

Как мы, студенты-программисты, используем ИИ в обучении

Презентация о влиянии ИИ на учебный процесс студентов-программистов.

Введение

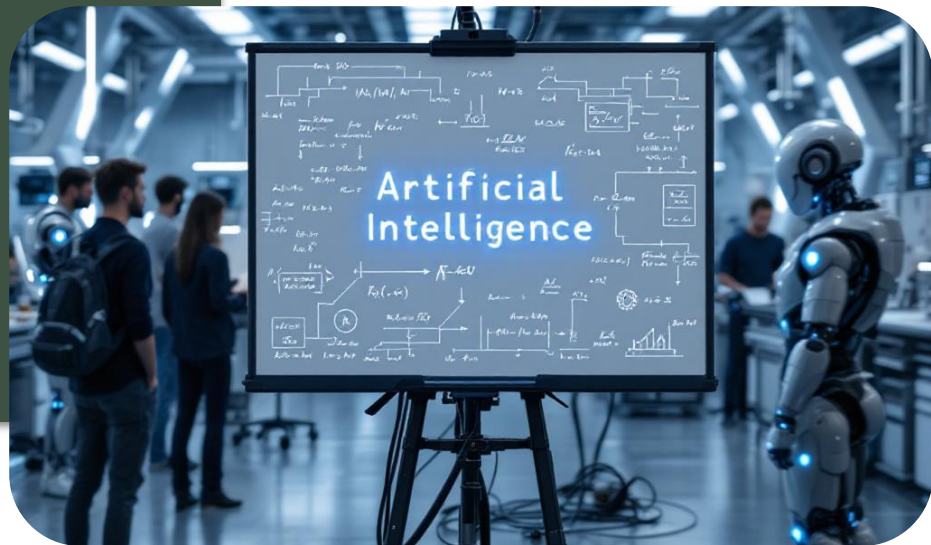
Цель презентации — показать, как ИИ помогает в обучении.




Что такое ИИ?

Искусственный интеллект – это способность машин обучаться и выполнять задачи, которые обычно требуют человеческого ума.

Примеры: распознавание речи, автодополнение кода.





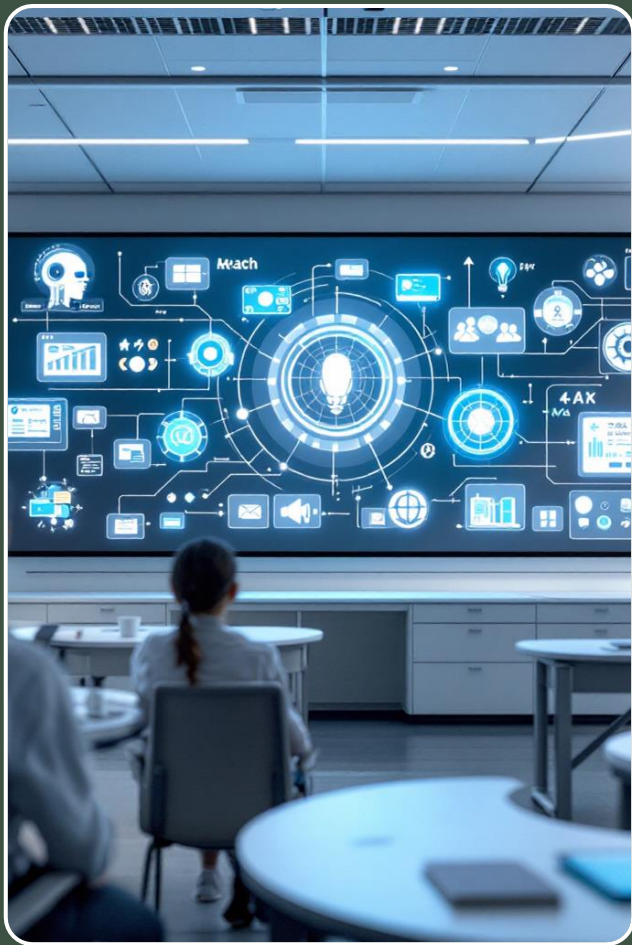
**Персонализация
образовательного
процесса**

