каждую монету и каждый кубик считай отдельно.

Запомни. Вероятность – сколько должно выходить в среднем, частота – сколько вышло в нашем опыте. На малом числе опытов они могут сильно различаться, на большом – сближаются. Ожидаемое число случаев = $P \cdot N$.

Запомни. Проценты модели – её оценка, а не факт про один снимок. Честность модели проверяют на многих ответах: из тех, где она «уверена на 80 %», верными должны оказаться около 80 из 100. Вероятность фразы из нескольких слов – произведение вероятностей вдоль её ветки дерева.

Разбор для старта

Задача 3.1 Сценарии бота-звездочёта

разминка · дерево возможностей, правило умножения

Для планетария делают чат-бота «Звездочёт». Сначала он предлагает тему: Луна, Марс, Солнце, кометы или МКС. Потом спрашивает, как рассказывать: текстом, картинками или голосом. В конце загадывает по теме загадку – лёгкую или трудную, на выбор посетителя. Сценарий разговора – вся цепочка выборов. Сколько разных сценариев может быть у «Звездочёта»? В ответ введи целое число.

Дано: тем – 5; способов рассказа – 3; уровней загадки – 2.

Найти: N – число сценариев.

Решение

Шаг 1. Что происходит. Разговор идёт в три шага, на каждом посетитель выбирает один вариант. Сценарий = тема + способ рассказа + уровень загадки. Это выбор «и»: по варианту на каждом шаге.

Шаг 2. Можно ли умножать. Число способов рассказа не зависит от темы: и про Луну, и про кометы бот рассказывает тремя способами. Уровней загадки после любой темы – два. Значит, правило умножения работает.

Шаг 3. Считаем. В дереве из начала выходят 5 веток-тем, каждая делится на 3 ветки-способа: $5 \cdot 3 = 15$ пар «тема + способ». Каждая пара делится ещё на 2 уровня загадки: $15 \cdot 2 = 30$.

Проверим с другого конца. Выпишем начало: «Луна – текст – лёгкая», «Луна – текст – трудная», «Луна – картинки – лёгкая»... На одну только Луну приходится $3 \cdot 2 = 6$ сценариев, а тем пять: $6 \cdot 5 = 30$. Сходится.

 **Типичная ошибка.** Сложить: $5 + 3 + 2 = 10$. Это число всех кнопок, которые бот может показать, а не число разговоров. Складывают, когда выбирают что-то одно из всех вариантов, а здесь выбирают на каждом шаге.

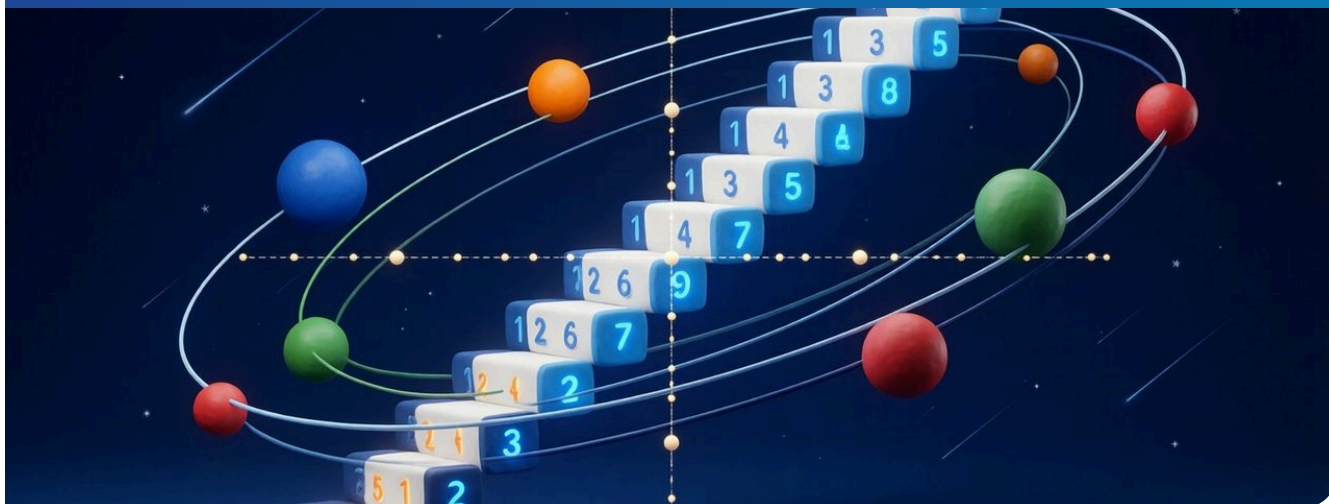
Ответ: 30.

По мотивам задачи отборочного этапа НТО Junior 2023/24 «Дерево возможностей»

Проверь себя

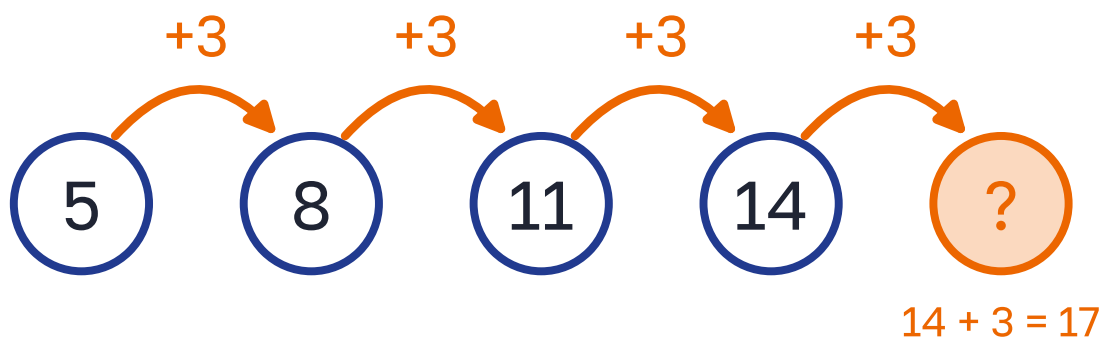
- 3.1.** У велосипедного замка три колёсика, на каждом – цифры от 0 до 9. Сколько разных кодов можно набрать на таком замке?
- 3.2.** Бот-экскурсовод краеведческого музея предлагает выбрать зал (их 4), язык рассказа (русский или английский) и длительность экскурсии: 15, 30 или 45 минут. Сколько разных сценариев экскурсии может получиться?
- 3.3.** Бот-меню школьной столовой предлагает на первое суп (2 вида) или салат (3 вида), а к нему – один из 4 напитков. Сколько разных обедов «первое + напиток» можно заказать через бота?
- 3.4.** Бот доставки сначала спрашивает: «Пицца или роллы?» Пицц в меню 6 видов, наборов роллов – 3. Потом бот предлагает один из двух соусов или «без соуса». Сколько разных заказов «еда + соус» можно оформить через бота?

Закономерности, последовательности, уравнения



Прогноз погоды на завтра, подсказка следующего слова в телефоне, цена поездки в приложении такси – за всем этим одна идея: найти правило и по нему угадать то, чего ещё не видели. В этой главе мы ищем правила в рядах чисел, записываем их буквами, решаем уравнения, а в конце проводим прямую через облако точек. Так работает самая простая модель машинного обучения – линейная регрессия. Задачи на эти темы обычно короткие: ряд чисел с пропуском, история, из которой нужно составить уравнение, таблица или график, по которым надо сделать прогноз. Чаще всего в ответ вводят одно число.

НА ПАЛЬЦАХ



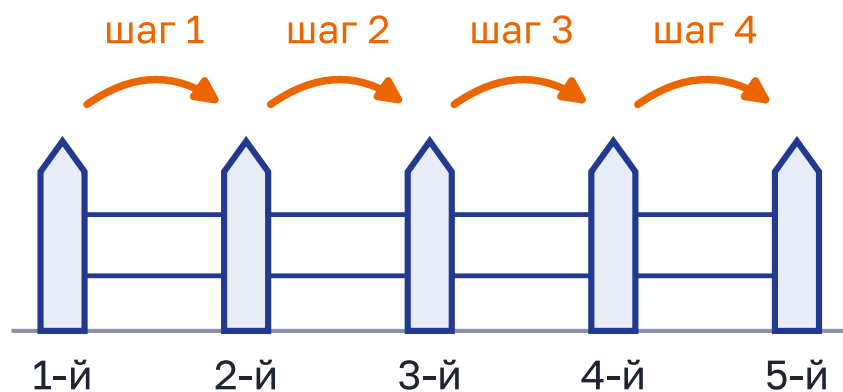
смотри не на сами числа, а на прыжки между соседями

Рисунок 4.1 – Ряд чисел, в котором спрятано правило

Между соседями каждый раз один и тот же прыжок: $+3$. Нашёл прыжок – знаешь, какое число будет дальше.

Запомни. Порядок поиска: разности, отношения, разности разностей, знакомые ряды, два ряда вперемешку. Правило, которое подходит к первым двум числам, но не подходит к третьему, – не правило.

