



# Вводный семинар. Библиотека NumPy

Андрей Куртасов  
Системный аналитик

# Проверка связи



Отправьте «+», если меня видно и слышно

Если у вас нет звука или изображения:

- перезагрузите страницу
- попробуйте зайти заново
- откройте трансляцию в другом браузере (используйте Google Chrome или Microsoft Edge)
- с осторожностью используйте VPN, при подключении через VPN видеопотоки могут тормозить

# План семинара



1. Знакомство и мотивация
2. Предмет машинного обучения
3. Структура нашего курса
4. Инструменты машинного обучения в экосистеме Python
5. Основы NumPy
6. Немного практики

# Знакомство и мотивация

# Давайте знакомиться



## Андрей Куртасов

- Системный аналитик в ООО «Техсофт» – компании по разработке медицинского ПО
- Участвовал в проектах по анализу медицинских данных, обработке изображений и текстов
- Также преподаю программирование на Python

Пожалуйста, напишите ответ в чате

- По какой специальности обучаетесь в магистратуре?
- Каковы ваши ожидания от курса «Продвинутые методы машинного обучения»?



# Как будем работать?



- Не стесняйтесь задавать вопросы!
  - На занятии пишем вопросы в чат Pruffme. Постараюсь зачитать и ответить.
  - Задавая вопрос в чате, формулируйте его полностью в виде одного сообщения.
  - Если в чат обратился кто-то из ваших коллег, и вы знаете ответ на вопрос, можете смело отвечать и поддерживать дискуссию.

# Как будем работать?



- Не стесняйтесь задавать вопросы!
  - На занятии пишем вопросы в чат Pruffme. Постараюсь зачитать и ответить.
  - Задавая вопрос в чате, формулируйте его полностью в виде одного сообщения.
  - Если в чат обратился кто-то из ваших коллег, и вы знаете ответ на вопрос, можете смело отвечать и поддерживать дискуссию.
- Подключитесь к чату курса в Telegram: [t.me/+\\_cX\\_V-c\\_psFIZDAy](https://t.me/+_cX_V-c_psFIZDAy)
  - После семинара задаем вопросы [в соответствующей ветке](#) (до 14:00 понедельника).
  - Организационные вопросы направляем [@curator\\_mipt](#).
  - Поддерживайте связь с другими слушателями курса. Это возможность найти полезные контакты и расширить свои знания.
  - Не следует просить участников чата полностью решить за вас задачу. Сперва покажите, что у вас уже получилось (фрагменты кода, стектрейсы с ошибками).

