

Съдържание

Забележка: В съдържанието уроците от притурката са маркирани с **[НОВО]**. За останалите уроци трябва да използвате стария учебник.

Преговор

Проверка на уменията и знанията (входно ниво)

ОБРАБОТКА НА ИНФОРМАЦИЯТА И КОМУНИКАЦИЯ

1. Оценяване на валидността и надеждността на информацията
2. Съвременни технологични средства за дигитално сътрудничество
- [НОВО]** 3. Изкуственият интелект и информацията 2
- [НОВО]** ТЕСТ 4

ПРОГРАМИРАНЕ И ИЗКУСТВЕН ИНТЕЛЕКТ

4. Алгоритъм
- [НОВО]** 5. Представяне на алгоритми чрез псевдокод и език за програмиране ... 5
- [НОВО]** 6. Първи стъпки с данни в Python 7
- [НОВО]** 7. Набори от данни: откриване и зареждане 9
- [НОВО]** 8. Обработка и визуализация на данни (упражнение) 11
- [НОВО]** 9. Как се обучава изкуственият интелект 13
- [НОВО]** 10. Данните около нас и работата с ИИ 15
- [НОВО]** 11. Проект: „Анализ на реален набор от данни“ 17
- [НОВО]** ТЕСТ 18

РЕШАВАНЕ НА ПРОБЛЕМИ И БЕЗОПАСНОСТ

12. Решаване на проблеми, възникнали при използване на дигитални технологии
13. Решаване на проблеми, възникнали при използване на дигитални технологии (упражнение)
- [НОВО]** 14. Чатбот с изкуствен интелект като помощник при решаване на проблеми 19
15. Използване на дигиталните технологии и въздействието им върху здравето и околната среда
- [НОВО]** ТЕСТ 21

- [НОВО]** Какво научихте в 10. клас (обобщение) 22
- [НОВО]** Какво научихте в 10. клас (тест) 23

Електронен учебник

<https://ebook.domino.bg/books/10/it>

Задачи и материали към учебника

<https://kmit.bg>

Днес голяма част от информацията, която достига до нас през интернет, минава по някакъв начин през изкуствен интелект (ИИ). Това става по различни начини:

- Директно – като използваме ИИ, за да търсим конкретни данни.
- Косвено – търсачките подреждат резултатите с алгоритми за машинно обучение, социалните мрежи решават кои публикации да ни покажат, а все по-голям дял от текстовете, изображенията и видеоклиповете онлайн са създадени или редактирани от генеративен ИИ.

Това носи много нови рискове за достоверността на информацията.

Как изкуственият интелект влияе на информацията

Влиянието на ИИ върху информацията може да се обобщи в три посоки: как информацията се създава, как се подбира и стига до нас и как може да бъде подправена.

Масово създаване на съдържание

С помощта на генеративен ИИ за секунди може да се произведе статия, коментар, изображение или цяло видео. Това означава, че обемът на съдържанието в интернет нараства много по-бързо отпреди, но не всяко съдържание е проверено и достоверно. ИИ може да напише убедителен текст по тема, по която няма надеждни данни, и резултатът да звучи правдоподобно, без да е верен.

Подбор на информацията: препоръчителни системи

Платформите за видео, музика, новини и пазаруване използват препоръчителни системи – алгоритми, които предвиждат какво ще ни хареса въз основа на предишното ни поведение. Така виждаме предимно съдържание, близко до интересите и възгледите ни. Това удобство има и обратна страна: създава се т.нар. **информационен балон** (ехо-камера), в който срещаме все по-малко различни гледни точки и лесно оставаме с впечатлението, че „всички мислят като нас“.

Задача 1. Отворете началната страница на видеоплатформа или социална мрежа, която използвате. Разгледайте първите 10 препоръчани публикации и помислете защо точно те са ви показани. Обсъдете в клас какви теми НЕ виждате и защо.

Дълбоки фалшификации (deepfake)

Дълбоката фалшификация (deepfake) е изображение, аудио или видео, в което ИИ е заменил лице, глас или действия така, че да изглеждат истински. В 8. клас видяхме, че deepfake се създава чрез генеративно-състезателна мрежа (GAN), в която генераторът прави все по-убедителни фалшификати, а дискриминаторът се опитва да ги разпознае. С развитието на технологията фалшификатите стават все по-трудни за откриване с просто око.

Признаци, по които можем да заподозрем deepfake:

- Неестествено мигане, странни сенки или осветление, които не съвпадат с фона.
- Размазани или трептящи ръбове около лицето, косата или зъбите.
- Разминаване между движението на устните и звука.
- Прекалено гладка кожа или леки изкривявания при бързо движение.
- Източникът е непознат, а съдържанието е сензационно и предизвиква силни емоции.

Важно! Тези признаци стават все по-незабележими.

Затова най-надеждната защита е да проверим източника и да потърсим потвърждение от надеждни медии.

Задача 2. Разгледайте файловете от папката **urok 3** на приложените материали. За всеки от тях запишете кои признаци ви карат да го определите като истински или съмнителен. Сравнете преценките си с тези на съучениците си.

Основни понятия

Препоръчителна система – алгоритъм, който подбира и подрежда съдържание според предполагаемите интереси на потребителя.

Информационен балон – състояние, при което потребителят среща предимно информация, която потвърждава вече съществуващите му възгледи.

Погрешната информация – невярно съдържание, разпространено без умисъл. **Дезинформация** – невярно съдържание, разпространено умишлено с цел да заблуди, манипулира или навреди.

Любопитно

Първите регулации за етиктиране на deepfake съдържание влизат в сила в Китай в началото на 2023 г. Те изискват синтетичните видеоклипове и снимки ясно да се обозначават, за да не се заблуждава обществото. Подобни правила се обсъждат и в Европейския съюз.

Погрешна информация (неумишлено невярна)

- **Намерение** – липсва умисъл за вреда или измама.
- **Мотив** – споделя се добронамерено поради неразбиране или грешка.
- **Пример** – човек споделя стара прогноза за времето, мислейки я за днешна.
- **Разпространение** – често се предава бързо, защото хората вярват, че помагат на близките си.

Дезинформация (умишлено невярна)

- **Намерение** – създадена е целенасочено с цел измама, манипулация или вреда.
- **Мотив** – финансова печалба, политическо влияние, всяване на паника или увреждане на репутация.
- **Пример** – изфабрикуван видеозапис на политик, целящ да промени изборния вот.
- **Разпространение** – използва координирани мрежи, фалшиви профили (ботове) и силни емоционални провокации.

Въпроси и задачи

1. Посочете три начина, по които изкуственият интелект влияе на информацията, която достига до нас.
2. Какво представлява дълбоката фалшификация (deepfake) и защо е опасна за достоверността на информацията?
3. Каква е разликата между погрешна информация и дезинформация? Дайте по един пример.
4. Защо плавният и убедителен текст на чатбот не е гаранция за истинност?
5. Обяснете какво е информационен балон (echo-camera). Как препоръчителните системи го създават?
6. Създайте кратък списък с правила за проверка на съмнителна публикация. Споделете го със съучениците си в Google Drive (Гугъл Диск).

Генериран текст и „халюцинации“

Големите езикови модели (като тези зад чатботовете с ИИ) предсказват коя дума е най-вероятно да следва. Затова текстът им е плавен и убедителен – но плавността не е гаранция за истинност. Когато на модела му липсват данни, той понякога „халюцинира“: уверено съчинява факти, цитати, дати или източници, които не съществуват.

Освен това много модели са обучени с данни до определен момент и не познават най-новите събития. А данните, с които са обучени, може да съдържат пристрастия (предубеждения), които моделът несъзнателно възпроизвежда.

Важно! Текст, изображение или видео, генерирани от ИИ, не са източник на истина. Те са предположение, изградено върху данни. Винаги проверявайте важните твърдения в надежден, независим източник.

