# Дизайн систем машинного обучения

## 1. Машинное обучение на практике

**https://ods.ai/tracks/ml-system-design-23**

# Команда курса

- Дмитрий Колодезев https://kolodezev.ru/cv2023.html
- Ирина Голощапова
- Татьяна Архипова и команда ODS.AI
- Михаил Симаков
- Волонтеры из ИТМО
- Екатерина Колодезева
- И много кто еще

# О чем будем рассказывать

- Делать модели машинного обучения легко
- Трудно сделать так, чтобы они работали хорошо
- Еще труднее сделать, чтобы ими пользовались
- Будем изучать ML-системы в реальной жизни
- С точки зрения кода, оборудования и бизнеса

# Чего в курсе нет

- Алгоритмы машинного обучения
- Дизайн пользовательского интерфейса
- Статистика
- Как писать код
- Как учить нейронные сети
- Как делать веб-сайты
- Как проходить собеседование по system design

# План курса

1) **Машинное обучение на практике — Вы находитесь здесь**
2) Основы проектирования ML-систем
3) Обучающие данные
4) Подготовка и отбор признаков
5) Выбор и обучение ML-модели
6) Улучшение модели через данные
7) Оценка качества модели
8) Развертывание
9) Диагностика ошибок и отказов
10) Жизненный цикл модели
11) Потоковые данные
12) Языковые модели в продуктовом окружении
13) Временные ряды и графы
14) Безопасность, этика и восстание машин
15) Интеграция в бизнес-процессы

# ! ACHTUNG !

Курс объемный, делается в свободное от жизни время.

- Организаторы — приятные в общении, внимательные и чуткие люди, вынуждены на время курса переключиться в безопасный режим, а именно:
- банить направо и налево без предупреждения
- игнорировать любые разумные аргументы.

Опыт показывает, что в этих условиях можно многому научиться, но жаловаться бесполезно.

# Зачем вам это может пригодиться

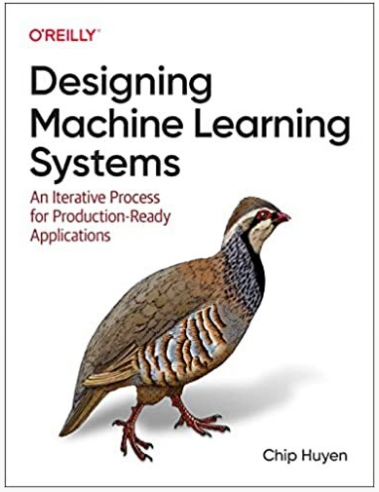
- Запустить свой пет-проект для портфолио
- Запустить свой стартап
- Переосмыслить то, что делаете на работе
- Для расширения кругозора (главное назначение)

# Как проектируют

- Задают вопросы
- Ставят под сомнение факты
- Оценивают имеющиеся ресурсы
- Вспоминают, как сделали на прошлой работе
- Половина успеха — выявление ограничений
- Вторая половина успеха — типовые решения
- Третья половина успеха — насмотренность
- Четвертая половина успеха — выход из тупиков

# Стоим на плечах гигантов

- Курс cs329 Machine Learning System Design
- Книга Designing Machine Learning Systems




# Предварительные требования

- Любой ВУЗовский курс статистики
- Онлайн-курс https://stepik.org/course/76/info

# Предварительные требования

- Любой ВУЗовский курс по программированию
- https://habr.com/ru/companies/yandex/articles/498856/

# Предварительные требования

- Любой ВУЗовский курс по машинному обучению
- https://habr.com/ru/companies/ods/articles/322626/

# Предварительные требования

- Практический опыт программирования под Linux
- Онлайн-курс https://missing-semester-rus.github.io/

# Предварительные требования

- Желательно — опыт работы с нейронными сетями
- например, pytorch. https://pytorch.org/tutorials/
- https://huggingface.co/learn
- https://www.deeplearning.ai/short-courses/

# Что будет

- Видеолекции
- Семинары
- Доклады участников
- Канал курса в телеграме
- Проект
- ML дизайн документ
- Лабораторные работы
- Дополнительные материалы
- Публичный лидерборд