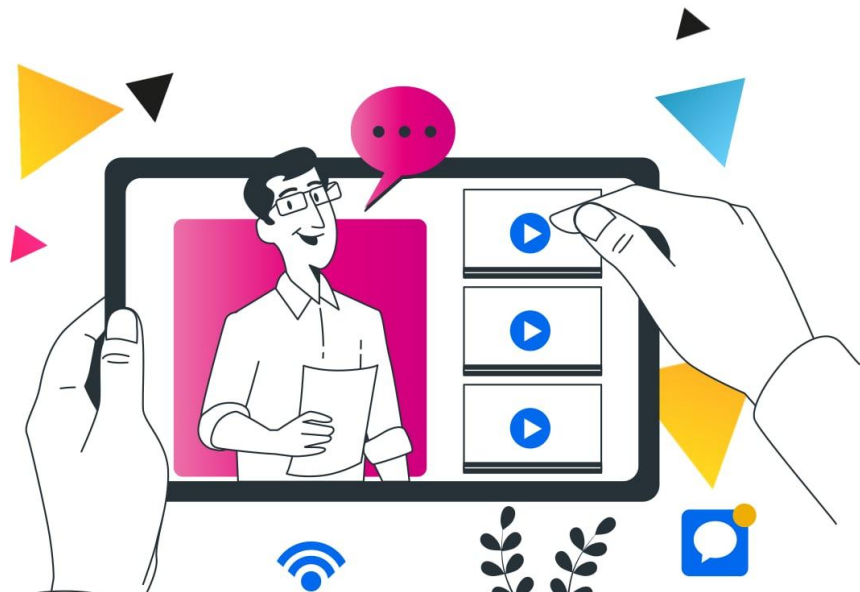


# Как будем работать?



- Не стесняйтесь задавать вопросы!
  - На занятии пишем вопросы в чат Pruffme. Постараюсь зачитать и ответить.
  - Задавая вопрос в чате, формулируйте его полностью в виде одного сообщения.
  - Если в чат обратился кто-то из ваших коллег, и вы знаете ответ на вопрос, можете смело отвечать и поддерживать дискуссию.
- Подключитесь к чату курса в Telegram: [t.me/+\\_cX\\_V-c\\_psFIZDAy](https://t.me/+_cX_V-c_psFIZDAy)
  - После семинара задаем вопросы [в соответствующей ветке](#) (до 14:00 понедельника).
  - Организационные вопросы направляем [@curator\\_mipt](#).
  - Поддерживайте связь с другими слушателями курса. Это возможность найти полезные контакты и расширить свои знания.
  - Не следует просить участников чата полностью решить за вас задачу. Сперва покажите, что у вас уже получилось (фрагменты кода, стектрейсы с ошибками).
- Записи семинаров будут доступны в LMS
  - Если смотрите семинар в записи, можете также задавать вопросы [в ветке чата](#).

Ваши вопросы?  
Пишите в чат или поднимайте руку.



# Предмет машинного обучения

# Что такое машинное обучение?

- Машинное обучение — это деятельность по разработке компьютерных систем, поведение которых определяется не явным программированием, но, в первую очередь, **данными**, получаемыми из реального мира.

# Что такое машинное обучение?

- Машинное обучение — это деятельность по разработке компьютерных систем, поведение которых определяется не явным программированием, но, в первую очередь, **данными**, получаемыми из реального мира.
- В основе таких систем лежат **алгоритмы**, позволяющие строить из доступных данных **модели**, исходя из которых система определяет свое поведение в ответ на данные, с которыми она не сталкивалась ранее.

Пожалуйста, напишите ответ в чате

Приведите примеры приложений  
машинного обучения. Что их объединяет?