НА ПАЛЬЦАХ



5 столбиков –
4 шага

шагов всегда
на один меньше,
чем номер

Рисунок 4.2 – Пять столбиков забора – четыре пролёта между ними

От первого члена до пятого – не пять шагов, а четыре. До сотого – 99.

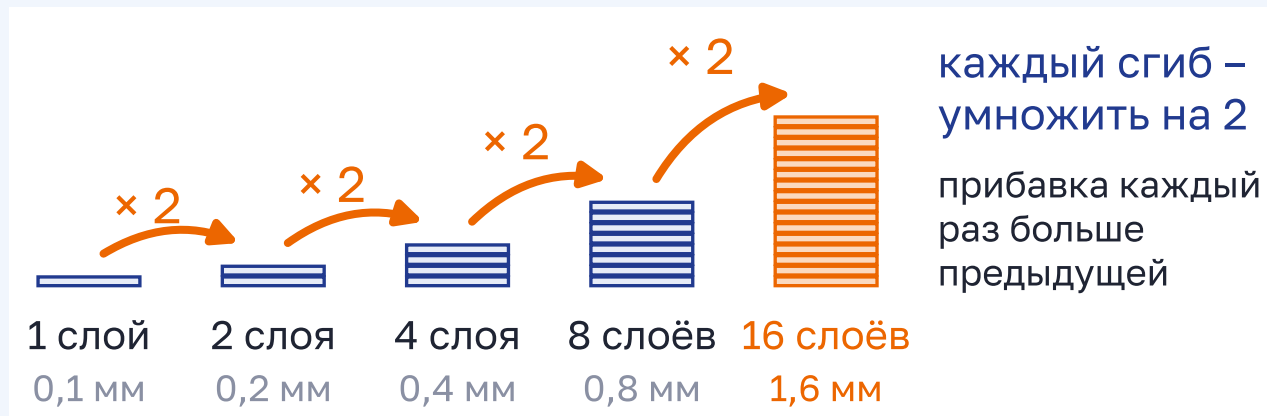


Рисунок 4.3 – Складываем лист пополам: слоёв каждый раз вдвое больше

Здесь каждый раз не прибавляют, а умножают на 2. Поэтому каждый следующий прирост больше предыдущего: сначала +0,1 мм, потом +0,2 мм, потом +0,4 мм.

Запомни. $a_n = a_1 + (n - 1) \cdot d$: прыжков на один меньше, чем номер. Номер по числу – это (число – первый член) : шаг, и к результату прибавить 1. Не делится нацело – такого числа в ряду нет. В геометрической прогрессии вместо прибавления – умножение: $b_n = b_1 \cdot q^{n-1}$.

Запомни. Переменная – место для числа. Подставил – посчитал в порядке: скобки, умножение и деление, сложение и вычитание. «На ...» – это плюс или минус, «в ... раз» – умножение или деление.

Запомни. Что делаешь с одной частью уравнения, делай и с другой. Действия с x снимаем в обратном порядке. Корень всегда проверяй подстановкой.

Запомни. $y = kx + b$: k – прибавка на каждую единицу x (со знаком), b – значение при $x = 0$. $k = (y_2 - y_1) : (x_2 - x_1)$. Прогноз – подставить x . «Когда станет...» – решить уравнение.

Запомни. Ошибка = настоящее – прогноз: плюс – модель занизила, минус – завысила. Две модели сравнивают по средней ошибке **без знака** – у кого она меньше, та и лучше. Одна странная точка – не повод гнуть прямую, а далеко за краем облака прямой не верь.

Разбор для старта

Задача 4.1 Какое число дальше

разминка · последовательности, разности, отношения

Найди следующее число в каждом ряду.

а) 7, 12, 17, 22, ...

б) 3, 6, 12, 24, ...

в) 2, 3, 6, 11, 18, 27, ...

Дано: три ряда чисел.

Найти: следующий член каждого ряда.

Решение

Шаг 1. Ряд а), разности. $12 - 7 = 5$, $17 - 12 = 5$, $22 - 17 = 5$. Все разности равны 5 – каждый раз прибавляем пять. Дальше $22 + 5 = 27$.

Шаг 2. Ряд б), разности. $6 - 3 = 3$, $12 - 6 = 6$, $24 - 12 = 12$ – разные. Числа растут всё быстрее – пробуем делить.

Шаг 3. Ряд б), отношения. $6 : 3 = 2$, $12 : 6 = 2$, $24 : 12 = 2$. Каждый раз умножаем на 2. Дальше $24 \cdot 2 = 48$.

Шаг 4. Ряд в), разности. $3 - 2 = 1$, $6 - 3 = 3$, $11 - 6 = 5$, $18 - 11 = 7$, $27 - 18 = 9$. Разности разные, но это нечётные числа подряд: 1, 3, 5, 7, 9. Разности разностей – всё время 2. Следующая разность $9 + 2 = 11$, следующий член $27 + 11 = 38$.

Ряд б) разгоняется, поэтому и ответ у него «прыгнул» сильнее всех: с 24 сразу на 48.

Посмотрим на все три ответа вместе. Ряды растут, и каждый ответ больше последнего числа своего ряда. В ряду б) рост ускоряется, поэтому и ответ «прыгнул» сильнее всех: с 24 сразу на 48.

 **Типичная ошибка.** Увидеть последнюю разность и прибавить её ещё раз. В ряду б) последняя разность 12, и выходит $24 + 12 = 36$. В ряду в) последняя разность 9, и выходит тоже $27 + 9 = 36$. Разность не обязана повторяться – сначала найди, по какому правилу меняются сами разности.

Ответ: а) 27; б) 48; в) 38.

Своя задача по программе сферы, модуль 2

Проверь себя

4.1. Найди следующее число в ряду: 81, 27, 9, 3, ...

4.2. Найди следующее число в ряду: 1, 3, 7, 15, 31, ...

4.3.* В первый день ролик кружка посмотрели 5 человек. Каждый следующий день ролик смотрели втрое больше людей, чем накануне. В какой по счёту день число просмотров за день впервые станет больше 10 000?

4.4. Нейросеть рисует узор из клеток. В первом ряду 5 клеток, в каждом следующем – на 4 больше, чем в предыдущем. Сколько клеток в 30-м ряду?

Данные: собираем и наводим порядок



Любая нейросеть начинается не с нейронов, а с таблицы. Кто-то должен собрать примеры, записать их в файл, выкинуть повторы и опечатки – и только потом модель чему-то научится. В главе пройдем этот путь целиком: откуда берутся данные, как устроены файлы CSV и JSON, как программа забирает записи с сайта страница за страницей и как навести порядок в таблице, прежде чем считать. На отборе прошлых лет встречались как раз такие задачи: сколько страниц скачать у сервера, какие три книги самые популярные в жанре. Ответ в них – число или несколько шифров через запятую.

НА ПАЛЬЦАХ

записи в тетради

12 января
мороз 18°
снег 42 см

13 января
мороз 24°
снег 45 см

14 января
мороз 12°
снег 51 см

столбец = признак

день	мороз, °	снег, см
12 января	18	42
13 января	24	45
14 января	12	51

строка = объект (один день)

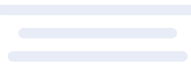
Рисунок 5.1 – Записи из тетради наблюдений превращаются в таблицу

Каждая запись в тетради – одна строка таблицы. Каждое поле записи – свой столбец.

Запомни. Строка – объект, столбец – признак (свойство), клетка – одно значение. Строка заголовка – не запись. Число клеток = число строк · число столбцов.

НА ПАЛЬЦАХ

число
42 см


текст
«Бердск»


да / нет


дата


складывать,
среднее

сравнить,
по алфавиту

посчитать,
сколько «да»

раньше
или позже

Рисунок 5.2 – Четыре типа данных и что с каждым можно делать

Тип значения решает, что с ним можно делать. Числа складывают, тексты сравнивают, «да» считают, даты ставят по порядку.

Запомни. Количественные данные можно складывать и усреднять, категориальные – только сравнивать и пересчитывать. Цифры в номере, шифре, индексе – это ярлык, а не количество. Числа из файла сначала переведи из текста в число.