# Лекции

- Раз в неделю на странице курса
- Примерно полчаса-час, как получится
- Текстовые расшифровки будут на kolodezev.ru
- Наполовину — повторение прошлого курса
- Записи будут в общем доступе
- Баллы — еженедельно за обсуждение лекции в чате
- 1 — за хороший вопрос или ответ на чужой вопрос
- 2 — за огонь вопрос или за огонь ответ

# Семинары

- По четвергам в 14:00 мск в SpatialChat
- Доклады участников
- Ответы на вопросы
- Примерно час, записей не будет

# Доклады участников

- Список тем для докладов будет к четвергу
- Можно свою тему
- Формат: 10-минутный рассказ на семинаре или статья на Хабре или статья на medium или репозиторий с кодом и презентацией в markdown
- Оценка от 0 до 20 баллов
- Если несколько — за один лучший
- Материал +10, подача +5, доп. Материалы +5
- Согласовать выступление / тему заранее

# Канал в телеграме

- Еженедельные отчеты о проектах
- Вопросы к лекциям
- Объявления
- Неактивные участники будут удаляться
- Добавить кого-то, кто не проходит курс, нельзя
- Всем, кто указал аккаунт в телеграме, придет инвайт.
- Все, кто не получил инвайта — пишите Татьяне

# Проект

- В этом году мы не предлагаем готовые темы
- Выбирайте проект, которым вы хотели бы заниматься после курса или уже занимаетесь
- Команда от 2 до 5 человек, одному нельзя
- Репозиторий на github, доступ @promsoft
- Оценки за проект всей команде
- Еженедельно до четверга — коммит и отчет
- Хороший коммит — 1 балл
- Коммит + хороший отчет — 2 балла
- Коммит + отчет + интересный результат — 3 балла

# ML дизайн документ

- Отдельный раздел в репозитории с ответами:
  - Что хотим построить и зачем
  - Как поймем, что проект успешен
  - Ограничения и возможности
  - Подходы к построению системы
- Оцениваются так же как проекты, раз в неделю
- Лекция про дизайндок https://youtu.be/HmdKhI2_6Os
- Шаблон и лекция прошлого года
- Можно упрощать, сохраняя общую идею

# Лабораторные работы

- Возможно, будут 4 лабораторные работы
- К 3, 6, 9 и 12 лекции
- Не оцениваются
- Практические задачи на «попробовать»
- Результат обсудим на семинаре

# Дополнительные материалы

- Статьи для самостоятельной проработки (30-50 страниц)
- Будут в телеграм-канале после каждой лекции.
- Задавайте вопросы по дополнительным материалам в телеграм-канале.
- Дополнительные материалы важнее лекций.

# Баллы

- Работа на лекциях — 30 баллов личное
- Доклады студентов — 20 баллов личное
- Работа над проектом — 42 балла команде
- Дизайн документ — 42 балла команде
- Защита проекта — 40 баллов команде
- <60 двойка, <90 тройка, < 110 четверка
- >=140 рекомендательное письмо от автора
- Сертификаты и мерч от ODS.AI — уточним позже

# Дизайн систем машинного обучения

- Процесс принятия решений про
  - Интерфейс
  - Алгоритмы, данные
  - Программную инфраструктуру
  - Оборудование
- Чтобы соответствовать требованиям и ограничениям, например, по:
  - Надежности (reliable)
  - Масштабируемости (scalable)
  - Обслуживаемости (maintainable)
  - Адаптируемости (adaptable)

# Дизайн начинается с ограничений

- Требования/ограничения придется выяснять самим
- Если кто-то выдал вам требования, они неполные и противоречивые
- Требования — всегда предположения
- Предположите что-нибудь
- Придумайте, как проверить предположение
- Узнав новое, уточните предположения

# Предположения ML

Машинное обучение — это автоматизированный подход к выявлению сложных шаблонов в имеющихся данных и использование этих шаблонов для предсказаний на новых данных.