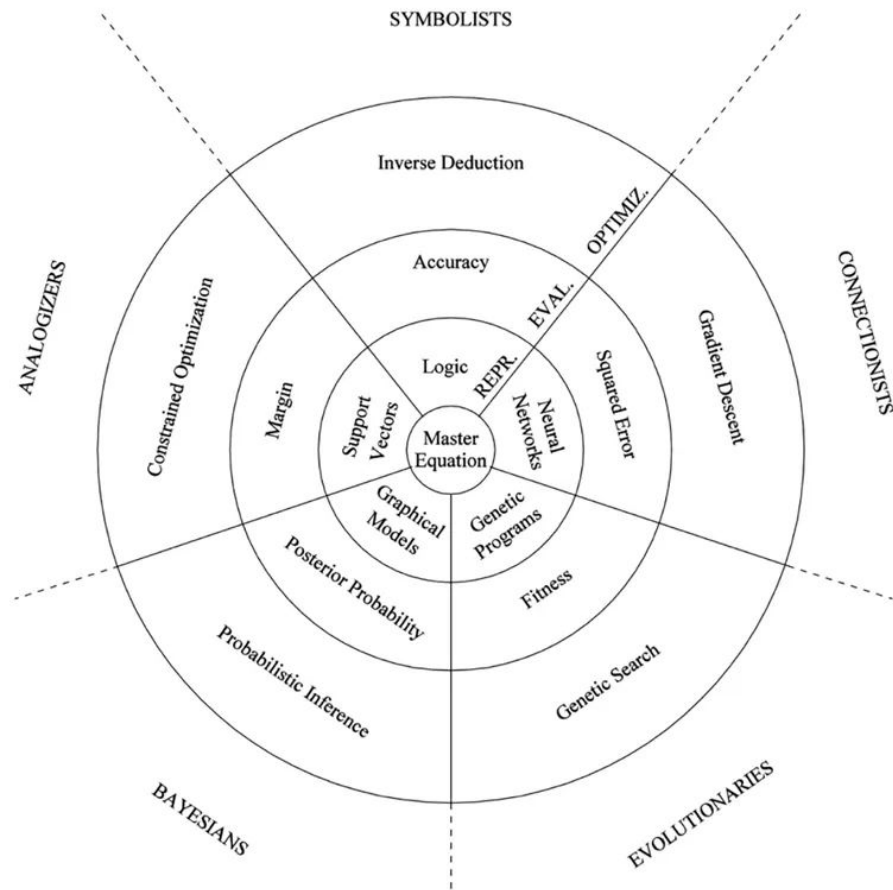


# Пять «школ» машинного обучения

1. Символисты
2. Коннекционисты
3. Эволюционисты
4. Байесовцы
5. Аналогисты

См. Домингос, Педро. *Верховный алгоритм: как машинное обучение изменит наш мир* / П. Домингос; пер. с англ. В. Горохова. — М. : Манн, Иванов и Фербер, 2016. — 336 с.

Источник изображения: [medium.com/42ai/the-5-tribes-of-the-ml-world-670ebce96b4c](https://medium.com/42ai/the-5-tribes-of-the-ml-world-670ebce96b4c)

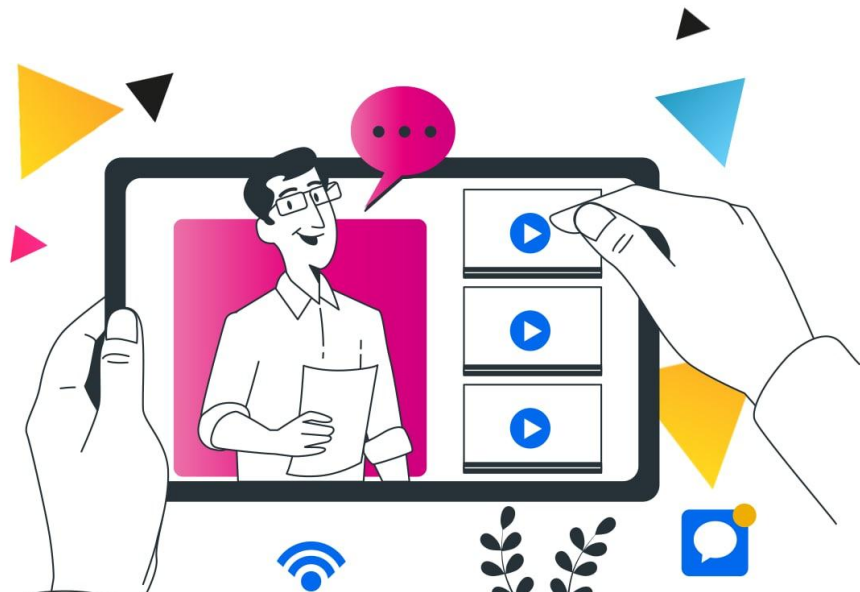


# Области знаний, связанные с МО

- Отдельные главы математики:
  - Линейная алгебра
  - Теория вероятности
  - Прикладная статистика
  - Численные методы оптимизации и др.
- Инжиниринг *больших данных* и ETL-системы
- Анализ данных
- Визуализация данных



Ваши вопросы?  
Пишите в чат или поднимайте руку.



# Структура нашего курса

## Модуль 1. Python для анализа данных

1. Библиотека NumPy
2. Получение и предобработка данных. Первичная работа с объектом DataFrame
3. Описательная статистика. Анализ данных с помощью Pandas
4. Статистика вывода<sup>1</sup>

<sup>1</sup>См. [www.kaggle.com/discussions/questions-and-answers/198927](https://www.kaggle.com/discussions/questions-and-answers/198927)

## Модуль 2. Визуализация данных

1. Введение в визуализацию. Библиотека matplotlib
2. Прикладные инструменты визуализации данных
3. Диаграммы в контексте: инфопанели и презентации
4. Примеры использования библиотек Python

## Модуль 3. Введение в машинное обучение

1. Введение в линейную алгебру для машинного обучения
2. Машинное обучение с учителем: линейные модели, измерение качества модели, ансамблевые модели
3. Машинное обучение без учителя: методы кластеризации, методы понижения размерности, рекомендательные системы
4. Нейросети: основы нейронных сетей, архитектуры нейронных сетей

- Метод ближайших соседей
- Задача регрессии. Линейная регрессия
- Задача классификации. Логистическая регрессия.  
Наивный Байесовский классификатор
- Метод опорных векторов
- Решающие деревья. Случайный лес. Градиентный бустинг
- Нейронные сети. Глубокое обучение
- Обработка текста. Векторные представления слов.  
Механизм внимания
- Обработка изображений. Сверточные нейронные сети

Вам слово! Пишите в чат или поднимайте руку

- Какие из перечисленных тем вам наиболее интересны?
- Есть ли какие-либо сомнения или опасения по поводу содержания или сложности курса?