НА ПАЛЬЦАХ

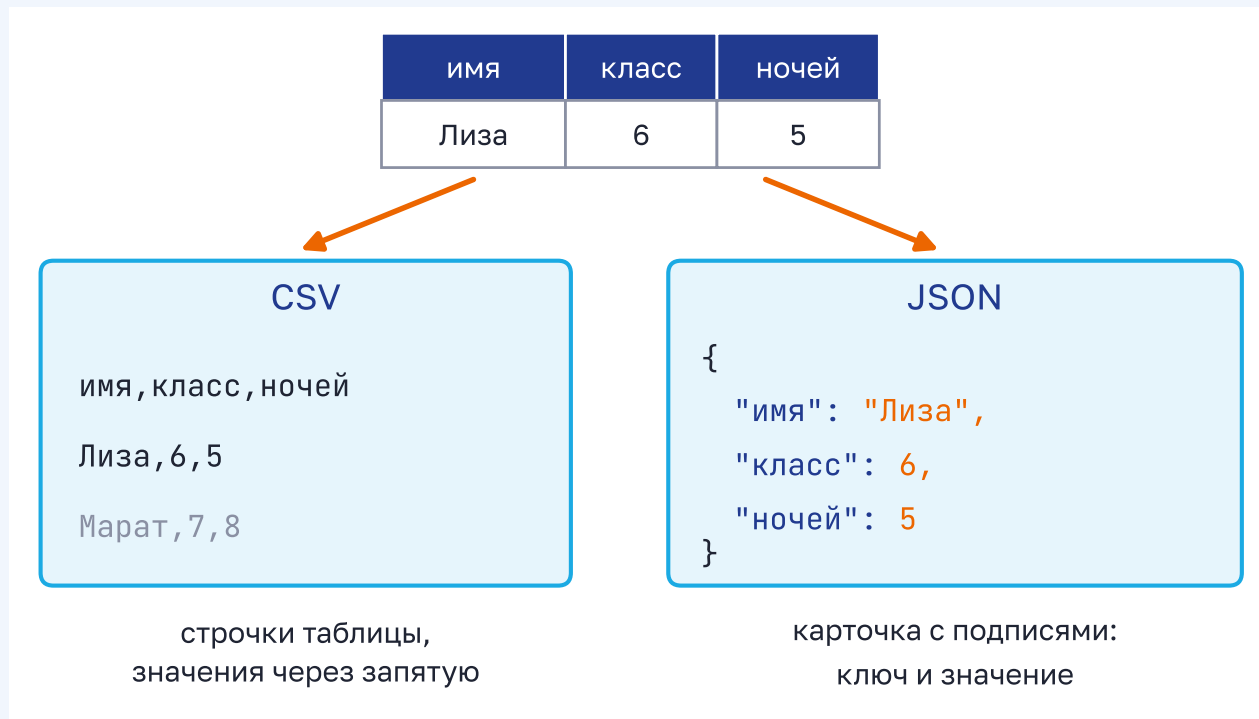


Рисунок 5.3 – Одна и та же запись в двух форматах

CSV записывает таблицу строками через запятую. JSON – как карточка, где у каждого значения есть подпись.

Запомни. CSV: первая строка – заголовок, записей на одну меньше, чем строк; запятых в строке на одну меньше, чем столбцов. JSON: {} – карточка с ключами, [] – список, счёт в списке с нуля.

Запомни. Страниц = всего : на странице, с округлением вверх. На последней = всего – (страниц – 1) · на странице. В цикле range(1, N + 1) : последнее число в range не входит.

Запомни. Четыре беды: пропуск, дубликат, опечатка, выброс. Пропуск не превращай в ноль. Для компьютера «Бердск» и «бердск» – разные слова. Сначала чистка, потом подсчёт.

Запомни. Фильтр – оставить строки по условию, сортировка – расставить по столбцу, подсчёт – свернуть в число: количество, сумма, среднее, максимум, минимум, количество уникальных. «Сумма по группе» и «среднее по группе» – разные вопросы.

Разбор для старта

Задача 5.1 Откуда взять данные

разминка · источники данных, датчик, снимок, архив, опрос

Шестиклассники из Кольцово готовят четыре проекта. Для каждого проекта выбери подходящий источник данных.

1. Каждые 10 минут записывать температуру и влажность воздуха в школьной теплице.
2. Узнать, в какие годы за последние полвека первый снег в Новосибирске выпал раньше 1 октября.
3. Выяснить, в какие кружки хотели бы ходить шестиклассники школы.
4. Посмотреть, какая часть Обского водохранилища покрыта льдом 1 декабря.

Источники: **А** – датчик; **Б** – снимки со спутника; **В** – архив метеонаблюдений; **Г** – опрос (анкета). Ответ запиши в виде «1А, 2...».

Дано: четыре проекта, четыре источника.

Найти: пары «проект – источник».

Решение

Шаг 1. Три вопроса к каждому проекту. Данные уже кто-то собрал или их надо измерить? Нужны сейчас или за прошлые годы? Их знают люди или показывают приборы?

Шаг 2. Проект 1. Температура и влажность прямо сейчас, и не один раз, а каждые 10 минут, днём и ночью. Человек так не просидит. Нужен прибор, который меряет сам, – датчик. 1А.

Шаг 3. Проект 2. Нужны прошлые 50 лет. Сегодня их не измеришь никаким прибором: можно только найти записи, которые метеостанции вели всё это время. Это архив. 2В.

Шаг 4. Проект 3. Кто в какой кружок хочет, не покажет ни один прибор – это знают только сами ребята. Надо их спросить – это опрос. 3Г.

Шаг 5. Проект 4. Водоохранилище огромное, лёд надо увидеть сразу целиком, сверху. Это снимок со спутника. 4Б.

Все буквы в деле? А, Б, В, Г – каждая досталась своему проекту, ни одна не осталась лишней и не повторилась.

 **Ловушка в проекте 2.** Рука тянется выбрать датчик: «там же про погоду». Датчик меряет только с той минуты, как его поставили. Про погоду 1980 года он ничего не знает. Прошлые – это всегда архив.

Ответ: 1А, 2В, 3Г, 4Б.

Проверь себя

5.1. Сопоставь задачу и источник данных. Задачи: 1) сколько шагов в день делает ученик; 2) какие фильмы больше нравятся семиклассникам школы; 3) как менялось население Новосибирска за последние сто лет; 4) программа должна каждые 5 минут получать с сайта свежее расписание электричек. Источники: А – датчик в фитнес-браслете; Б – опрос; В – архив открытых данных статистики; Г – API сайта.

5.2. Пятиклассники из Бердска готовят четыре проекта. Подбери каждому источник данных: 1) как менялась высота снега в Бердске каждую зиму за последние 30 лет; 2) сколько раз за учебный день открывается дверь в школьную столовую; 3) какие настольные игры хотели бы купить в класс сами ребята; 4) какая часть огромного поля за селом уже зазеленела весной – смотреть надо на всё поле сразу, сверху. Источники: А – снимки со спутника; Б – опрос; В – датчик на двери; Г – архив метеонаблюдений.

5.3. В наборе данных о погоде есть столбцы: «Дата», «Температура, °C», «Был снег» (да/нет), «Направление ветра» (С, Ю, З, В), «Скорость ветра, м/с», «Номер станции». Сколько из них количественных? Назови их.

5.4. В столбце «Дата» записаны даты наблюдений: 15.09.2026, 03.11.2026, 20.10.2026, 28.09.2026. Программа отсортировала их командой `sorted`, как обычный текст. Какая дата окажется первой в списке? Какая из этих дат на самом деле самая ранняя?

Графики и анализ данных



Таблица на сто строк молчит, пока её не превратишь в картинку. Один взгляд на график – и видно, где максимум, куда всё растёт и есть ли вообще то, что ты ищешь. В этой главе мы учимся выбирать график под вопрос, читать его без ошибок, считать среднее и медиану, проверять догадки и замечать графики, которые обманывают. На отборе такие задачи выглядят так: дан график или таблица, нужно ввести число, выбрать верные выводы из списка или сопоставить ситуацию и вид диаграммы.

НА ПАЛЬЦАХ

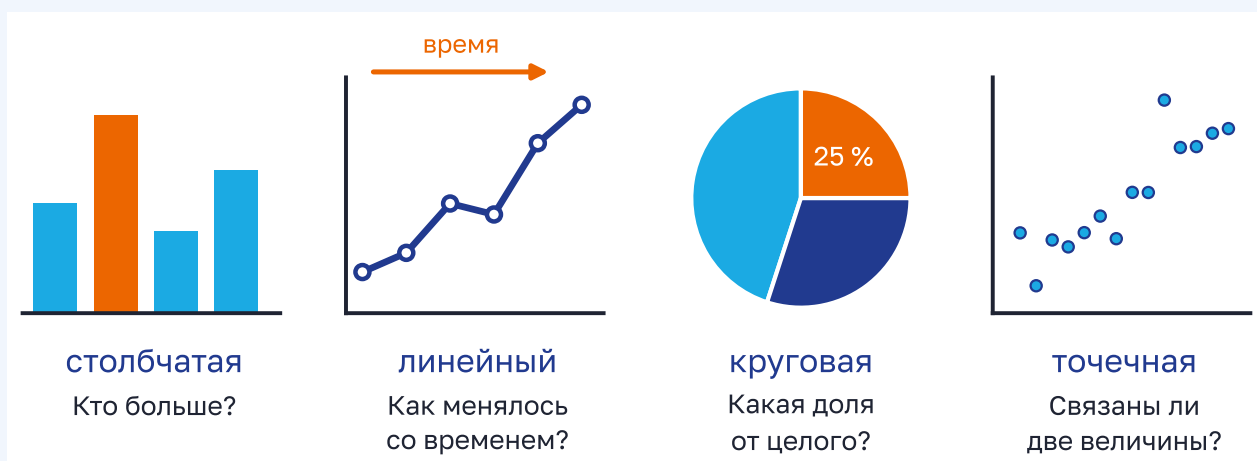
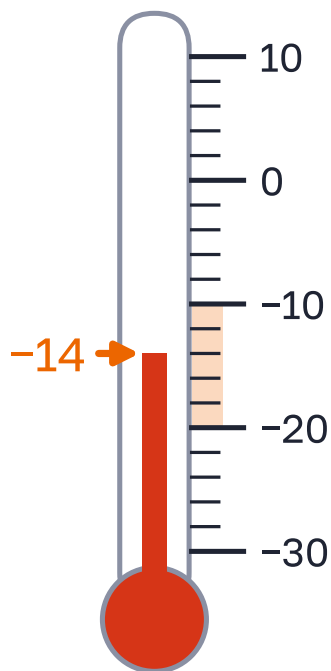


Рисунок 6.1 – Четыре вида графиков – и вопрос, на который отвечает каждый

Сначала реши, какой у тебя вопрос. Вид графика выбирают под вопрос, а не под настроение.