- Мы можем выявить шаблоны
- Шаблоны сложные
- Данные имеются
- Можем предсказывать
- Будут новые данные

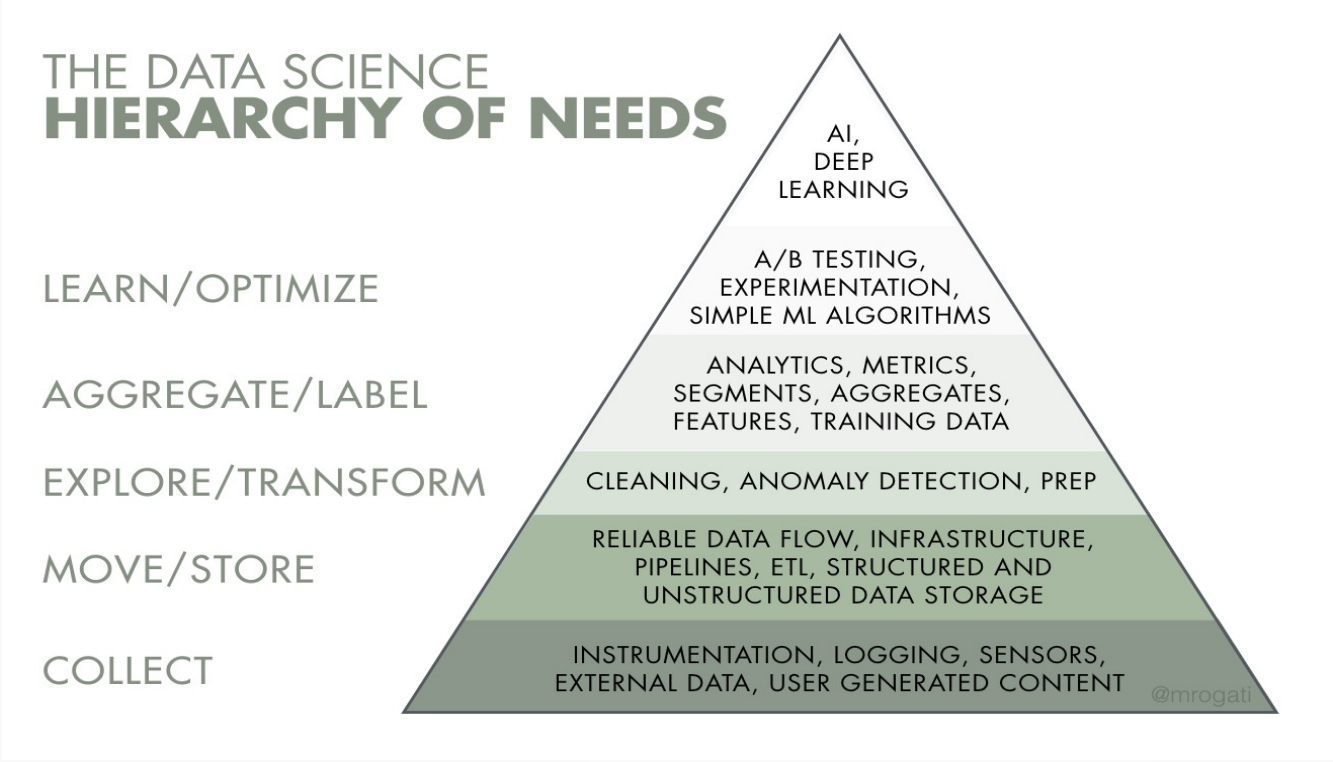
# Делать ML

- Часто повторяющаяся задача
- Цена ошибки невелика
- Большой масштаб
- Шаблоны постоянно меняются

# Не делать ML

- Это неэтично
- Простое правило решает проблему
- Данные недоступны
- Цена ошибки высока
- Каждое решение должно быть объяснимо
- Дешевле нанять человека

# Основание




https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007

# Индустрия vs Исследования


* Автор курса на стороне индустрии

Часто и небезосновательно:

- Ученые не умеют писать код
- Не воспроизводится
- Неприменимо

Часто и небезосновательно:

- Инженеры не знают основ
- Лишь бы работало
- Не проверяют базовые предположения

# Исследования

- Требования: Публикабельность
- Вычисления: Быстрое обучение и пропускная способность
- Данные: Обычно не меняются
- Интерпретируемость: Обычно не важна
- Поддерживаемость: Не важна
- Масштабируемость: Один и тот же масштаб