Погрешна информация и дезинформация

Не всяка невярна информация е създадена със зла умисъл. Важно е да различаваме двата случая, защото те изискват различно отношение. **Погрешната информация** е невярно съдържание, разпространено без умисъл. **Дезинформация** е невярно съдържание, разпространено умишлено, с цел да заблуди, манипулира или навреди. ИИ усилва и двата проблема: позволява дезинформацията да се произвежда масово и евтино (хиляди фалшиви коментари, статии и профили), а заради убедителността ѝ добронамерените хора по-лесно я препращат, превръщайки я в погрешна информация.

Подвеждащи заглавия (clickbait)

Подвеждащите заглавия (clickbait) са заглавия, създадени да предизвикат любопитство или силна емоция, за да ни накарат да щракнем върху връзката – често към реклама, нежелано или манипулирано съдържание. Типични белези са преувеличението („Това ще ви шокира!“), скритата информация („Не можете да познаете какво стана после...“) и обещания, които текстът после не изпълнява. Днес такива заглавия също често се генерират автоматично от ИИ.

Задача 3. Потърсете в интернет три заглавия, които смятате за подвеждащи. За всяко обяснете кой похват се използва (преувеличение, скрита информация, силна емоция). Предложете как би изглеждало честно, информативно заглавие на същата тема.

Как да се защитим: проверка на източника

Срещу подправена и невярна информация най-силното оръжие остава критичното мислене. Ето практични правила, които вече започнахте да прилагате в Урок 1 и сега разширяваме при използване на ИИ:

- **Открийте източника.** Кой публикува информацията? Има ли име на автор, дата и контакти? Анонимното и недатираното съдържание е по-съмнително.
- **Проверете в няколко независими източника.** Ако само едно място твърди нещо сензационно, бъдете предпазливи. Надеждните новини обикновено се потвърждават от няколко утвърдени медии.
- **Внимавайте със силните емоции.** Манипулативното съдържание е създадено да предизвика гняв, страх или възторг. Спрете и проверете, преди да споделите.
- **Помислете дали съдържанието може да е генерирано от ИИ.** Особено при изображения и видео с известни хора търсете официално потвърждение.
- **Не приемайте отговора на чатбот за окончателен.** Използвайте ИИ, за да се ориентирате, но проверявайте фактите самостоятелно.

Запомнете: дори дадена тема да е широко отразена и да я срещате на много места, това не означава автоматично, че е вярна – възможно е едно невярно твърдение да е било многократно преписано.

Задача 4. Изберете актуална новина, която ви е направила впечатление. Проследете я до първоизточника. Проверете я в поне два други независими източника и запишете: кой е първоизточникът, кога е публикувана новината и потвърждава ли се. Направете извод доколко ѝ имате доверие.

1. Заявката за търсене се нарича:

- А) библиотека, в която се търси по определена тема
- Б) специална програма, която събира информация от всички достъпни страници в интернет, обработва я и я съхранява в огромни бази от данни
- В) ключова дума, фраза или комбинация от думи, която се задава в полето за търсене
- Г) програма, предоставяща възможност за търсене едновременно чрез няколко търсачки

2. Дълбоката фалшификация (deepfake) е:

- А) нежелана рекламна поща
- Б) реалистично фалшиво изображение или видео, създадено с ИИ
- В) онлайн семинар
- Г) вид облачно хранилище

3. Легитимното използване на ресурси означава, че:

- А) оторизираните потребители имат непрекъснат достъп до информацията и ресурсите
- Б) получателят контролира дадена информация, за да се избегне нейното разпространение
- В) информацията и програмите се променят само по определен и разрешен начин
- Г) ресурсите не могат да се използват от неоторизирани потребители или по неразрешен начин

4. Загуба на целостта на данните означава, че:

- А) информацията става достъпна за неоторизирани потребители
- Б) нарушител е добавил, променил или изтрил информация
- В) услугите не могат да се използват пълноценно
- Г) дадена услуга не може да се изпълнява

5. Ако информацията става достъпна за неоторизирани потребители, това води до загуба на:

- А) контрол
- Б) конфиденциалност на данните
- В) услуги
- Г) целостта на данните

6. Спамът е:

- А) получена нежелана от потребителя информация
- Б) информация, която става достъпна за неоторизирани потребители
- В) услуга, която не може да се изпълнява
- Г) криптиране на изпращаните и декриптиране на получаваните съобщения

7. Заглавие, създадено да предизвика любопитство и да ни накара да щракнем към реклама или манипулирано съдържание, се нарича:

- А) deepfake
- Б) clickbait
- В) спам
- Г) уебинар

8. „Халюцинация“ при чатбот с ИИ означава, че:

- А) моделът убедено твърди нещо невярно
- Б) моделът се изключва от претоварване
- В) връзката с интернет прекъсва
- Г) потребителят е въвел грешна парола

9. Невярна информация, споделена БЕЗ намерение да се заблуждава, се нарича:

- А) дезинформация
- Б) погрешна информация
- В) спам
- Г) фишинг

10. Невярна информация, създадена и разпространявана НАРОЧНО, за да заблуди, се нарича:

- А) погрешна информация
- Б) спам
- В) дезинформация
- Г) фишинг

Да си припомним основните операции на скриптовия език за програмиране – Python.

Какво прави	Запис на Python
Въвеждане и извеждане на данни	<code>input()</code> / <code>print()</code>
Присвояване на стойност	<code>x = a * b</code>
Проверка на условие	<code>if ... :</code> / <code>elif</code> / <code>else</code>
Повторение определен брой пъти	<code>for i in range(...)</code>

Задача 1. Създайте линеен алгоритъм за намиране лице на правоъгълник. Представете алгоритъма чрез псевдокод и с език за програмиране – Python.

Алгоритъм, описан с псевдокод:

Въведи a;
 Въведи b;
 Пресметни $S = a * b$;
 Изведи S;

Алгоритъм, описан с Python:

```
a = float(input("Дължина a: "))
b = float(input("Широчина b: "))
S = a * b
print("Лице на правоъгълника:", S)
```

Задача 2. Създайте алгоритъм за намиране на най-голямото от три числа, като използвате за a, b, c съответно стойностите 9, 7, 11. Представете алгоритъма чрез псевдокод и с език за програмиране Python.

Алгоритъм, описан с псевдокод:

Въведи a = 9;
 Въведи b = 7;
 Въведи c = 11
 m = a;
 Ако b > m тогава m = b;
 Ако c > m тогава m = c;
 Изведи m;

Алгоритъм, описан с Python:

```
a = 9
b = 7
c = 11
m = a # приемаме, че a е най-голямото
if b > m:
    m = b
if c > m:
    m = c
print("Най-голямото е:", m)
```

Важно!

В Python отстъпът (обикновено 4 интервала) не е само за прегледност, а е част от програмата. Редовете с еднакъв отстъп образуват един блок. Грешен отстъп води до грешка или до различно поведение.

Задача 3. Създайте алгоритъм за изчисляване сумата на първите n естествени числа. Представете алгоритъма чрез псевдокод и с език за програмиране Python.

Алгоритъм, описан с псевдокод:

```
Въведи n
s = 0
i = 1
за i от 1 до n повтаряй:
    s = s + i
    i = i + 1
Изведи s
```

Алгоритъм, описан с Python:

```
n = int(input("Въведете n: "))
S = 0
for i in range(1, n + 1):
    S = S + i
print(S)
```

Проследяване на код (трасиране)

Важно умение е да предвидим резултата на кратък код, без да го изпълняваме.

Задача 4. Разгледайте дадения алгоритъм. Какъв ще бъде резултатът от изпълнението на кода?

```
x = 3
y = 4
if x + y > 6:
    print("Голямо")
else:
    print("Малко")
```

Тъй като $x + y$ е 7, а $7 > 6$ е вярно, програмата ще изведе **Голямо**.