**Мгновенная обратная
связь и адаптивное
обучение**

**Доступ к огромному
объёму информации**

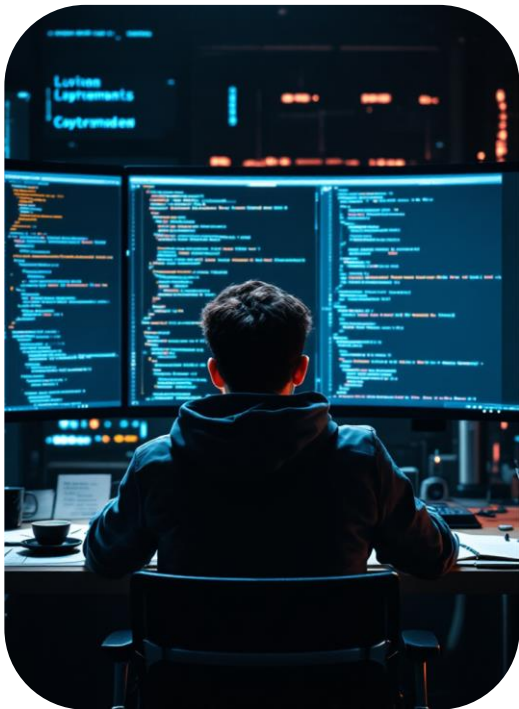
**Экономия времени за счёт
сжатия информации**

Роль ИИ в обучении



Обзор инструментов ИИ

- DeepSeek – для генерации идей и объяснения сложных тем
- GitHub Copilot – для автодополнения и генерации кода
- Codewars, LeetCode – платформа для практики навыков кодирования с элементами ИИ

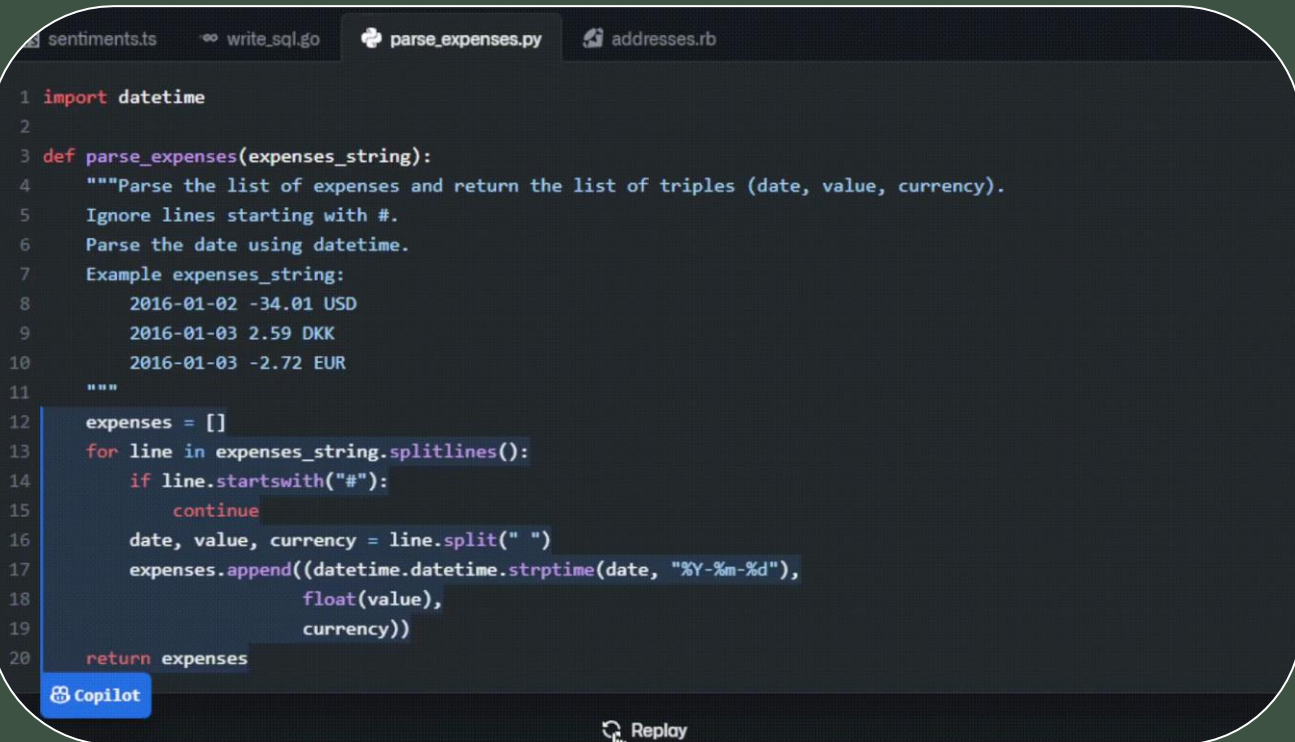


Генерация кода с помощью ИИ

ИИ помогает писать и оптимизировать код через автодополнение и подсказки.

Снижение времени на рутинные задачи.