Запомни. Сравнить – столбики. Время – линия. Доли одного целого – круг. Две величины у каждого объекта – точки.

НА ПАЛЬЦАХ



1. Между подписями -20 и -10 разница 10 градусов
2. Между ними 5 делений
3. Одно деление – $10 : 5 = 2$ градуса
4. Столбик на 3 деления выше -20 :
 $-20 + 3 \cdot 2 = -14$ градусов

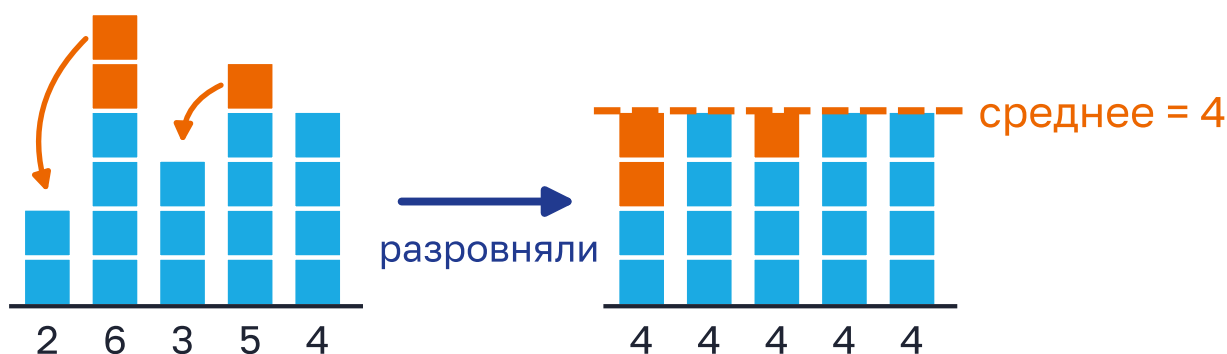
Рисунок 6.2 – Градусник за окном и цена деления

Подписана не каждая чёточка, а только каждая пятая. Сколько градусов в одном промежутке между чёточками, считаем сами: разницу подписей делим на число промежутков.

Запомни. Сначала подписи, потом картинка: что по осям, в каких единицах, с какого числа начинается ось. Цена деления – разница соседних подписей, делённая на число промежутков между ними.

было: 2, 6, 3, 5, 4

стало: у всех поровну



всего кубиков 20, столбиков 5: $20 : 5 = 4$

Рисунок 6.3 – Среднее – это высота, которая получится, если разровнять все столбики

Возьми лишние кубики с высоких столбиков и отдай низким. Когда все сравнялись, их высота и есть среднее.

Запомни. Медиана – середина **упорядоченного** ряда; при чётном количестве – среднее двух средних. Выброс тянет среднее и почти не трогает медиану. Сумма, среднее, максимум по группам – разные вопросы и разные ответы.

Запомни. Гипотеза – конкретная догадка, которую можно проверить таблицей. Точки полосой вверх или вниз – связь есть, кашей – не видно. Связь не доказывает, что одно вызывает другое.

Запомни. Перед сравнением столбиков посмотри, с какого числа начинается ось. Сравнивай сами числа, а не высоту картинки: во сколько раз – делением, на сколько процентов – от того значения, с которым сравниваешь.

Запомни. Дашборд – главное на одном экране: плитки с числами и несколько графиков. Эссе – вопрос, данные, график, вывод с числами и оговорки. Вывод без чисел – это мнение, а не вывод.

Разбор для старта

Задача 6.1 Год в Новосибирске

разминка · чтение линейного графика, размах, изменение

По графику средней температуры в Новосибирске (раздел 6.2) ответь на вопросы. а) Какой месяц самый холодный и какая в нём средняя температура? б) Сколько месяцев в году средняя температура выше нуля? в) Найди размах средних температур за год. г) Между какими соседними месяцами средняя температура выросла сильнее всего и на сколько градусов?

Дано: линейный график, 12 точек с подписанными значениями.

Найти: самый холодный месяц; число месяцев выше нуля; размах; самый большой рост между соседями.

Решение

Шаг 1. Самый холодный месяц. Ищем самую нижнюю точку. Это январь, -17°C .

Шаг 2. Выше нуля. Считаем точки над линией нуля: апрель (2), май (10), июнь (17), июль (19), август (16), сентябрь (10), октябрь (2). Всего 7 месяцев.

Шаг 3. Размах. Максимум – июль, 19°C , минимум – январь, -17°C . Размах $19 - (-17) = 19 + 17 = 36$ градусов.

Шаг 4. Самый большой рост. Рост – это «следующий минус предыдущий». Смотрим весну, где линия идёт вверх: январь $\rightarrow$ февраль: $-15 - (-17) = 2$; февраль $\rightarrow$ март: $-8 - (-15) = 7$; март $\rightarrow$ апрель: $2 - (-8) = 10$; апрель $\rightarrow$ май: $10 - 2 = 8$; май $\rightarrow$ июнь: $17 - 10 = 7$; июнь $\rightarrow$ июль: $19 - 17 = 2$. Самый большой – 10 градусов, с марта по апрель. На графике это самый крутой подъём линии.

В Новосибирске и правда холоднее всего в январе, а в апреле сходит снег – так что резкий скачок весной не удивляет.

 **Типичная ошибка.** Потерять минус у -17 и посчитать размах как $19 - 17 = 2$. Размах не может быть меньше разницы между любыми двумя точками, а июль и январь явно отличаются больше чем на 2 градуса. Ещё путают «самый большой рост» с самой высокой точкой (июль). Высокая точка – не то же самое, что крутой подъём.

Ответ: а) январь, -17°C ; б) 7 месяцев; в) 36°C ; г) с марта по апрель, на 10°C .

Своя задача по программе сферы, модуль 4

Проверь себя

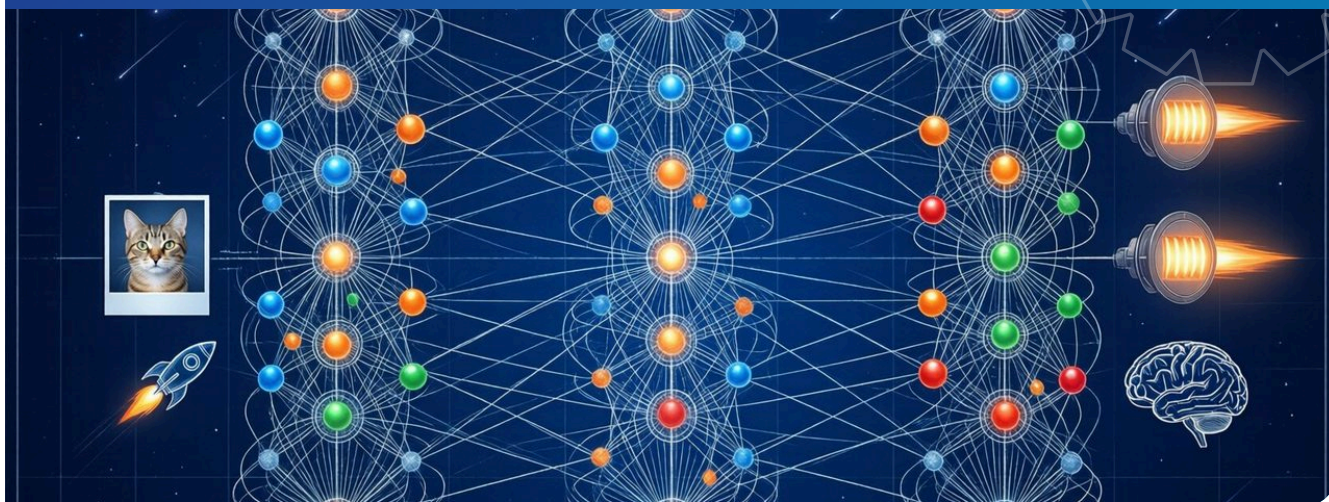
6.1. По графику температуры в Новосибирске (раздел 6.2): а) сколько месяцев в году средняя температура ниже -10°C ? б) Какой месяц теплее: май или сентябрь?

6.2. На вертикальной оси подписаны 0, 50, 100, 150, а между соседними подписями – по 5 одинаковых промежутков. Сколько стоит одно деление? Столбик кончается на третьем делении выше подписи 100. Какое число он показывает?