Задача 5. Дадена ви е програмата за индекс на телесна маса (BMI) по зададени височина $height$ (в см) и тегло $weight$ (в кг). Стартирайте я в Python за височина 170 см и тегло 65 кг и проверете каква категория извежда.

```
height = float(input("Височина (см): "))
weight = float(input("Тегло (кг): "))
bmi = weight / (height/100)**2
if bmi < 18.5:
    print("Поднормено тегло")
elif bmi <= 25.0:
    print("Нормално тегло")
elif bmi <= 30.0:
    print("Наднормено тегло")
else:
    print("Затлъстяване")
```

Въпроси и задачи

1. На кой елемент от Python съответства проверката на условие?
2. Защо отстъпът е важен в програмата на Python?
3. Какъв ще бъде резултатът от изпълнението на кода от Задача 4, ако $x = 10$ и $y = 1$?
4. Създайте код за намиране корените на квадратно уравнение.
5. Представете алгоритъма от Задача 5 чрез псевдокод.

Основни понятия

Списък (list) – структура от данни, която съхранява няколко стойности в определен ред под едно име.

Индекс – поредният номер на елемент в списъка, като броенето започва от 0.

Списък – набор от стойности под едно име

Досега работихме с отделни числа. Но истинската сила на програмирането проличава, когато обработваме много стойности наведнъж – например всички оценки на ученик или температурите за един месец. За това в Python използваме списък.

Списъкът съхранява няколко стойности, подредени една след друга. Записва се в квадратни скоби, а стойностите се разделят със запетая. Достъп до елемент от списъка имаме чрез неговия индекс (пореден номер):

```
ocenki = [5, 6, 4, 5, 3, 6]
print(ocenki)
print("Първата оценка е:", ocenki[0])
```

Забележка: Елементите се броят от 0: `ocenki[0]` е първият елемент, `ocenki[1]` – вторият, и т.н.

Агрегиращи функции

Често не се интересуваме от всяка стойност поотделно, а от обобщен резултат – най-голямата (`max`), най-малката (`min`), средната стойност (`average`) или сумата на числата (`sum`). За това Python има готови вградени функции, наречени агрегиращи (обобщаващи).

Задача 1. Създайте алгоритъм за пресмятане на средната стойност като сума, разделена на броя числа. Разгледайте таблицата с функциите. Използвайте тези, които ви трябват, на мястото на точките?

```
ocenki = [5, 6, 4, 5, 3, 6]

print("Брой оценки:", len(ocenki))
print("Най-висока:", max(ocenki))
print("Най-ниска:", min(ocenki))
print("Сбор:", sum(ocenki))

sreden = .....
print("Среден успех:", sreden)
```

Функция	Какво връща
<code>len()</code>	броя на елементите
<code>max()</code>	най-голямата стойност
<code>min()</code>	най-малката стойност
<code>sum()</code>	сбора на всички стойности
<code>sorted()</code>	нов, подреден списък

Попълване на списък от въведени данни

В един списък можем да добавяме или да премахваме стойности. За добавяне използваме метода `append` (добави). Ето как можем да организираме добавянето на n стойности към един списък от потребителя:

```
ocenki = []
n = int(input("Колко оценки?"))
for i in range(n):
    x = float(input("Въведете оценка: "))
    ochenki.append(x)

print("Среден успех:", sum(ocenki) / len(ocenki))
```

Тъй като в Python няма готова функция за средно аритметично, разделяме сумата на оценките (`sum`) на техния общ брой (`len`).

Забележка: Методът `ocenki.append(x)` добавя стойността на променливата `x` в края на списъка `ocenki`.

Задача 2. Въведете програмата по-горе и я изпробвайте с оценките си по няколко предмета. Разширете я така, че освен средния успех да извежда и най-високата и най-ниската оценка.

Задача 3. Създайте списък с дневните температури за една седмица. Изведете средната седмична температура и колко градуса е разликата между най-топлия и най-студения ден (използвайте функциите `max` и `min`).

Задача 4. В час по физическо възпитание учениците бягат 60 метра. Напишете програма, която пита колко ученици са бягали, след което извежда времето на всеки от тях в секунди и го добавя в списък с `append`.

Програмата да извежда: всички времена, подредени от най-доброто към най-слабото (използвайте `sorted()`), броя на бягалите ученици, най-доброто и най-слабото време и средното време за класа.

Задача 5. Бизнесмен проследява дневните приходи на своя онлайн магазин за една работна седмица (от понеделник до петък). Напишете програма, която изисква от потребителя да въведе последователно дневните приходи (дробни числа) за петте дни и ги запазва в списък `prihodi`.

Програмата да извежда: Общия оборот за седмицата, Среднодневния приход, Максималния приход за един ден, Минималния приход за един ден.

Въпроси и задачи

1. Какво е списък и с какво е по-удобен от много отделни променливи?
2. Кой индекс има първият елемент на списък в Python?
3. Как се пресмята средна стойност чрез вградени функции?
4. Какво прави методът `append()`?

Любопитно

Данните се наричат „новият нефт“ на XXI век. Те захранват експертните системи, търсачките и изкуствения интелект. Колкото повече и по-качествени са данните, толкова по-добре работят системите, обучени с тях. Затова умението да намираме и да обработваме данни става все по-ценно.

Основни понятия

Набор от данни (dataset) – организирана съвкупност от данни, обикновено подредени в таблица с редове (записи) и колони (признаци).

Набори от данни

Малките списъци, които въвеждаме на ръка, са удобни за упражнения. В реалния свят обаче се използват големи количества данни, съхранени в структури – наречени набори от данни (dataset). Тяхната обработка е сред основните задачи на съвременните информационни технологии и е в основата на изкуствения интелект.

Откъде да вземем набори от данни?

Съществуват публични хранилища, които предоставят данни безплатно за свободно използване (т.нар. отворени данни):

- Портал за отворени данни на egov.bg – официален сайт с данни за България (население, образование, транспорт и др.).
- Eurostat – статистическа служба на Европейския съюз с данни за всички държави членки.
- Kaggle – световна платформа с хиляди набори от данни по най-различни теми.

Форматът CSV

Най-разпространеният прост формат за таблични данни е CSV (Comma-Separated Values – стойности, разделени със запетая). Това е обикновен текстов файл, в който всеки ред е ред от таблица (запис), а стойностите от колоните са разделени със запетая. Първият ред обикновено съдържа имената на колоните (заглавен ред).

Пример: Файл *temperaturi.csv* – пет записа с по три стойности (месец, минимална и максимална температура).

```
месец,мин. температура,макс. температура
Януари, -2,12
Февруари, -4,8
Март, -3,14
Април, 2,18
Май, 5,28
```

Задача 1. Разгледайте примерния файл *temperaturi.csv*. Потърсете в интернет източник на данни за месечните температури във вашия град. Свалете ги като csv файл или ги заменете в съществуващия файл.

Зареждане на CSV с модула csv

За четене на CSV ще използваме вградения модул csv. Той е част от Python и не се инсталира допълнително. Първо просто прочитаме и извеждаме всеки ред:

```
import csv

with open("temperaturi.csv", encoding="utf-8") as f:
    reader = csv.reader(f)
    for red in reader:
        print(red)
```

Забележка: `open()` отваря файла; `encoding="utf-8"` осигурява правилно четене на кирилицата.

По-полезно е да вземем само стойностите от една колона и да ги сложим в списък, за да ги обработим с вече познатите агрегиращи функции:

```
import csv

temperaturi = []
with open("temperaturi.csv", encoding="utf-8") as f:
    reader = csv.reader(f)
    next(reader)          # пропускаме заглавния ред
    for red in reader:
        temperaturi.append(float(red[1]))

print("Средна температура:", sum(temperaturi) /
len(temperaturi))
print("Най-висока:", max(temperaturi))
```

Забележка: `red[1]` е втората колона (температурата); `next(reader)` прескача реда със заглавията.

Важно!

Файлът с данни трябва да е в същата папка като програмата, иначе Python няма да го намери. При зареждане на чужди данни винаги проверявайте първите няколко реда, за да разберете структурата (кои са колоните и в какъв ред са).

Задача 2. Посетете Портала за отворени данни egov.bg. Намерете набор от данни по тема, която ви интересува, и разгледайте от какви колони се състои. Запишете: как се казва наборът, колко колони има и какво описва всяка от тях.