Генерация кода при помощи GitHub Copilot



The screenshot shows a code editor with four tabs: `sentiments.ts`, `write_sql.go`, `parse_expenses.py` (active), and `addresses.rb`. The `parse_expenses.py` file contains the following Python code:

```
1 import datetime
2
3 def parse_expenses(expenses_string):
4     """Parse the list of expenses and return the list of triples (date, value, currency).
5     Ignore lines starting with #.
6     Parse the date using datetime.
7     Example expenses_string:
8         2016-01-02 -34.01 USD
9         2016-01-03 2.59 DKK
10        2016-01-03 -2.72 EUR
11    """
12    expenses = []
13    for line in expenses_string.splitlines():
14        if line.startswith("#"):
15            continue
16        date, value, currency = line.split(" ")
17        expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
18                        float(value),
19                        currency))
20    return expenses
```

GitHub Copilot suggestions are visible as blue highlights on lines 13-19. At the bottom left, there is a blue button with the Copilot logo and the text "Copilot". At the bottom center, there is a "Replay" button with a circular arrow icon.



**Автоматический поиск
ошибок**

**Рекомендации по
исправлению кода**

**Интеграция с популярными
IDE**

**Как ещё используется
GitHub Copilot.
Отладка и поиск
ошибок**

Решение задач и помощь с домашними заданиями

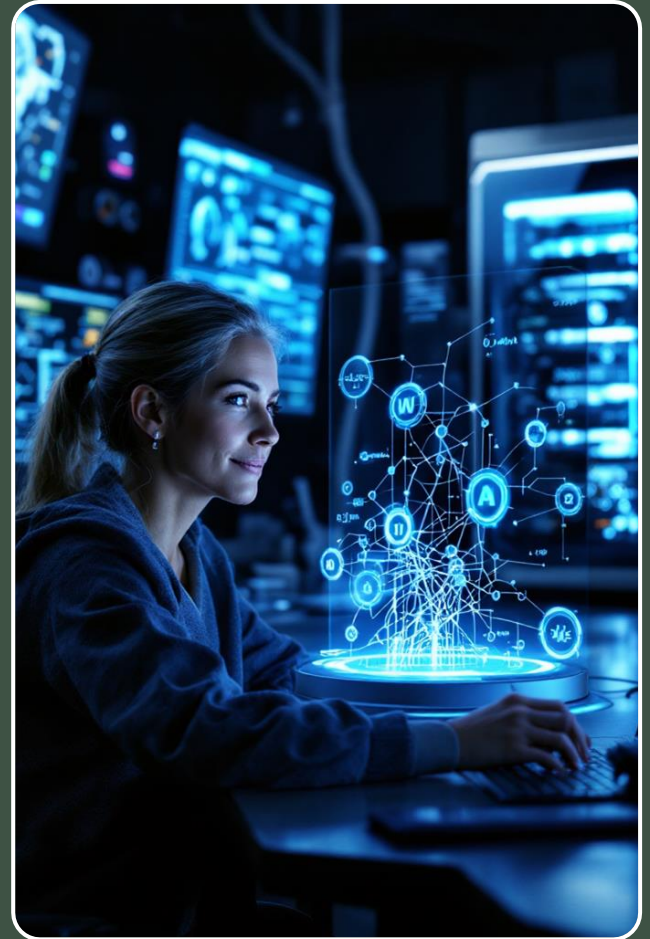
ИИ помогает в пошаговом объяснении алгоритмов и рекомендациях по решению задач.

Быстрый поиск оптимальных решений.



Поиск информации и проведение исследований

- Быстрый поиск релевантных статей и исследований
- Сбор и анализ данных
- Формирование библиографий и обзор литературы



Как помогает DeepSeek

Объяснение сложной темы

Как настроить параметр order в модели ARIMA из библиотеки statsmodel

1. Проверка стационарности ряда (определение d):

- **Цель:** Добиться стационарности ряда (постоянное среднее и дисперсия).

- **Методы:**

- Визуальный анализ графика ряда.

- **Тест Дики-Фуллера (ADF-test):**

```
python Copy

from statsmodels.tsa.stattools import adfuller

result = adfuller(series)
print(f'ADF Statistic: {result[0]}, p-value: {result[1]}')
```

- Если $p\text{-value} > 0.05$: ряд нестационарен → дифференцируйте его ($d += 1$).
- Повторяйте тест, пока не получите стационарность.
- На практике d редко превышает 2.

2. Определение p и q (AR и MA):

- **Анализ ACF и PACF:**

- **ACF (Autocorrelation Function)** помогает определить q .
- **PACF (Partial Autocorrelation Function)** помогает определить p .

```
python Copy

from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

plot_acf(series_diff, lags=20) # Для определения q
plot_pacf(series_diff, lags=20) # Для определения p
```

- **Правила:**

- p : Значение, после которого PACF резко обрывается.
- q : Значение, после которого ACF резко обрывается.