# Индустрия

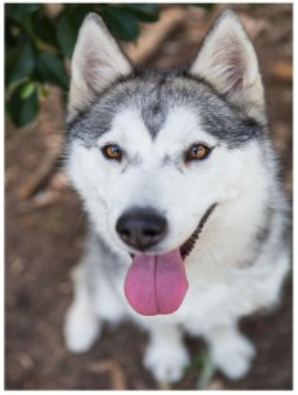
- Требования: Разные требования внутри организации
- Вычисления: Быстрый инференс и низкая задержка
- Данные: Постоянный сдвиг данных
- Интерпретируемость: Может быть очень важна
- Поддерживаемость: Может быть определяющей метрикой
- Масштабируемость: Может быть определяющей метрикой

# Разные интересы

## Stakeholder objectives

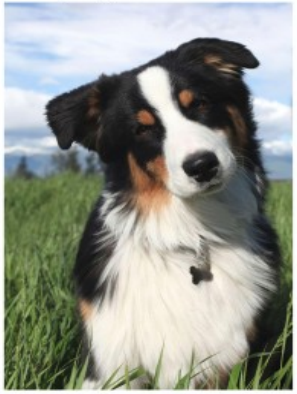
**ML team**
highest accuracy

**Sales**
sells more ads



**Product**
fastest inference

**Manager**
maximizes profit
= laying off ML teams



https://docs.google.com/presentation/d/1XKBXoLW3ED3FcvDRK3jfQeIlLCK2QTnsoZeopR6ziRQ/edit#slide=id.gb4b40ae9d8_0_16