Задача 3. Създайте текстов файл `temperaturi.csv` с данните от примера. Въведете втората програма и проверете каква средна температура извежда. Добавете ред, който извежда и най-ниската температура от колоната.

Задача 4. Допълнете файла `temperaturi.csv` със записи за останалите месеци до края на годината (потърсете приблизителни данни за вашия град в интернет). Напишете програмата, която:

- зарежда в един списък минималните, а в друг – максималните температури;
- извежда средната минимална и средната максимална температура за годината;
- изчислява за всеки месец температурната амплитуда (разликата между максималната и минималната температура) и извежда най-голямата от тях.

Задача 5. Създайте файл `ocenki.csv` (или използвайте приложения файл), в който всеки ред съдържа учебен предмет и вашата годишна оценка по него. Първият ред е заглавен:

```
предмет, оценка
Математика, 5
Информационни технологии, 6
Английски език, 5
```

Напишете програма, която зарежда оценките от втората колона в списък и извежда: броя на предметите, средния успех, най-високата и най-ниската оценка. Допълнете програмата така, че да извежда съобщение „Отличен успех!“, когато средният успех е над 5.50.

Въпроси и задачи

1. Какво е набор от данни (dataset)? Дайте пример.
2. Посочете три публични хранилища за отворени данни.
3. Как е устроен файл във формат CSV?
4. Защо в програмата прескачаме заглавния ред с `next(reader)`?

Обработка и визуализация на данни (Упражнение)

Подбор на данни: филтриране

Често ни трябва само част от данните – например само дните, в които температурата е над определена стойност. Това се нарича филтриране и може да се направи с условие в цикъл:

Задача 1. Създайте алгоритъм за извеждане броя на дните, в които температурата е със стойност над 20 градуса. Разгледайте файлът "temperatures_july.csv". Колко са общите часове с температура над 20 градуса за дните, за които има данни? По-долу е даден елементарен примерен алгоритъм. За да решите задачата, променете алгоритъма, като използвате програмите от предишните уроци.

```
temperaturi = [2, 4, 9, 14, 19, 23, 27]

goreshti = []
for t in temperaturi:
    if t > 20:
        goreshti.append(t)

print("Дни над 20 градуса:", len(goreshti))
print("Стойности:", goreshti)
```

Забележка: В списъка goreshti попадат само стойностите, които изпълняват условието $t > 20$.

Задача 2. Променете кода така, че да показва броя на дните със средна дневна температура над 25 градуса

Подреждане: сортиране

Функцията `sorted()` връща нов, подреден списък. Така лесно намираме крайните стойности и подреждаме данните.

Задача 3. Създайте алгоритъм за подреждане на стойностите на дните и за намиране стойността на най-студения и най-топлия ден.

```
temperaturi = [12, 14, 9, 14, 5, 19, 23, 19, 27]

podredeni = sorted(temperaturi)
print("Подредени:", podredeni)
print("Най-студено:", podredeni[0])
print("Най-топло:", podredeni[-1])
```

Забележка: индекс `-1` означава последния елемент – удобен начин да вземем най-голямата стойност след сортиране.

Визуализация: графика с модула matplotlib

Числата са по-разбираеми, когато ги изобразим графично. За тази цел ще използваме модула matplotlib. Той не е вграден – инсталира се веднъж, както модулите в 9. клас (от менюто на средата или с команда в конзолата `pip install matplotlib`).

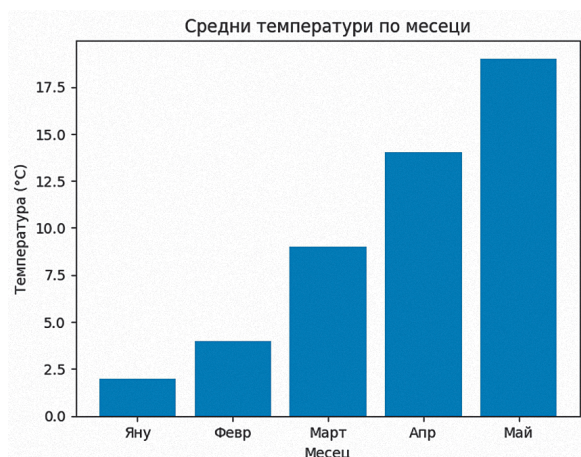
Задача 4. Създайте алгоритъм за изчертаване на стълбовидна диаграма, начертана с модула matplotlib.

```
import matplotlib.pyplot as plt

mesetsi = ["Яну", "Февр", "Март", "Апр", "Май"]
temperaturi = [2, 4, 9, 14, 19]

plt.bar(mesetsi, temperaturi)
plt.title("Средни температури по месеци")
plt.xlabel("Месец")
plt.ylabel("Температура (°C)")
plt.show()
```

Забележка: `plt.bar()` чертае стълбовидна графика; `plt.show()` отваря прозорец с готовата диаграма.



Стълбовидна графика, начертана с matplotlib

Да си припомним

В 9. клас инсталирахме модули като `pyjokes`, `translate` и `qrcode` през менюто на средата или с `pip`. Matplotlib се добавя по същия начин.

Ако при `import` се появи грешка, че модулът липсва – първо го инсталирайте.

Задача 5. Заредете набор от данни от папката `uogk 8` на приложените материали.

- Изведете средната, най-високата и най-ниската стойност на една колона.
- Филтрирайте записите по избран от вас критерий и пребройте колко са.
- Начертайте стълбовидна графика на данните. Запазете графиката като изображение.

Въпроси и задачи

1. Какво означава „филтриране“ на данни? Как се прави с код?
2. Какво връща функцията `sorted()` и как се взема най-голямата стойност след сортиране?
3. Кой модул използвахме за графики и как се добавя към проекта?

Да си припомним

- Машинното обучение позволява на системите да се „учат“ от примери, вместо да бъдат програмирани изрично за всеки случай.
- Невронните мрежи се вдъхновяват от устройството на човешкия мозък и категоризират големи обеми данни – изображения, звуци, текст.
- За да разпознава котка, обучаваме системата с хиляди снимки на котки (напр. играта Quick Draw).

В предишните уроци се научихме да обработваме набори от данни. Сега ще видим за какво още служат тези данни: те са „горивото“, с което се обучава изкуственият интелект. В 8. клас разбрахме, че данните са нужни за машинното обучение и невронните мрежи. Тук ще задълбочим как точно данните превръщат една система в „умна“ и как проверяваме дали тя наистина се е научила.

Машинно самообучение: учене от данни

При традиционното програмиране ние даваме на компютъра данни и точни правила. Компютърът пресмята и дава отговор.

При машинното самообучение постъпваме обратно: даваме на системата данни и много примери с верни отговори и я оставяме сама да открие правилата (закономерностите) в тях. Резултатът от това обучение се нарича модел – програма, която после може да извършва действията, за които е обучена, с нови непознати данни.

Основни типове машинно обучение:

• **Обучение с учител** – компютърът се обучава с предварително етикетирани данни (пример: показваме му 1000 снимки, на които изрично е написано "това е котка" и "това е куче"). Приложение: филтри за спам в имейла, разпознаване на лица.

• **Обучение без учител** – компютърът получава сурови данни без етикети и сам трябва да открие скрити модели, прилики или групи в тях. Приложение: препоръчване на филми във видеоплатформи или продукти в онлайн магазини въз основа на сходни интереси.

• **Обучение чрез подсилване** – роботът се учи чрез проба и грешка в дадена среда, като получава "награди" за правилни действия и "наказания" за грешни. Приложение: автономни автомобили, компютърни игри и шахматни програми.

Тренировъчни и тестови данни

Един от най-важните принципи в машинното обучение е, че данните се разделят на две части. С тренировъчния набор системата се обучава. С тестовия набор данни, които моделът не е виждал по време на обучение – проверяваме доколко добре се е научил. Наборът от данни се разделя – с по-голямата част моделът се обучава, а с по-малката се проверява.

Задача 1. Защо е нужно разделянето на данните при машинното обучение?