6.3. Сопоставь ситуацию и вид графика (А – столбчатая, Б – линейный, В – круговая, Г – точечная). 1) Как менялось число подписчиков канала кружка по месяцам. 2) Какую долю учеников школы привозит автобус, привозят родители, приходят пешком (каждый назвал один способ). 3) Рост и масса 25 учеников: есть ли связь. 4) Сколько книг прочитал каждый из пяти друзей.

6.4. Два опроса в 6 классах. Первый: «Где ты провёл больше всего времени летом?» (один ответ) – на даче 45 %, в лагере 30 %, в городе 25 %. Второй: «Где ты бывал летом? Можно отметить несколько» – на даче 60 %, в лагере 35 %, в городе 70 %. Для какого опроса можно нарисовать круговую диаграмму? Какой график нужен для другого?

Нейросети своими руками



В главе 1 ты познакомился с одним нейроном: входы умножаются на веса, складываются и сравниваются с порогом. Теперь соберём из нейронов сеть, посчитаем её ответ вручную и посмотрим, как она учится, – что такое эпоха, скорость обучения и почему сеть иногда «зубрит». В конце главы – сервисы, где нейросеть собирают мышкой: Teachable Machine, Machine Learning for Kids и конструктор GraphNet, с которыми работают на сфере. В задачах будем считать выход маленькой сети и связи между слоями, искать по таблице точности, где сеть переобучилась, и подбирать сеть под задачу.

каждый судья смотрит на всё

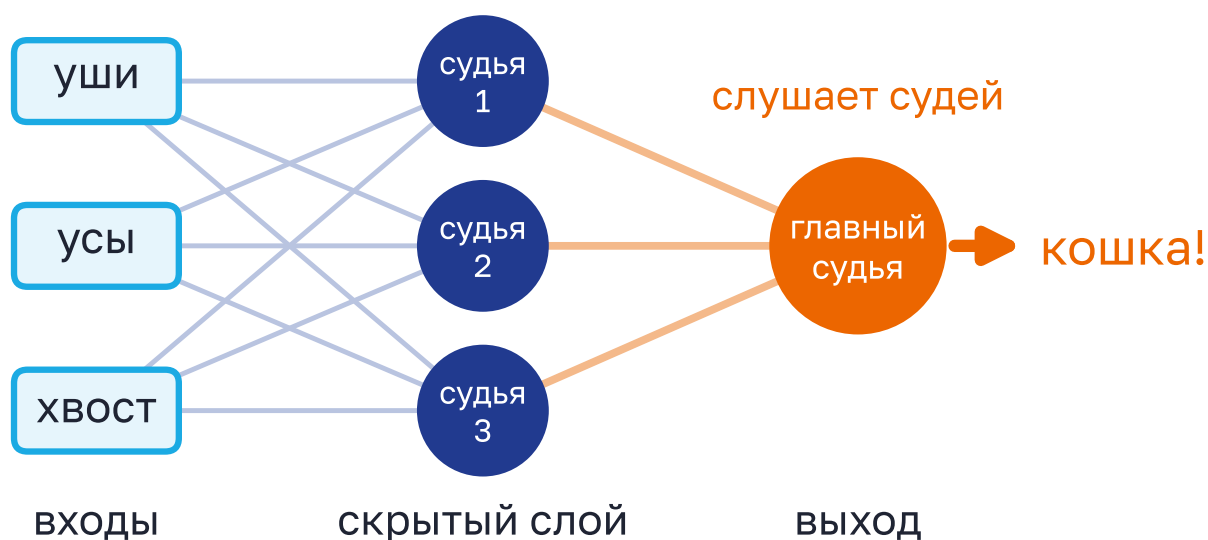


Рисунок 7.1 – Нейросеть похожа на жюри конкурса

Каждый судья скрытого слоя смотрит на все признаки сразу и ставит свою оценку. Главный судья слушает судей, у каждого – свой вес. Его решение и есть ответ сети.

Запомни. Связи между соседними слоями = произведение числа нейронов в них; связи сети – сумма по всем парам соседних слоёв. Смещения есть у каждого нейрона, кроме входов. Входов – сколько признаков, выходов – сколько классов, для числа – один выход.

Шаг 1: скрытый слой



Шаг 2: выход

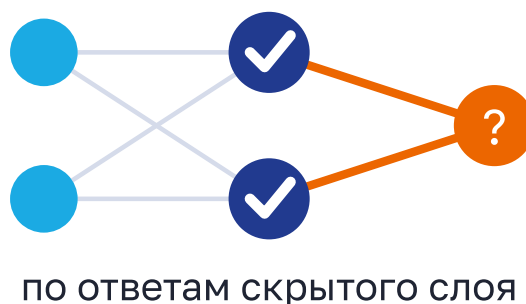


Рисунок 7.2 – Сеть считает по очереди, слой за слоем

Пока скрытый слой не досчитал, выходу нечего складывать. Сначала находим ответы всех нейронов скрытого слоя, потом по ним – ответ выхода.

Запомни. Порядок всегда один: сумма «вход · вес» по всем входам, плюс смещение, потом функция активации. Считаем слой за слоем: входами выходного нейрона служат **ответы** скрытых нейронов, а не исходные x .

НА ПАЛЬЦАХ

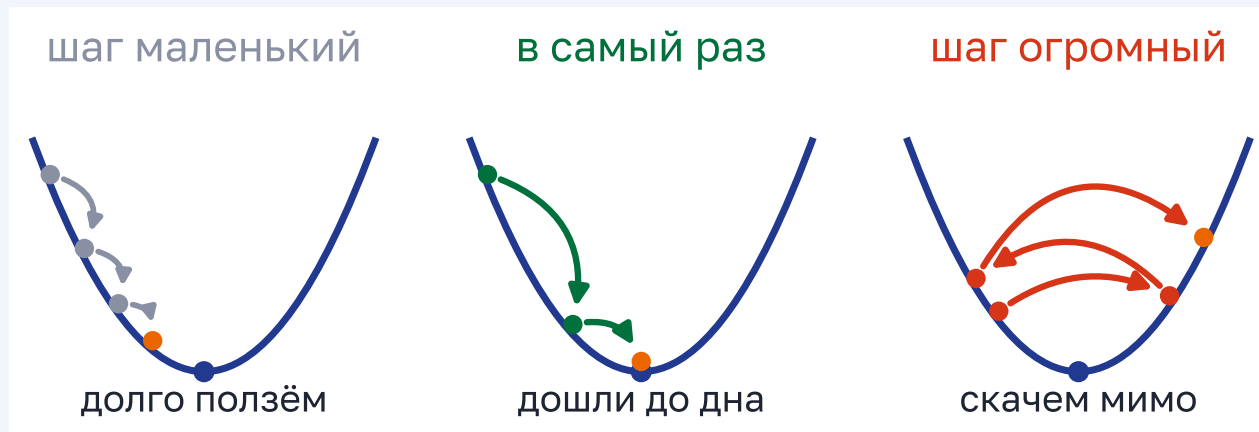


Рисунок 7.3 – Ищем дно ямы, шагая вниз. Высота – ошибка сети, шаг – поправка веса

Маленькими шажками до дна идти долго. Слишком большими – перепрыгиваешь его и скачешь со склона на склон.

Запомни. Ошибка = верный ответ – ответ модели. Новый вес = вес + скорость · ошибка · вход. Эпоха – один проход по всем обучающим примерам. Большая скорость – ошибка скачет и растёт, маленькая – учимся медленно.

Запомни. Переобучение – на обучении точность растёт, а на проверке падает. Лучшая эпоха – та, где **выше всего точность на проверке**, а не на обучении. Низкая точность на обеих выборках – недообучение.

Запомни. Low-code: собрал примеры по классам → обучил → проверил на новых примерах. Классы – сопоставимо, без сильного перекоса, примеры – разные. Ответ сети – класс с наибольшей уверенностью.

Разбор для старта

Задача 7.1 Сколько связей в сети

разминка · полносвязные слои, число связей

Сеть по 6 признакам поездки (расстояние, погода, день недели и так далее) советует, чем добраться из Новосибирска до Академгородка: автобусом, электричкой или на машине. В сети один скрытый слой из 4 нейронов, все соседние слои полносвязные. Сколько связей (весов) в этой сети? Смещения не считай.

Дано: сеть 6–4–3, слои полносвязные.

Найти: N – число связей.

Решение

Шаг 1. Выписываем слои. 6 признаков – 6 входов. Скрытый слой – 4 нейрона. Способов доехать 3 – значит, 3 выхода, по одному на класс. Сеть 6–4–3.

Шаг 2. Связи между входами и скрытым слоем. Каждый из 6 входов соединён с каждым из 4 нейронов: $6 \cdot 4 = 24$ связи.

Шаг 3. Связи между скрытым слоем и выходами. $4 \cdot 3 = 12$ связей.

Шаг 4. Всего. $N = 24 + 12 = 36$.

Прикинем. Нейронов в сети всего $6 + 4 + 3 = 13$, а связей почти втрое больше: каждый нейрон тянет стрелки сразу к нескольким соседям. Если бы спросили параметры вместе со смещениями, добавилось бы $4 + 3 = 7$ смещений – вышло бы 43.