# Инструменты МО в экосистеме Python



## Проведем опрос

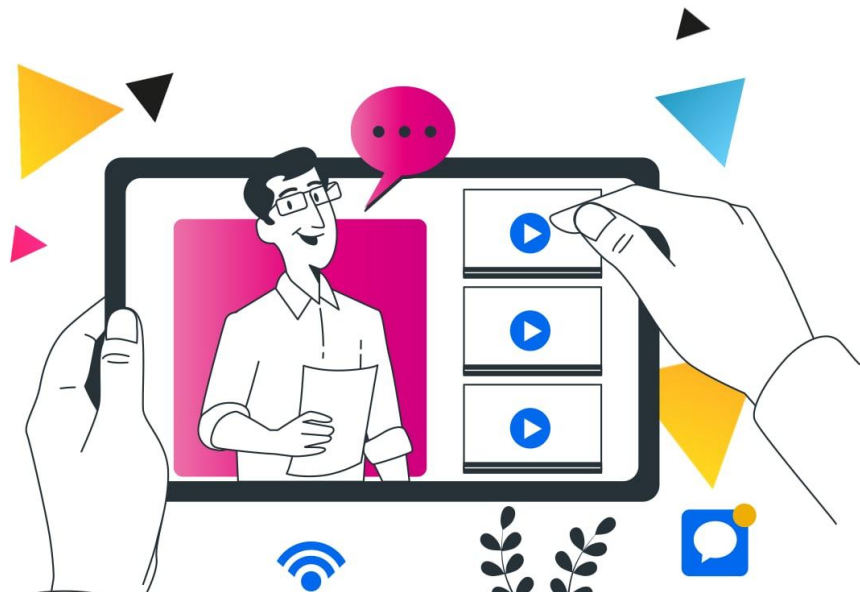
С какими из этих инструментов вам уже приходилось иметь дело?

- Язык Python
- NumPy
- SciPy
- pandas
- scikit-learn
- PyTorch



- Google Colaboratory – реализация iPython Notebook от компании Google
- Доступна в бесплатной версии
- Установлен Python 3.8 и все типовые библиотеки
- Можно устанавливать дополнительные библиотеки через `pip` и даже дополнительное ПО через `apt-get` (внутри Ubuntu)
- Ноутбуками можно делиться!

Ваши вопросы?  
Пишите в чат или поднимайте руку.



# Основы NumPy

- NumPy – библиотека для вычислительных операций с  $n$ -мерными однородными массивами (все элементы массивов – одного типа).
- Написана на языке C – массивы представляются как векторы, что обеспечивает скорость работы.
- В библиотеку включен широкий набор математических функций, включая функции линейной алгебры.
- По сравнению с pandas:
  - Меньшее потребление памяти при обработке чисел
  - Более быстрое индексирование
  - → Часто используется в сочетании с pandas

# Векторы и матрицы



```
vector_row = np.array([1,2,3])
```

```
print(vector_row)
```

```
[1 2 3]
```

```
print(vector_row.shape)
```

```
(3,)
```

```
vector_column = np.array([[1],[2],[3]])
```

```
print(vector_column)
```

```
[[1]
```

```
 [2]
```

```
 [3]]
```

```
print(vector_column.shape)
```

```
(3, 1)
```

```
matrix = np.array([(1,2,3),(4,5,6)])
```

```
print(matrix)
```

```
[[1 2 3]
```

```
 [4 5 6]]
```

```
print(matrix.shape)
```

```
(2, 3)
```

# Умножение матриц

```
a = 3 * np.ones((5, 5))
b = np.eye(5) + 1
print(f'a = {a}')
```

```
print(a * b)  # Поэлементное умножение
print(a @ b)  # Матричное умножение
```