- **Пример:**

- Если PACF обрывается на лаге 2 → $p = 2$.
- Если ACF обрывается на лаге 3 → $q = 3$.

Как помогает DeepSeek

Поиск ошибок в коде

Почему при запуске парсера в список не попадает ни одного элемента:

```
url = 'https://www.imdb.com/chart/top/'
driver = webdriver.Chrome()
driver.get(url)
response = driver.page_source

#response = req.get(url, headers=headers)

soup = bs(response, 'html.parser')
titles = soup.find_all('a', calss_='ipc-title-link-wrapper', limit=250)
print(len(titles))
```

Ошибка возникает из-за опечатки в параметре `class_` (вместо `calss_`) и, возможно, из-за неверного указания класса элемента. Вот исправленный код:

python

Copy

```
from selenium import webdriver
```



Мозговой штурм с ИИ

**Генерация новых
концепций и решений**

**Адаптация идей под
конкретные задачи**

**Генерация идей для
проектов**



Визуализация данных с помощью ИИ

ИИ помогает создавать графики и диаграммы, улучшая восприятие сложных данных.

Автоматизированное создание визуализаций.

Построение ER-диаграммы при помощи DeepSeek

Ответ от ИИ

Мой ответ (код Mermaid):

mermaid

Copy

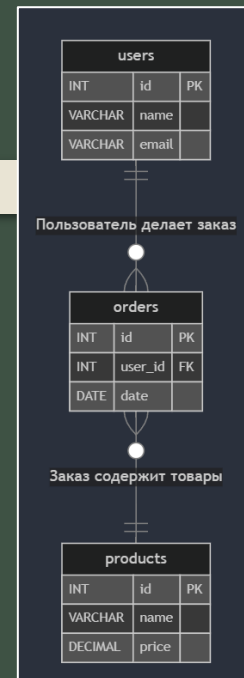
```
erDiagram
    users {
        INT id PK
        VARCHAR name
        VARCHAR email
    }

    orders {
        INT id PK
        INT user_id FK
        DATE date
    }

    products {
        INT id PK
        VARCHAR name
        DECIMAL price
    }

    users ||--o{ orders : "Пользователь делает заказ"
    orders }o--|| products : "Заказ содержит товары"
```

Результат



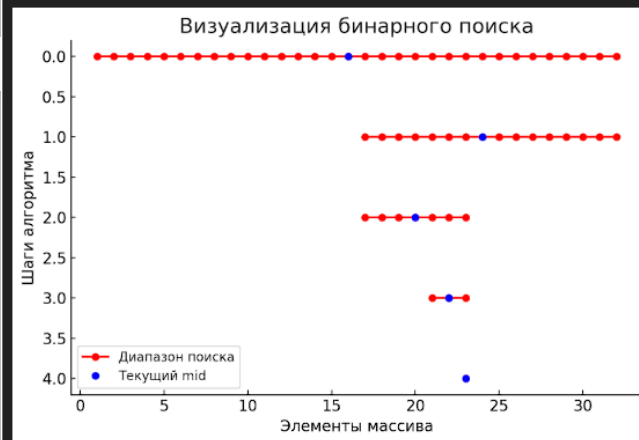
Визуализация бинарного поиска при помощи ChatGPT

Ответ от ИИ

Визуализируй принцип работы алгоритма бинарного поиска (при помощи графика).

Бинарный поиск — это алгоритм поиска элемента в отсортированном массиве, который работает по принципу деления диапазона пополам. Я визуализирую его работу с помощью графика, показывая, как область поиска уменьшается на каждом шаге.

Сейчас сгенерирую график. 📊




На графике показаны шаги работы бинарного поиска.

• Красные точки и линии — диапазон поиска на каждом шаге.

• Синие точки — текущий центральный элемент (`mid`).

Каждый шаг поиска сокращает диапазон вдвое, пока не будет найден искомый элемент (23). 🎯



**Экономия времени на
рутинных задачах**

**Повышение качества и
точности выполнения
заданий**

**Расширение возможностей
для саморазвития**

Преимущества использования ИИ

Вызовы и ограничения

Не все ИИ бесплатны

Злоупотребление ИИ делает человека неспособным к самостоятельному поиску информации и приводит к неспособности человека думать своей головой

Возможные ошибки



Заключение

ИИ с каждым годом всё больше
вливается в повседневную
человеческую жизнь, тем самым влияя
и на процесс обучения.

ИИ помог нам интерпретировать
сложные вещи простым языком.



Спасибо за внимание!