 **Типичная ошибка.** Перемножить все слои подряд: $6 \cdot 4 \cdot 3 = 72$. Слои соединены парами, поэтому произведения соседей складывают. Бывает и так: посчитали нейроны ($6 + 4 + 3 = 13$), а спрашивали связи.

Ответ: 36.

Своя задача по программе сферы, модуль 5

Проверь себя

7.1. Сеть 3–5–2, все соседние слои полносвязные. Сколько в ней связей (без смещений)?

7.2. Сеть 8–6–6–1 (два скрытых слоя по 6 нейронов), слои полносвязные. а) Сколько в ней связей? б) Сколько всего параметров, если к связям добавить смещения? Ответ – два числа через точку с запятой.

7.3.* а) Сеть 10–?–4 с одним скрытым слоем, слои полносвязные, всего 196 связей. Сколько нейронов в скрытом слое? б) В сети 10–?–?–4 два скрытых слоя с одинаковым числом нейронов, всего 312 связей. Сколько нейронов в каждом скрытом слое? (Подсказка: подставляй по очереди 10, 11, 12... и считай связи, пока не получится 312.)

7.4. Суммы трёх нейронов скрытого слоя (уже со смещением) равны –4; 0; 2,5. Функция активации – ReLU. Запиши ответы нейронов через точку с запятой.

Python, диалоги и промпты



Эта глава – про разговор с компьютером. Сначала на языке Python: переменные, условия, циклы, списки и строки. Типичная олимпиадная задача по программированию – короткий код и вопрос «что он напечатает?» или «где здесь ошибка?». Потом посмотрим, как устроен разговор чат-бота: его рисуют схемой из реплик и стрелок и считают по ней ветки. А в конце научимся просить нейросеть так, чтобы она поняла с первого раза, и проверять код, который она за нас написала.

НА ПАЛЬЦАХ

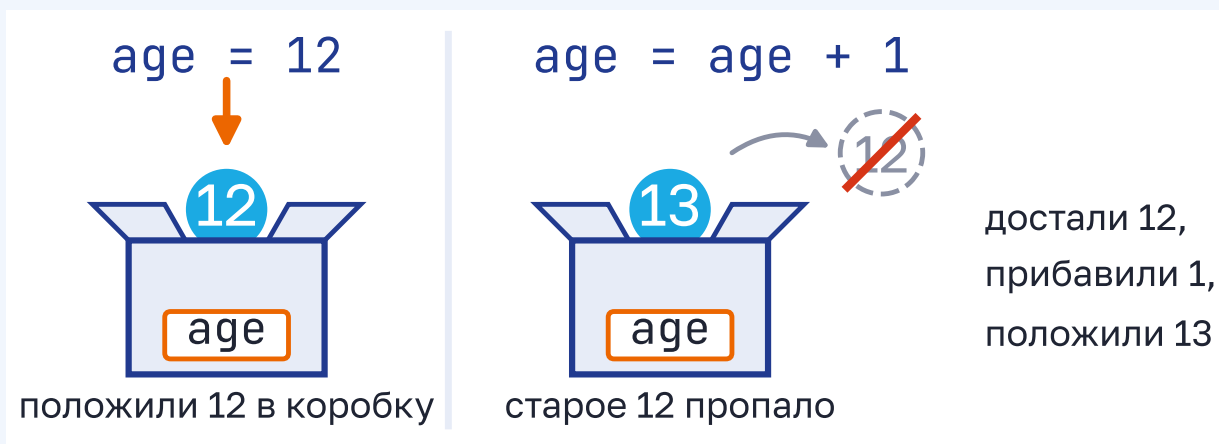


Рисунок 8.1 – Переменная – коробка с наклейкой-именем

Знак = кладёт значение в коробку. Положили новое – старое пропало, в коробке всегда лежит что-то одно.

Запомни. = кладёт значение в переменную, новое вытесняет старое. Правая часть считается целиком, а уже потом результат кладётся в коробку слева. `input` всегда отдаёт текст – для чисел оборачивай его в `int(...)`.

НА ПАЛЬЦАХ

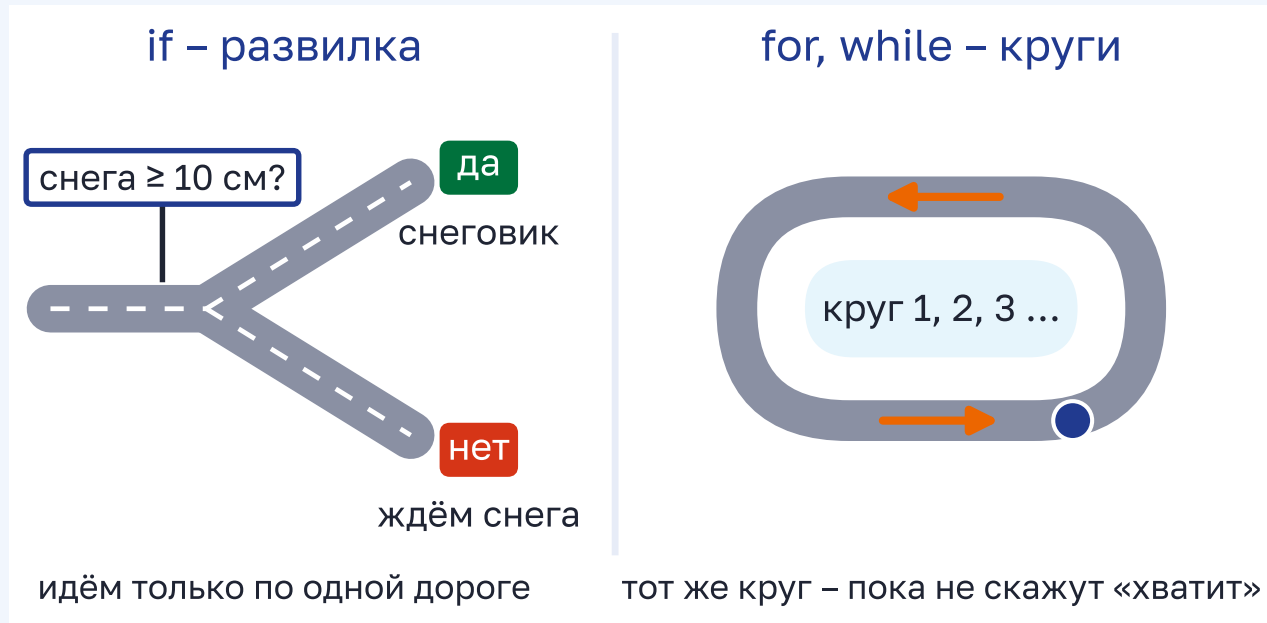


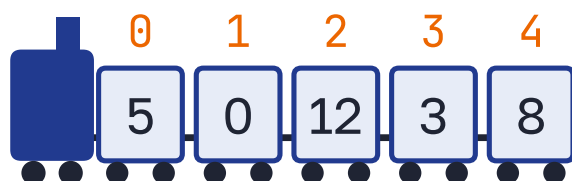
Рисунок 8.2 – Условие – развилка, цикл – круги по стадиону

На развилке идёшь по одной дороге из двух. На стадионе бежишь один и тот же круг, пока тренер не скажет «хватит».

Запомни. После `if`, `elif`, `else`, `for`, `while` – двоеточие, а строки внутри – с отступом. Из цепочки `if` – `elif` – `else` выполняется только первая подходящая ветка. `range(1, 6)` – это 1, 2, 3, 4, 5, без шести.

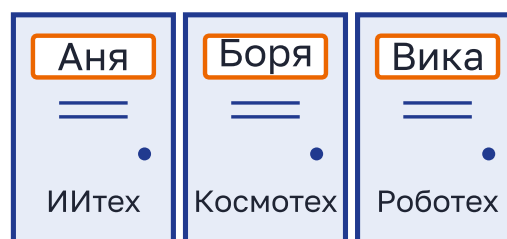
список: ищем по номеру

номер:

`sneg[2] → 12`

номер 2 – это третий вагон

словарь: ищем по имени

`kruzhki["Боря"] → Космотех`

на дверце – ключ, внутри – значение

Рисунок 8.3 – Список – вагоны с номерами, словарь – шкафчики с именами

В списке ищешь по номеру вагона, и считать начинают с нуля. В словаре ищешь по подписи на дверце.

Запомни. Номера в списке и в строке начинаются с 0, последний элемент – `[-1]`, а номер последнего равен длине минус один. Словарь ищет по ключу, ключи не повторяются. "Снег" и "снег" для Python – разные слова.

Запомни. Узел – реплика, стрелка – переход, ветка – путь от начала до конца. Длина ветки – число реплик в ней. Число веток = число у конца, если подписать у каждого узла сумму чисел узлов, из которых в него входят стрелки. Интент – намерение человека, которое бот узнаёт по фразе.

Запомни. Промпт = контекст и роль + цель + исходные данные + ограничения + формат. Чего не написал – нейросеть додумает сама. Любой ответ нейросети – черновик, пока ты его не проверил.

Запомни. Код от нейросети проверяют запуском на тестах, где ответ известен заранее. Падение программы – это подсказка, где искать. Самая опасная ошибка – когда программа работает и выдаёт правдоподобное неверное число.