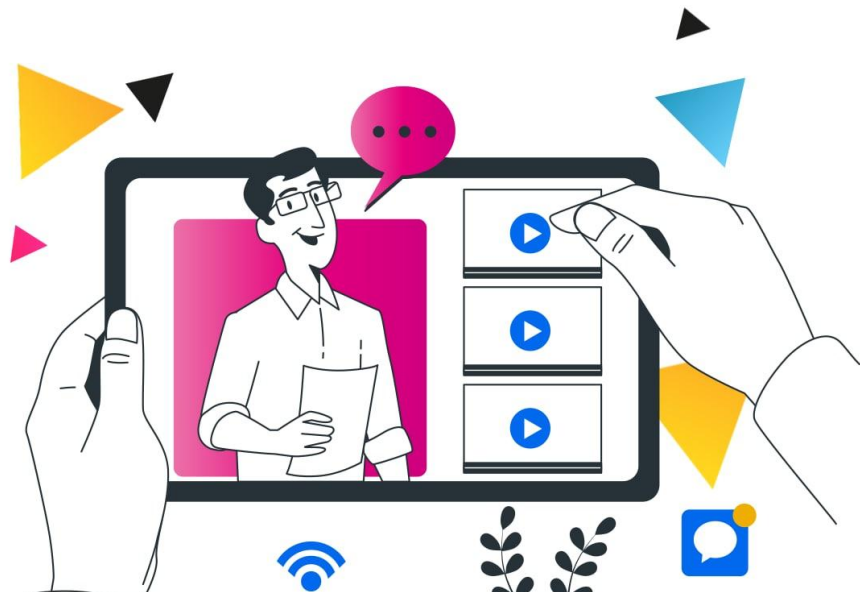
```
a = [[3. 3. 3. 3. 3.]
      [3. 3. 3. 3. 3.]
      [3. 3. 3. 3. 3.]
      [3. 3. 3. 3. 3.]
      [3. 3. 3. 3. 3.]]
```

```
[[6. 3. 3. 3. 3.]
 [3. 6. 3. 3. 3.]
 [3. 3. 6. 3. 3.]
 [3. 3. 3. 6. 3.]
 [3. 3. 3. 3. 6.]]
```

```
b = [[2. 1. 1. 1. 1.]
      [1. 2. 1. 1. 1.]
      [1. 1. 2. 1. 1.]
      [1. 1. 1. 2. 1.]
      [1. 1. 1. 1. 2.]]
```

```
[[18. 18. 18. 18. 18.]
 [18. 18. 18. 18. 18.]
 [18. 18. 18. 18. 18.]
 [18. 18. 18. 18. 18.]
 [18. 18. 18. 18. 18.]]
```

Ваши вопросы?  
Пишите в чат или поднимайте руку.





# Немного практики

# Упражнение



1. Решить систему линейных уравнений.
2. Записать корни системы в переменные  $x$ ,  $y$ ,  $z$ .

$$x + y + z = 6$$

$$2y + 5z = -4$$

$$2x + 5y - z = 27$$

Пользуйтесь материалами лекций или документацией NumPy:

[numpy.org/doc/stable/reference/routines.linalg.html](https://numpy.org/doc/stable/reference/routines.linalg.html)

# Упражнение



1. Решить систему линейных уравнений.
2. Записать корни системы в переменные  $x$ ,  $y$ ,  $z$ .
3. Убедиться, что произведение матрицы коэффициентов  $A$  на матрицу корней  $X$  равно матрице  $B$  правой части уравнения.

$$x + y + z = 6$$

$$2y + 5z = -4$$

$$2x + 5y - z = 27$$

# Упражнение

1. Решить систему линейных уравнений.
2. Записать корни системы в переменные  $x, y, z$ .
3. Убедиться, что произведение матрицы коэффициентов  $A$  на матрицу корней  $X$  равно матрице  $B$  правой части уравнения.
4. Найти матрицу  $A_{inv}$ , обратную матрице  $A$ , и убедиться, что произведение  $A_{inv}$  на  $A$  дает единичную матрицу.

$$x + y + z = 6$$

$$2y + 5z = -4$$

$$2x + 5y - z = 27$$

## Свободная дискуссия

# Ваши вопросы? Пожелания?



До встречи!