Набор от данни

Тренировъчните данни (training data) са в основата на машинното обучение. Това е огромният набор от информация (текст, снимки, видео, аудио), който се подава на един алгоритъм, за да може той да се научи да разпознава модели, да прави връзки и да взема решения. Ако сравним изкуствения интелект с ученик, тренировъчните данни са неговите учебници и помагала. Процесът на обучение преминава през три основни стъпки:

- Подаване на данни – алгоритъмът анализира примерите.
- Търсене на зависимости – ИИ открива скрити правила.
- Създаване на модел – резултатът от това учене е готов математически модел, който може да прилага наученото върху съвсем нови, непознати данни.

Тренировъчните данни могат да бъдат маркирани (използват се при обучение с учител) и немаркирани (използват се при обучение без учител).

Тестовите данни (test data) представляват самостоятелен набор от информация, който се използва като финален изпит за вече обучен модел. Те се подават на алгоритъма едва след като неговото обучение е напълно приключило, за да се провери реалната му точност.

Основната им цел е да покажат как изкуственият интелект ще се справя в реалния свят, когато се сблъска с данни, които никога преди не е виждал. Тестовите данни никога не трябва да участват в процеса на обучение. Ако моделът е имал достъп до тях по време на обучението, той просто ще ги запамети (наизусти). Това води до измамно високи резултати и се нарича „изтичане на данни“.

Задача 2. Защо тестовите данни се използват задължително при машинното обучение?

Тестовите данни дават реална оценка на точността – разкриват дали моделът наистина е разбрал логиката и зависимостите, или просто е „запомнил“ тренировъчния комплект. Чрез тях се показва дали моделът е готов за внедряване в реални системи (производство) без риск от грешки.

Предизвикателства пред данните

Използването на голям обем от данни при обучението на ИИ крие и рискове, които се изразяват в:

- **Качество срещу количество** – ИИ се нуждае от огромни обеми данни, но ако те са грешни или неточни, моделът също ще работи лошо. Затова голяма част от времето при разработване на системи с машинно обучение се отделя не за самото обучение, а за събиране, почистване, проверка и подготовка на данните.

- **Преобучаване (overfitting)** – проблем, при който моделът се е приспособил твърде точно към конкретните примери, вместо да е уловил общата закономерност. Това е като ученик, който, вместо да разбере материала, научава наизуст всички задачи от учебника. На изпита обаче получава нови задачи и не може да ги реши.

Причините за възникването на проблема могат да бъдат:

- твърде сложен модел;
- недостатъчно обучаващи данни;
- прекалено дълго обучение;
- наличие на шум или грешки в данните.

Затова качеството и разнообразието на данните са решаващи – моделът е толкова добър, колкото са данните, с които е обучен. Целта на един ИИ модел не е да запомни примери, а да обобщава наученото и да го прилага успешно в нови ситуации.

- **Пристрастност** – ако тренировъчните данни съдържат човешки предразсъдъци или са едностранчиви, ИИ ще ги копира и ще взема несправедливи или неточни решения. Основният парадокс е, че ИИ не е пристрастен, а хората са. Важно е да се подчертае, че ИИ не „измисля“ тази пристрастност сам. В много случаи той я научава от данните, с които е обучен. Пристрастността означава, че качеството и представителността на обучаващите данни влияят пряко върху качеството и справедливостта на решенията, които моделът взема.

Важно! Изкуственият интелект не „мисли“ като човек. Той открива закономерности в данните, с които е обучен. Ако данните са непълни или пристрастни, такъв ще бъде и моделът. Затова към отговорите на ИИ винаги подхождаме критично.

Задача 3. Отворете средство за обучение на модел в браузъра (напр. Teachable Machine на Google или Quick Draw). „Обучете“ прост модел с няколко примера от две категории (например изображения на чаша и на молив). После го „тествайте“ с нови примери. Запишете в кои случаи греша и предположете защо.

Задача 4. Обсъдете в клас: ако обучим модел да разпознава кучета само със снимки на едри кучета, как ще се справи той със снимка на малко куче? Какъв тип проблем илюстрира това?

Основни понятия

Машинно самообучение – процес, при който система открива закономерности в данни и подобрява действието си с натрупването на повече примери.

Модел – резултатът от обучението; програма, която прави предсказания за нови данни.

Тренировъчни данни – набор от информация, който се използва, за да може алгоритъм да се научи да разпознава модели, да прави връзки и да взема решения.

Тестови данни – самостоятелен набор от информация, който се използва като финален изпит за вече обучен модел.

Въпроси и задачи

1. Каква е разликата между традиционното програмиране и машинното самообучение?
2. Защо разделяме данните на тренировъчни и тестови?
3. Какво е преобучаване (overfitting)? Дайте пример с думи.
4. Защо качеството на данните е толкова важно за изкуствения интелект?



Процес на използване на данните от ИИ

Любопитно

Системата CAPTCHA (напр. „изберете всички изображения със светофари“) пази сайтовете от автоматизирани ботове. Но докато я решаваме, ние едновременно помагаме да се обучават ИИ модели за разпознаване на образи – отговорите на милиони хора се превръщат в безплатни тренировъчни данни.

Ключови предизвикателства при писането на заявки:

- Неяснотата – Когато заявката е твърде обща, ИИ не знае какъв точно отговор се очаква.
- Липса на контекст – ИИ не знае предварително за какво ще бъде използван отговорът.
- Липса на конкретика – ако не посочите желаната форма на отговора, резултатът може да не отговаря на очакванията.
- Сложни заявки – когато една заявка съдържа много различни задачи едновременно, ИИ може да пропусне част от тях.
- Халюцинации – Ако структурата на заявката ви е подвеждаща, моделът може да си измисли напълно фалшиви факти.

Откъде идват огромните количества данни, с които се обучава изкуственият интелект? Голяма част от тях създаваме самите ние – всеки ден, често без да се замисляме. В този урок ще видим как се раждат данните, как да задаваме добри заявки към средства с ИИ и защо трябва да внимаваме с отговорите им.

Сензорите като източник на данни

Сензорите са устройства, които измерват физични величини от околната среда и ги преобразуват в цифрови данни, които могат да бъдат обработвани от компютри и системи с ИИ. Те са „сетивата“ на ИИ. Ако алгоритмите за машинно обучение са мозъкът, сензорите са очите, ушите и нервната система, които му позволяват да усеща, да измерва и да взаимодейства с реалния физически свят. Благодарение на тях ИИ преминава от работа с абстрактни дигитални файлове към управление на физически процеси в реално време.

Пример: Камера → изображение → ИИ разпознава пешеходец → автомобилът спира.

Основни видове сензори и техните ИИ приложения

- Камери и оптични сензори – събират изображения и видео. Използват се за разпознаване на лица, откриване на предмети, автономни автомобили, медицинска диагностика.
- Микрофони и акустични сензори – улавят звукови вълни и честоти. Използват се за гласови асистенти, разпознаване на реч, превод в реално време.
- Индустриални и екологични сензори – измерват температура, влажност, налягане, газови емисии, вибрации. Използват се в интелигентни домове, индустриални системи, метеорологични станции, умно земеделие.
- Биометрични сензори – измерват пулс, нива на кислород в кръв-

та, ЕКГ, мозъчна активност. Използват се в смартчасовници и друга преносима електроника.

- GPS приемници – определят местоположението. Използват се при навигация, проследяване на превозни средства, автономни роботи.

- Лазерни сензори (LiDAR) и радари – използват лазерни лъчи или радиовълни за откриване на обекти и измерване на тяхното разстояние и скорост. Използват се в автономни автомобили, роботи, картографиране.

- Акселерометри и жирокопи – измерват ускорение, наклон и ориентация в пространството. Използват се в смартфони, дронове, роботи, фитнес тракери.

Данните от сензорите имат уникални характеристики – те не спират, генерират постоянен поток от информация в реално време, което изисква огромна изчислителна мощ за обработка на момента. Всяко използване на приложение, всяко търсене и всяко харесване в социалните мрежи също са данни. Една от най-важните техники в съвременния ИИ е сензорният синтез – комбиниране на информация от няколко различни сензора едновременно.