Разбор для старта

Задача 8.1 Сколько снеговиков

разминка · переменные, ввод, строки и числа

Программа спрашивает, сколько снеговиков слепили вчера и сколько сегодня:

```
vchera = input()
segodnya = input()
print(vchera + segodnya)
print(int(vchera) + int(segodnya))
vsego = int(vchera) * 2
vchera = vsego - int(segodnya)
print(vchera, vsego)
```

Пользователь ввёл сначала 7, потом 5. Запиши, что появится на экране: три строки.

Дано: код из 7 строк; ввод 7, затем 5.

Найти: три строки вывода.

Решение

Шаг 1. Что лежит в переменных после ввода. `input` отдаёт текст. В `vchera` лежит строка "7", в `segodnya` – строка "5". Не числа!

Шаг 2. Строка 3. "7" + "5" – это склейка строк: "75". Первая строка вывода – 75.

Шаг 3. Строка 4. Здесь обе строки превращены в числа: $7 + 5 = 12$. Вторая строка вывода – 12.

Шаг 4. Строки 5–6, трассировка.

Таблица 8.1

Строка	Считаем	vchera	segodnya	vsego
5	$7 \cdot 2 = 14$	"7"	"5"	14
6	$14 - 5 = 9$	9	"5"	14

В строке 6 в `vchera` кладут новое значение – теперь это число 9, а строка "7" пропала.

Шаг 5. Строка 7. Печатаем `vchera` и `vsego` через пробел: 9 14.

Прикинем на глаз: 12 снеговиков за два дня – обычная работа для двора. А 75 – явно не сумма, а две цифры, приставленные друг к другу: так + обходится со строками.

⚠ Типичная ошибка. Написать в первой строке 12: не заметить, что `input` отдаёт текст. А в последней строке легко напечатать 7 14, забыв, что строка 6 заменила значение `vchera`.

Ответ: 75, 12, 9 14.

Своя задача по программе сферы, модуль 3

Проверь себя

8.1. Что напечатает программа? Запиши два числа через пробел.

```
x = 4
y = x * 3
x = y - x
print(x, y)
```

8.2. Что напечатает каждая строка?

```
print("3" + "4")
print(3 + 4)
print("ла" * 3)
```

8.3. На экскурсию в планетарий едут 58 школьников. В один микроавтобус садится 12 человек. Что напечатает `print(58 // 12, 58 % 12)`? Сколько микроавтобусов нужно заказать, чтобы поехали все?

8.4. В школьной игре «Снежный квест» за баллы дают значки. Что напечатает программа, если ввести 73?

```
bally = int(input())
if bally ≥ 85:
    print("золотой значок")
elif bally ≥ 70:
    print("серебряный значок")
elif bally ≥ 50:
    print("бронзовый значок")
else:
    print("значок участника")
```

Ответы и решения

Сначала реши задачу сам и только потом сверяйся. Не сошлось – найди в решении шаг, на котором твой путь разошёлся с нашим.

К главе 1. Что такое искусственный интеллект

1.1. Ответ: 2132.

Подсказка: в пункте г) число людей – величина, а не номер класса.

1.2. Ответ: 246.

Подсказка: про каждое устройство спроси: оно выполняет правило, которое записал человек, или научилось на примерах?

1.3. Ответ: удалить «Фамилия», «Номер в журнале», «Прошёл в финал»; тип – классификация.

1.4. Ответ: ГБДАВ.

К главе 2. Множества и логика

2.1. Ответ: 5 учеников.

Подсказка: задача «наоборот»: сначала найди, сколько учеников любят хотя бы один предмет, а потом – пересечение.

2.2. Ответ: 7; 17.

Подсказка: общая часть не больше меньшей группы, а хотя бы в одну секцию ходят не больше 30 человек – весь класс.

2.3. Ответ: 10.

2.4. Ответ: $n(A \cap B) = 3$, $n(A \cup B) = 13$.

Подсказка: общие – числа, которые делятся и на 2, и на 3.

К главе 3. Комбинаторика и вероятность

3.1. Ответ: 1000.

Подсказка: цифры на колёсиках могут повторяться – код 555 тоже годится.

3.2. Ответ: 24.

3.3. Ответ: 20.

3.4. Ответ: 27.

К главе 4. Закономерности, последовательности, уравнения

4.1. Ответ: 1.

Подсказка: ряд убывает, но разности разные – проверь отношения: каждый раз делим на 3.

4.2. Ответ: 63.

Подсказка: разности 2, 4, 8, 16 сами растут не на одно и то же, а удваиваются.

4.3. Ответ: на 8-й день.

Подсказка: это ряд с умножением на 3; выпиши просмотры по дням, начиная с 5 в первый день.

4.4. Ответ: 121 клетка.

К главе 5. Данные: собираем и наводим порядок

5.1. Ответ: 1А, 2Б, 3В, 4Г.

5.2. Ответ: 1Г, 2В, 3Б, 4А.

5.3. Ответ: 2 – «Температура» и «Скорость ветра».

5.4. Ответ: первой окажется 03.11.2026, а самая ранняя – 15.09.2026.

Подсказка: текст сравнивается знак за знаком слева, а первым в такой дате стоит день, а не год.

К главе 6. Графики и анализ данных

6.1. Ответ: а) 3 месяца; б) одинаково, в обоих 10 °С.

Подсказка: март (–8) и ноябрь (–7) не подходят: они теплее –10.

6.2. Ответ: 10; 130.

6.3. Ответ: 1 – Б, 2 – В, 3 – Г, 4 – А.

6.4. Ответ: круговая – для первого опроса; для второго – столбчатая.

Подсказка: сложи проценты: в первом опросе вышло 100 %, во втором 165 % – ответы пересекались.

К главе 7. Нейросети своими руками

7.1. Ответ: 25.

7.2. Ответ: 90; 103.

Подсказка: связи считай для каждой пары соседних слоёв – их тут три; смещения есть у всех нейронов, кроме входов.

7.3. Ответ: а) 14; б) 12.

Подсказка: обозначь число нейронов скрытого слоя буквой h : связей $10 \cdot h + h \cdot 4$; во втором пункте не забудь связи между двумя скрытыми слоями, $h \cdot h$.

7.4. Ответ: 0; 0; 2,5.

Подсказка: ноль не отрицательный – ReLU оставляет его нулём.

К главе 8. Python, диалоги и промпты

8.1. Ответ: 8 12 .

8.2. Ответ: 34 , 7 , лалала .

8.3. Ответ: программа напечатает 4 10 ; заказать нужно 5 микроавтобусов.

Подсказка: `//` – сколько автобусов заполнятся целиком, `%` – сколько человек осталось; их тоже надо везти, поэтому округляем вверх.

8.4. Ответ: серебряный значок .

Подсказка: из цепочки `if - elif - else` выполняется только первая верная ветка – дальше проверки не идут.

Всё, что может понадобиться под рукой во время решения: формулы и правила из глав, мини-шпаргалка по Python, словарь слов из мира ИИ и все «Запомни» из глав на нескольких страницах.

Шпаргалка: всё «Запомни» из глав

Глава 1. Что такое искусственный интеллект

Запомни. ИИ – программы для задач, где человеку нужен интеллект. Машинное обучение – та часть ИИ, которая учится на примерах, а нейросети – часть машинного обучения. Если программа только выполняет правила, записанные человеком (звонок, калькулятор, светофор по таймеру), – это не ИИ.

Запомни. Строка таблицы – объект. Столбцы, по которым угадываем, – признаки, а столбец, который надо угадать, – ответ (целевая переменная). Номера и фамилии, которые лишь называют строку, – не признаки: перед обучением их удаляют вместе с ответом.

Запомни. Готовые группы и ответы в примерах – классификация. Число – регрессия. Ответов нет, ищем группы сами – кластеризация. Первым делом смотри, **что должно получиться на выходе**.

Запомни. Обучающая выборка – учимся, проверочная – сдаём экзамен. Проверочные примеры модель при обучении не видит. Точность, которой можно верить, – только на проверочной выборке.

Запомни. Нейрон: умножь каждый вход на его вес, сложи, сравни сумму с порогом. $S \geq T$ – выход 1, иначе 0. Отрицательный вес работает «против». Обучение – это подбор весов.

Запомни. Точность = верных : всего · 100 %. Доля ошибок = 100 % – точность. Считаем на проверочной выборке. Для регрессии – средняя ошибка: насколько в среднем промахнулись, без учёта знака.

Глава 2. Множества и логика

Запомни. Множество – набор без повторов, порядок в нём не важен. $n(A)$ – сколько в нём элементов. Подмножество – часть, все элементы которой есть в большом множестве.

Запомни. $A \cap B$ – «и то, и другое», $A \cup B$ – «хотя бы одно», « A , но не B » – круг A без общей части, «не A » – всё за пределами круга A . Одно понятие может занимать и один кусочек картинки, и несколько.

Запомни. Хотя бы в одном = $n(A) + n(B) - n(A \cap B)$. Ни в одном = все минус хотя бы в одном. Для трёх кругов заполняй кусочки от центра наружу, а в конце проверь сумму формулой.