Използването на сензорите като източник на данни има и своите предизвикателства – неточни измервания; понякога данните съдържат случайни смущения; голям обем данни; повреда на сензор.

Заявки към средства с изкуствен интелект

Заявките към средствата с изкуствен интелект са инструкциите, въпросите или текстовите описания, които подаваме на даден ИИ модел, за да получим желаната реакция или резултат.

Основни видове заявки:

- Информационни – целта е да се получи информация или обяснение по дадена тема. Пример: Какво е машинно самообучение?
- Трансформационни (обобщение, редакция, превод, анализ) – работа върху вече съществуващ текст или данни. Пример: Обобщи този текст в пет изречения.
- Генеративни – създаване на но-

во съдържание. Пример: Създай покана за училищно събитие.

- Креативни – решаване на проблем и планиране на дейности. Пример: Създай план за училищен проект.

- Заявки за изображения – генерира изображения по текстово описание. Пример: Създай схема на компютърна мрежа.

В 8. клас вече се запознахме със заявки към генеративен ИИ. Тук ще затвърдим как да формулираме добра заявка, за да получим полезен отговор. Качеството на отговора зависи пряко от качеството на заявката.

Добрата заявка обикновено спазва определени **правила**:

- Яснота – формулирайте недвусмислено какво искате.
- Специфичност – за кого или за какво е предназначен отговорът (тема, обем, аудитория).
- Ограничения – поставете ясни рамки (дължина, език, стил, ниво на сложност).
- Желан формат – посочете как да изглежда отговорът и за кого е предназначен резултатът (списък, таблица, кратко резюме за ученици от девети клас).

Пример: Слаба заявка: „Разкажи за изменението на климата.“

Добра заявка: „Обясни в 5 изречения, подходящи за ученик от девети клас, какви са основните причини за изменението на климата.“

Умението да се формулират ясни и точни заявки е важна част от работата с ИИ. Колкото по-добре е описана задачата, толкова по-полезен и точен обикновено ще бъде полученият резултат. Това умение е известно и като **prompt engineering** (проектиране или формулиране на ефективни заявки).

Искусственият интелект като „черна кутия“

Концепцията за изкуствения интелект като „черна кутия“ описва ситуация, при която виждаме какво влиза в ИИ модела (входни данни) и какво излиза от него (резултат), но не можем да разберем точния математически процес, по който е взето решението. Алгоритмите с ИИ работят по непознат за крайния потребител начин.

Съвременните модели, особено дълбоките невронни мрежи, могат да съдържат милиарди параметри. Всеки параметър участва в изчисленията и влияе върху крайното решение. Взаимодействията между тях са толкова многобройни, че човек трудно може да проследи как точно моделът е стигнал до даден извод. Дори създателите им невинаги могат да обяснят всяко конкретно решение на модела. Това не означава, че моделът греша, а че логиката на неговите вътрешни изчисления не е прозрачна.

Въпрос: Защо това е проблем?

- Липса на обяснение – потребителят може да не разбере защо ИИ е взел определено решение.
- Липса на отчетност – ако ИИ сгреша при диагностициране на тумор в болница, лекарите не могат да обяснят защо се е стигнало до грешката..
- Скрита пристрастност – моделът може да дискриминира хора въз основа на раса, пол или възраст, без разработчиците да го забележат.
- Халюцинации – уверено изречени неистини.
- По-трудно откриване на грешки – ако моделът сгреша, може да е трудно да се установи причината.

Затова към отговора на ИИ се отнасяме като към предложение, което проверяваме, а не като към окончателна истина.

За да се справят с този проблем, учените развиват направлението обясним изкуствен интелект (Explainable AI, XAI). Неговата цел е да направи решенията на ИИ по-разбираеми. Анализира се крайният резултат и се показва кои входни данни са имали най-голяма тежест за решението

Важно! Не въвеждайте лична и чувствителна информация (пароли, адреси, данни на близки) в чатботове и онлайн ИИ услуги. Веднъж въведени, тези данни може да бъдат запазени и използвани по начини, които не контролирате.

Задача 1. Изберете тема и задайте на чатбот с ИИ една и съща задача по три различни начина: съвсем кратко и общо; с повече подробности; със зададен формат на отговора. Сравнете трите отговора по точност и полезност.

Задача 2. Помолете чатбот да опише събитие или личност, които познавате добре. Проверете отговора в надежден източник и отбележете дали има неточности или „халюцинации“.

Основни понятия

Сензор – устройство, което измерва физична величина (напр. светлина, звук, температура, местоположение) и я превръща в данни.

Любопитно

Една от най-активните области на съвременните изследвания е намирането на баланс между точността и обяснимостта. Много сложните модели често постигат най-висока точност, но са по-трудни за тълкуване. От друга страна, по-простите модели са по-лесни за разбиране, но понякога са по-малко точни. Разработването на системи, които съчетават и двете качества, е една от основните цели на бъдещото развитие на ИИ.



Въпроси и задачи

1. Посочете три сензора в смартфон и автономен автомобил. Какви данни събира всеки от тях.
2. Кои правила правят една заявка към ИИ добра?
3. Защо казваме, че ИИ работи като „черна кутия“?
4. Защо не бива да въвеждаме лична информация в онлайн ИИ услуги?

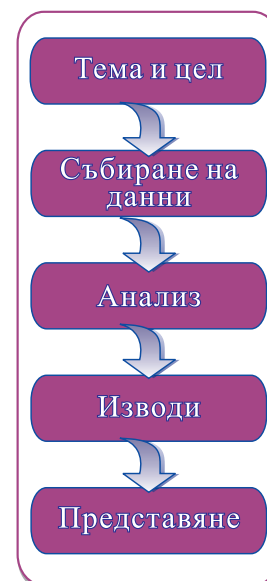
Проект: „Анализ на реален набор от данни“

Досега разгледахме как се търси набор от данни, как данните се обработват и как се представят графично с код. Това често се използва в образованието, науката и бизнеса при работа по проекти. Да си припомним, че проектът е организирана работа за постигане на определена цел чрез изпълнение на последователни задачи в определен срок.

Стъпки на проекта

За да реализирате даден проект свързан, с анализ на набор от данни, трябва да следвате определени стъпки:

1. Определяне на темата и целта.
 - Поставете въпрос – изберете ясен въпрос, на който данните могат да отговорят. Например: „През кой месец е най-топло в моя град?“ или „Кои области в България имат най-много населени места?“.
2. Събиране на данни – откъде ще получим информацията?
 - Изтеглете подходящ набор от публично хранилище (egov.bg, Eurostat, Kaggle) във формат csv.
3. Анализ на данните.
 - Обработете данните – заредете файла с Python, изчистете ненужното, приложете агрегиращи функции, филтрирайте и сортирайте.
 - Визуализирайте – начертайте подходяща графика с matplotlib.
4. Изводи – какво показват резултатите?
 - Формулирайте отговора на въпроса с 2 – 3 изречения.
5. Представяне на проекта.
 - Представете проекта пред класа чрез презентация, доклад, постер или кратко устно изложение.



Съвет

Може да използвате чатбот с ИИ като помощник – например да обясни съобщение за грешка или да предложи как да начертаете графиката. Но винаги проверявайте и разбирайте кода, който получавате – вие отговаряте за резултата, не моделът.

Формула на проекта

За да е постижим, един ученически проект може да използва формулата „1 + 1 + 1 + 1“:

- 1 цел – ясно формулирана (Какво искаме да разберем или да постигнем?).
- 1 набор от данни – от публично хранилище (С какви данни ще работим?).
- 1 анализ – който онагледява отговора (Какво ще анализираме в тези данни?).
- 1 извод – кратко обобщение (До какво заключение стигнахме?).

Проектна задача:

Изберете тема, която ви интересува (спорт, климат, население, образование). Преминете петте стъпки и подгответе кратко представяне (до 3 минути): покажете въпроса, графиката и извода. Може да работите самостоятелно или по двойки.

1. На кой елемент от Python съответства логическият блок (ромб) от блок-схемата?