Запомни. И – истина, только если истинны оба. ИЛИ – ложь, только если ложны оба. НЕ – наоборот. Порядок: скобки, НЕ, И, ИЛИ. Для k высказываний в таблице истинности 2^k строк: 4, 8, 16...

Запомни. $\&$ – И – пересечение: найдётся меньше. $|$ – ИЛИ – объединение: найдётся больше. Пересечение по трём числам: $n(A \cap B) = n(A) + n(B) - n(A \cup B)$.

Глава 3. Комбинаторика и вероятность

Запомни. Выбор по шагам – перемножаем, сколько вариантов на каждом шаге: $a \cdot b \cdot c$. Так можно, только если число вариантов на шаге не зависит от того, что выбрали раньше.

Запомни. «И» – берём по варианту на каждом шаге – умножаем. «Или» – берём одно из нескольких групп – складываем. Если у групп есть общие элементы, общее вычитаем один раз.

Запомни. Расставляем по местам: на каждое следующее место вариантов на один меньше. Все n предметов в ряд – $n! = n \cdot (n - 1) \cdot \dots \cdot 1$. Пары, где порядок не важен, – $\frac{n \cdot (n - 1)}{2}$.

Запомни. $P = \frac{\text{подходящие}}{\text{все}}$, и все исходы должны быть равновероятными. $0 \leq P \leq 1$. «Не случится» – $1 - P$. У двух монет 4 исхода, у двух кубиков – 36: каждую монету и каждый кубик считай отдельно.

Запомни. Вероятность – сколько должно выходить в среднем, частота – сколько вышло в нашем опыте. На малом числе опытов они могут сильно различаться, на большом – сближаются. Ожидаемое число случаев = $P \cdot N$.

Запомни. Проценты модели – её оценка, а не факт про один снимок. Честность модели проверяют на многих ответах: из тех, где она «уверена на 80 %», верными должны оказаться около 80 из 100. Вероятность фразы из нескольких слов – произведение вероятностей вдоль её ветки дерева.

Глава 4. Закономерности, последовательности, уравнения

Запомни. Порядок поиска: разности, отношения, разности разностей, знакомые ряды, два ряда вперемешку. Правило, которое подходит к первым двум числам, но не подходит к третьему, – не правило.

Запомни. $a_n = a_1 + (n - 1) \cdot d$: прыжков на один меньше, чем номер. Номер по числу – это (число – первый член) : шаг, и к результату прибавить 1. Не делится нацело – такого числа в ряду нет. В геометрической прогрессии вместо прибавления – умножение: $b_n = b_1 \cdot q^{n-1}$.

Запомни. Переменная – место для числа. Подставил – посчитал в порядке: скобки, умножение и деление, сложение и вычитание. «На ...» – это плюс или минус, «в ... раз» – умножение или деление.

Запомни. Что делаешь с одной частью уравнения, делай и с другой. Действия с x снимаем в обратном порядке. Корень всегда проверяй подстановкой.

Запомни. $y = kx + b$: k – прибавка на каждую единицу x (со знаком), b – значение при $x = 0$. $k = (y_2 - y_1) : (x_2 - x_1)$. Прогноз – подставить x . «Когда станет...» – решить уравнение.

Запомни. Ошибка = настоящее – прогноз: плюс – модель занизила, минус – завысила. Две модели сравнивают по средней ошибке **без знака** – у кого она меньше, та и лучше. Одна странная точка – не повод гнуть прямую, а далеко за краем облака прямой не верь.

Глава 5. Данные: собираем и наводим порядок

Запомни. Строка – объект, столбец – признак (свойство), клетка – одно значение. Строка заголовка – не запись. Число клеток = число строк · число столбцов.

Запомни. Количественные данные можно складывать и усреднять, категориальные – только сравнивать и пересчитывать. Цифры в номере, шифре, индексе – это ярлык, а не количество. Числа из файла сначала переведи из текста в число.

Запомни. CSV: первая строка – заголовок, записей на одну меньше, чем строк; запятых в строке на одну меньше, чем столбцов. JSON: {} – карточка с ключами, [] – список, счёт в списке с нуля.

Запомни. Страниц = всего : на странице, **с округлением вверх**. На последней = всего – (страниц – 1) · на странице. В цикле `range(1, N + 1)` : последнее число в `range` не входит.

Запомни. Четыре беды: пропуск, дубликат, опечатка, выброс. Пропуск не превращай в ноль. Для компьютера «Бердск» и «бердск» – разные слова. Сначала чистка, потом подсчёт.

Запомни. Фильтр – оставить строки по условию, сортировка – расставить по столбцу, подсчёт – свернуть в число: количество, сумма, среднее, максимум, минимум, количество уникальных. «Сумма по группе» и «среднее по группе» – разные вопросы.

Глава 6. Графики и анализ данных

Запомни. Сравнить – столбики. Время – линия. Доли одного целого – круг. Две величины у каждого объекта – точки.

Запомни. Сначала подписи, потом картинка: что по осям, в каких единицах, с какого числа начинается ось. Цена деления – разница соседних подписей, делённая на число промежутков между ними.

Запомни. Медиана – середина **упорядоченного** ряда; при чётном количестве – среднее двух средних. Выброс тянет среднее и почти не трогает медиану. Сумма, среднее, максимум по группам – разные вопросы и разные ответы.

Запомни. Гипотеза – конкретная догадка, которую можно проверить таблицей. Точки полосой вверх или вниз – связь есть, кашей – не видно. Связь не доказывает, что одно вызывает другое.

Запомни. Перед сравнением столбиков посмотри, с какого числа начинается ось. Сравнивай сами числа, а не высоту картинки: во сколько раз – делением, на сколько процентов – от того значения, с которым сравниваешь.

Запомни. Дашборд – главное на одном экране: плитки с числами и несколько графиков. Эссе – вопрос, данные, график, вывод с числами и оговорки. Вывод без чисел – это мнение, а не вывод.

Глава 7. Нейросети своими руками

Запомни. Связи между соседними слоями = произведение числа нейронов в них; связи сети – сумма по всем парам соседних слоёв. Смещения есть у каждого нейрона, кроме входов. Входов – сколько признаков, выходов – сколько классов, для числа – один выход.

Запомни. Порядок всегда один: сумма «вход · вес» по всем входам, плюс смещение, потом функция активации. Считаем слой за слоем: входами выходного нейрона служат **ответы** скрытых нейронов, а не исходные x .

Запомни. Ошибка = верный ответ – ответ модели. Новый вес = вес + скорость · ошибка · вход. Эпоха – один проход по всем обучающим примерам. Большая скорость – ошибка скачет и растёт, маленькая – учимся медленно.

Запомни. Переобучение – на обучении точность растёт, а на проверке падает. Лучшая эпоха – та, где **выше всего точность на проверке**, а не на обучении. Низкая точность на обеих выборках – недообучение.

Запомни. Low-code: собрал примеры по классам → обучил → проверил на новых примерах. Классы – сопоставимо, без сильного перекоса, примеры – разные. Ответ сети – класс с наибольшей уверенностью.

Глава 8. Python, диалоги и промпты

Запомни. = кладёт значение в переменную, новое вытесняет старое. Правая часть считается целиком, а уже потом результат кладётся в коробку слева. `input` всегда отдаёт текст – для чисел оборачивай его в `int(...)`.

Запомни. После `if`, `elif`, `else`, `for`, `while` – двоеточие, а строки внутри – с отступом. Из цепочки `if` – `elif` – `else` выполняется только первая подходящая ветка. `range(1, 6)` – это 1, 2, 3, 4, 5, без шести.

Запомни. Номера в списке и в строке начинаются с 0, последний элемент – `[-1]`, а номер последнего равен длине минус один. Словарь ищет по ключу, ключи не повторяются. "Снег" и "снег" для Python – разные слова.

Запомни. Узел – реплика, стрелка – переход, ветка – путь от начала до конца. Длина ветки – число реплик в ней. Число веток = число у конца, если подписать у каждого узла сумму чисел узлов, из которых в него входят стрелки. Интент – намерение человека, которое бот узнаёт по фразе.

Запомни. Промпт = контекст и роль + цель + исходные данные + ограничения + формат. Чего не написал – нейросеть додумает сама. Любой ответ нейросети – черновик, пока ты его не проверил.

Запомни. Код от нейросети проверяют запуском на тестах, где ответ известен заранее. Падение программы – это подсказка, где искать. Самая опасная ошибка – когда программа работает и выдаёт правдоподобное неверное число.

Об авторе-составителе



Артём Осиновский

Методист и разработчик детского технопарка «Кванториум» (Новосибирск). Придумывает образовательные программы и сам делает для них цифровые инструменты: платформу смены «Миссия на Луну» для фестиваля «Технопром-2026», сайт с памятками НТО Юниоры для Новосибирской области. Эту книгу собрал для школьников области, чтобы к отбору

можно было подготовиться самостоятельно – даже без кружка рядом с домом.

Нашли ошибку – пишите на osinovsky.aa@kvantorium54.ru (рабочая почта): сфера, версия книги и страница. По другим вопросам и проектам – личная: artem081033@gmail.com.