- A) цикъл for Б) условие if ... else
В) функцията print() Г) списък

2. Какъв ще бъде резултатът от изпълнението на кода:

```
S = 0
for i in range(1, 4):
    S = S + i
print(S) ?
```

- A) 3 Б) 6 В) 4 Г) 10

3. Какъв е индексът на първия елемент на списък в Python?

- A) 1 Б) 0 В) -1 Г) зависи от дължината

4. Коя функция връща броя на елементите в списък?

- A) sum() Б) max() В) len() Г) sorted()

5. Как се пресмята средната стойност на списък оценки?

- A) `max(ocenki) / 2` Б) `sum(ocenki) / len(ocenki)`
B) `len(ocenki) / sum(ocenki)` В) `sorted(ocenki)`

6. Какво представлява набор от данни (dataset)?

- А) единична променлива
Б) организирана съвкупност от данни
В) програма за чертане на графики
Г) вид сензор

7. Файлт във формат CSV е:

- А) изпълнима програма
Б) текстов файл със стойности, разделени със запетая
В) некомпесирано изображение
Г) файл с аудио- и видеоданни

8. С коя част от данните проверяваме обучения модел?

- А) с тренировъчния набор
Б) с тестовия набор
В) с целия набор наведнъж
Г) моделът не се проверява

9. Преобучаване (overfitting) означава, че моделът:

- А) работи добре с нови данни
Б) се е приспособил твърде точно към конкретни данни
В) няма достатъчно данни и е склонен към халюциниране
Г) е без сензори

10. Кое НЕ е правило за добра заявка към ИИ?

- А) яснота
Б) специфичност
В) въвеждане на лични пароли
Г) посочване на желан формат

11. Защо казваме, че ИИ работи като „черна кутия“?

- А) защото е черен на цвят
- Б) защото не виждаме как точно стига до отговора си
- В) защото няма нужда от данни
- Г) защото работи само офлайн

12. Кой модул се използва за чертане на графики?

- A) csv B) random B) matplotlib Г) turtle

В предишните уроци се научихме да откриваме и да отстраняваме технически проблеми – с диспечера на устройствата (Device Manager) и диспечера на задачите (Task Manager). Днес ще добавим още един съвременен помощник: чатбот с изкуствен интелект. Той може бързо да ни ориентира при непознат проблем и да ни предложи възможни методи за решение, но трябва да го използваме разумно и да проверяваме съветите му. ИИ не просто дава готови отговори, а може да ви преведе стъпка по стъпка през процеса на отстраняване на повреди.

Кога чатботът с ИИ е полезен

Чатботът е особено удобен в началния етап на решаване на проблем, когато още не знаем откъде да започнем. Той може:

- да изброи възможните причини за даден симптом;
- да предложи последователност от стъпки за проверка;
- да обясни непознато съобщение за грешка с разбираеми думи;
- да предложи няколко алтернативни метода, между които да избираме.
- чрез насочващи въпроси ви помага да разберете дали проблемът е софтуерен (драйвер, вирус) или хардуерен.

Важно е да разбираме ролята му: чатботът предлага посока, а решението и проверката остават наша отговорност. Той не вижда нашия компютър и не познава точната ни ситуация – затова съветите му са общи и невинаги верни.

Как да формулираме проблема пред чатбота

Качеството на отговора зависи от качеството на заявката. При технически проблем добрата заявка съдържа:

- **Симптоми** – какво точно се случва (или не се случва).
- **Контекст** – кога започна проблемът (след актуализация, инсталиране на програма или токов удар).
- **Операционна система** – Windows, macOS, Linux и т.н.
- **Спецификации** – модел, процесор, видеокарта, оперативна памет на компютъра.
- **Грешки** – точен текст на съобщения за грешка или кодове.

Проблем: *USB устройство не се разпознава на вашия компютър.*

- Поставяме флашпамет, но тя не се появява в списъка с дисковете. Ето как би изглеждал примерен разговор с чатбот:

Ученик: Поставям USB флашпамет в лаптоп с ОС Windows, но тя не се появява в **This PC (Този компютър)**. На друг компютър работи. Какви са възможните причини и как да ги проверя поред?

ИИ: Възможни причини:

- 1) проблем с USB порта – пробвайте друг порт;
- 2) буквата на устройството липсва – проверете в **Disk Management (Управление на дискове)**;
- 3) драйверът е повреден – вижте в **Device Manager (Диспечер на устройствата)** за устройство с предупредителен знак;
- 4) файловата система не се разпознава. Започнете от най-простото.

Ученик: Как да отворя **Disk Management (Управление на дискове)**?

ИИ: Натиснете десен бутон върху бутона **Start (Старт)** и изберете **Disk Management (Управление на дискове)**.

Избор на метод между няколко предложени

Често чатботът предлага няколко метода за достигане до една и съща цел. В такива случаи той може да помогне да се сравнят различните методи и да се избере най-подходящият според поставената цел. Тогава наша задача е да изберем най-подходящия според ситуацията – например най-простия и безопасния. Това е умение за вземане на решение, а не сляпо изпълнение.

Да си припомним

- При технически проблем първо описваме точно симптома: какво се случва, кога започва, какво съобщение за грешка се появява.
- Чатботовете с ИИ са генеративен ИИ и че могат да „халюцинират“ – да изричат уверено неверни неща.

Правила за безопасност при работа с ИИ

Не споделяйте с чатбота лични данни, пароли или служебна информация.

Проверявайте съветите на чатбота, преди да ги изпълните – особено когато предлага да изтриете, форматируете или деинсталируете нещо.

Риск от „халюцинации“ – може да генерира убедително звучащи, но грешни или остарели инструкции.

Проблем: Компютърът работи много бавно.

- Ученикът пита чатбота:
„Компютърът ми с ОС Windows 10 работи много бавно. Кой е най-добрият начин да го ускоря?“
- ИИ може да предложи няколко метода:

Метод	Кога е подходящ	Предимства	Недостатъци
Освобождаване на място на диска	Дискът е почти пълен.	Лесно и безопасно.	Ефектът може да е ограничен.
Деактивиране на ненужни програми при стартиране	Компютърът стартира бавно.	Ускорява зареждането.	Не решава всички проблеми.
Сканиране за вируси	Има съмнение за злонамерен софтуер.	Повишава сигурността.	Отнема време.
Добавяне на RAM памет	Недостатъчна оперативна памет	Значително подобрява работата.	Изисква хардуерно надграждане.
Подмяна на HDD със SSD диск	Компютърът използва стар твърд диск.	Голямо увеличение на скоростта.	Изисква нов хардуер и прехвърляне на данните.

- Вместо веднага да препоръча един метод, чатботът може първо да зададе уточняващи въпроси и след това да предложи най-подходящото решение.

Основни понятия

Чатбот с изкуствен интелект – програма, базирана на генеративен ИИ, която води разговор на естествен език и може да помага при решаване на задачи.

Важно!

Ролята на чатбота е да предлага посока, а решението и проверката остават наша отговорност. Използването на чатбот не отменя нуждата да разбираме какво правим. Той не вижда нашия компютър и не познава точната ни ситуация – затова съветите му са общи и невинаги верни. При съмнение се консултирайте с учител или с надежден източник.

Задача 1. Изберете технически проблем, който сте срещали (програма, която не се отваря, липсваща мрежа). Формулирайте ясна заявка към чатбота, включваща симптома и какво сте опитали. Запишете предложените методи и ги подредете от най-безопасния към най-рисковия.

Задача 2. Помолете чатбота да обясни конкретно съобщение за грешка (реално или примерно). Проверете обяснението в още един източник. Съвпадат ли? Имаше ли неточности?

Задача 3. Създайте пет правила за сигурност под формата на плакат при работа с ИИ.

Въпроси и задачи

1. В кой етап от решаването на проблем чатботът с ИИ е най-полезен?
2. Какво трябва да съдържа добрата заявка при технически проблем?
3. Защо съветите на чатбота трябва да се проверяват преди изпълнение?
4. Кой носи отговорността за решението – чатботът или потребителят?

1. Кой от описаните проблеми НЕ е хардуерен?

- А) проблеми с операционната система
- Б) проблеми с видеокартата
- В) проблеми с централния процесор
- Г) намалена производителност на компютърната система

2. С помощта на диспечера за управление на хардуера (Device Manager) НЕ можете:

- А) да промените хардуерните настройки
- Б) да изключвате, включвате и деинсталирате устройства
- В) да инсталирате обновления за драйверите на устройствата
- Г) да проверявате работата на текущите програми

3. С помощта на диспечера на задачите (Task Manager) НЕ можете:

- А) да следите за използването на процесора, паметта и мрежовия трафик
- Б) да отстранявате проблемите, възникнали при работа с устройствата
- В) да проверявате работата на текущите програми
- Г) да премахвате зададени задачи

4. Кога чатботът с ИИ е най-полезен при решаване на проблем?

- А) в началото, за да ни ориентира за възможните причини и методи
- Б) когато проблемът вече е решен
- В) само за изтриване на файлове
- Г) никога не е полезен

5. Какво е добре да съдържа заявката към чатбот при технически проблем?

- А) само думата „помощ“
- Б) паролата на потребителя
- В) устройството, симптома, съобщението за грешка и какво вече е опитано
- Г) нищо конкретно

6. Кой носи отговорността за изпълнено решение, предложено от чатбот?

- А) чатботът
- Б) потребителят
- В) производителят на компютъра
- Г) никой

7. Кое твърдение за ИИ и околната среда е вярно?

- А) обучението на ИИ модели не изразходва енергия
- Б) ИИ има въглероден отпечатък, но помага и за опазване на природата
- В) ИИ работи без центрове за данни
- Г) ИИ няма връзка с околната среда

8. Защо препоръчителните системи могат да допринесат за дигитално пристрастяване?

- А) защото са бавни
- Б) защото изключват интернет
- В) защото не използват данни
- Г) защото са проектирани да задържат вниманието ни възможно най-дълго

ОБРАБОТКА НА ИНФОРМАЦИЯТА И КОМУНИКАЦИЯ

- Търсите информация в интернет, като оценявате нейната валидност и надеждност.

Задача: Кои са основните начини за търсене на информация в интернет?

- Описвате своите нужди за сигурността на информацията от гледна точка на поверителност, цялост, наличност, легитимност и липса на отказ.

- Познавате основните заплахи и предизвикателства в интернет.

- Спам е изпращането на нежелана информация на потребителите в интернет пространството.

Задача: Кои са основните начини за предпазване от получаване на нежелана поща?

Задача: Избройте основните правила за ползване на електронна поща.

- Познавате съвременните технологични средства за дигитално сътрудничество.

Задача: Кои са основните видове облачни технологии?

Задача: Създайте текстов документ с Google Docs (Гугъл Документи) и опишете в него различните облачни услуги в образованието. Съхранете и споделете документа в облачното хранилище за данни Google Drive (Гугъл Диск).

- Уебинар е провеждането на онлайн семинар в уебпространството.

Задача: Кои са основните ползи за образователните институции от провеждане на уебинари?

ПРОГРАМИРАНЕ И ИЗКУСТВЕН ИНТЕЛЕКТ

- Алгоритъмът е последователност от определен брой действия (команди) над определен набор от данни, с които се решава дадена задача.

Задача: Кои са основните изисквания при съставяне на алгоритъм?

- Познавате основните свойства на алгоритмите – определеност, масовост, резултатност, крайност, дискретност, формалност, сложност.

- Познавате основните видове алгоритми – линеен, разклонен, цикличен.

- Познавате различните средства за описание на алгоритмите.

- Списъкът в Python представлява структура от данни, която съхранява няколко стойности в определен ред под едно име.

- В Python има готови вградени функции, наречени агрегиращи (обобщаващи).

- Наборът от данни (dataset) е организирана съвкупност от данни, обикновено подредени в таблица с редове (записи) и колони (признаци).

Задача: Създайте линеен алгоритъм, който разменя стойностите на две променливи.

- Машинно самообучение е процес, при който система открива закономерности в данни и подобрява действието си с натрупването на повече примери.

- Моделът е резултата от обучението; програма, която прави предсказания за нови данни.

- Сензорът е устройство, което измерва физична величина (светлина, звук, температура, местоположение) и я превръща в данни.

Задача: Помолете чатбот да опише данните за брутния вътрешен продукт на България. Проверете данните в сайта на Националния статистически институт и отбележете дали има неточности или „халюцинации“.

РЕШАВАНЕ НА ПРОБЛЕМИ И БЕЗОПАСНОСТ

- Познавате най-често срещаните хардуерни и софтуерни проблеми, възникнали при използване на дигитални технологии.

- Можете да работите с диспечера за управление на хардуера (Device Manager) и диспечера на задачите (Task Manager).

- Можете да решавате конкретен проблем, възникнал при работа с компютърната система.

Задача: С помощта на Device Manager актуализирайте драйвера на устройство от компютърната система.

- Можете да използвате чатбот с ИИ, за да откривате и отстранявате технически проблеми.

Задача: Формулирайте ясна заявка към чатбот да опише какви действия да предприемете, ако на вашия компютър има наличие на вирус.

- Познавате въздействието, което оказва използването на дигиталните технологии върху здравето и околната среда.

- Използването на ИИ навлиза и в грижата за здравето.

1. За изготвяне на общ документ от няколко души, които се намират на голямо разстояние помежду си, е най-подходящо:

- А) да използват електронна поща
- Б) да използват облачни услуги
- В) да използват общ компютър веднъж седмично за редакция на документа
- Г) да обсъждат промените по телефона

2. Обмен на информация по дадена тема в определено време между хора, които използват компютърна мрежа, се извършва:

- А) чрез електронна поща
- Б) чрез уеб базиран семинар
- В) чрез телефон
- Г) чрез факс

3. Какви действия трябва да бъдат предприети, за да се гарантира безопасно използване при работа с дигиталните технологии и опазване на околната среда?

- А) да се унищожи изхабената батерия, като се изгори
- Б) да се изхвърли изхабената батерия в кош за боклук
- В) да се предаде изхабената батерия за рециклиране

4. Какви действия трябва да бъдат предприети, ако не може да се стартира компютърна програма?

- А) да се провери достатъчен ли е обемът на оперативната памет
- Б) да се прекъсне връзката с интернет
- В) да се преустанови работата на локалната мрежа
- Г) да се изтрият някои от другите компютърни програми

5. Каква ще бъде стойността на a , която ще се получи след изпълнение на дадения алгоритъм, ако са въведени числата $a = 5$ и $b = 14$?

Въведи a .

Въведи b .

Докато $a \neq b$, повтаряй:

ако $a > b$, тогава $a = a - b$;

в противен случай $b = b - a$.

Изведи a .

6. Кое от изброените НЕ е начин, по който изкуственият интелект влияе на информацията?

- А) масово създаване на съдържание
- Б) подбор на съдържание чрез препоръчителни системи
- В) създаване на дълбоки фалшификации
- Г) физическо повреждане на мрежовите кабели

7. Свойството формалност за един алгоритъм означава, че:

- А) той може да се използва с различни данни за решаване на определен тип задача, а не само с конкретния пример
- Б) може само да се изпълнява последователността от зададените команди, без да е нужно да се знае каква точно е задачата
- В) може да се извърши многократно, по всяко време, от различни хора, като за едни и същи начални данни ще се получи еднакъв резултат
- Г) се състои от отделни действия, като едва след изпълнението на текущото действие може да се пристъпи към изпълнението на следващото

8. Информационен балон (ехо-камера) се получава, когато:

- А) компютърът прегрява
- Б) потребителят среща предимно съдържание, потвърждаващо възгледите му
- В) паметта е запълнена
- Г) антивирусната програма блокира сайт

9. Кое е НАЙ-надеждното действие при съмнителна сензационна новина?

- А) веднага да я споделим
- Б) да проверим източника и да я потвърдим в няколко независими източника
- В) да я приемем за вярна, защото е на много места
- Г) да я изтрием от историята на браузъра