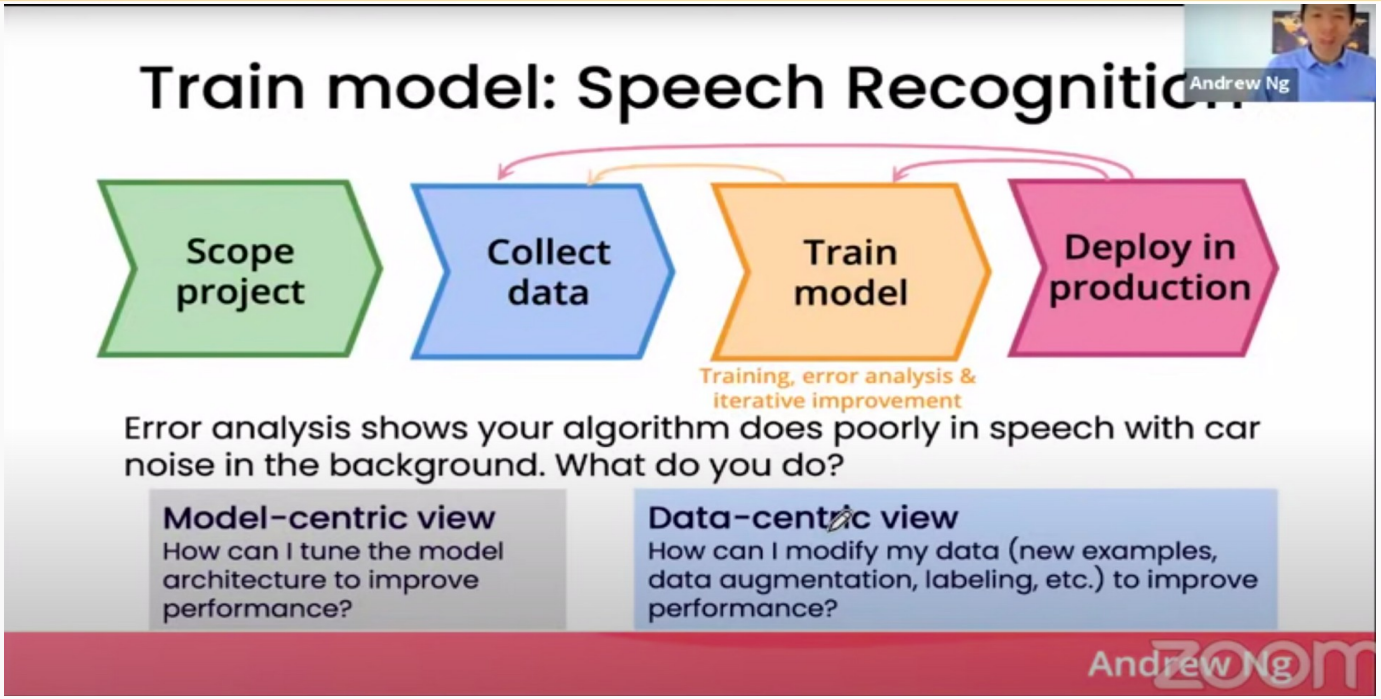
# Как ML меняет разработку

- Данные
- Управление проектом
- Мониторинг

# Данные

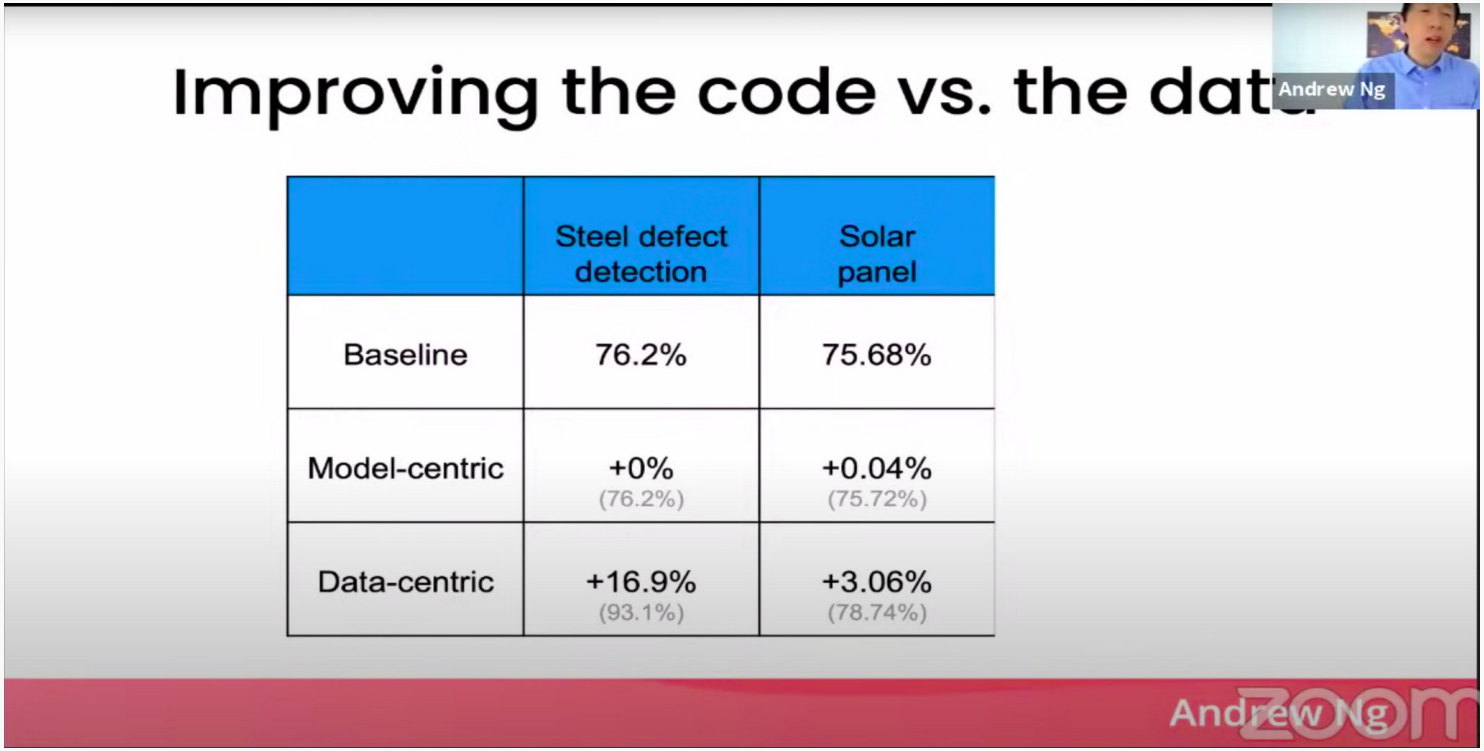
- ML-модель сцеплена с данными
- Данные **обычно** не версионируются
  - Версионируются схемы данных, но не данные
- Данные **обычно** необозримы
  - Какая точка данных портит вашу модель?
  - Данных **обычно** больше, чем кода
  - Diff **обычно** не работает
- Предположения о данных нужно проверять
  - Нужно, но трудно

# Данные важны



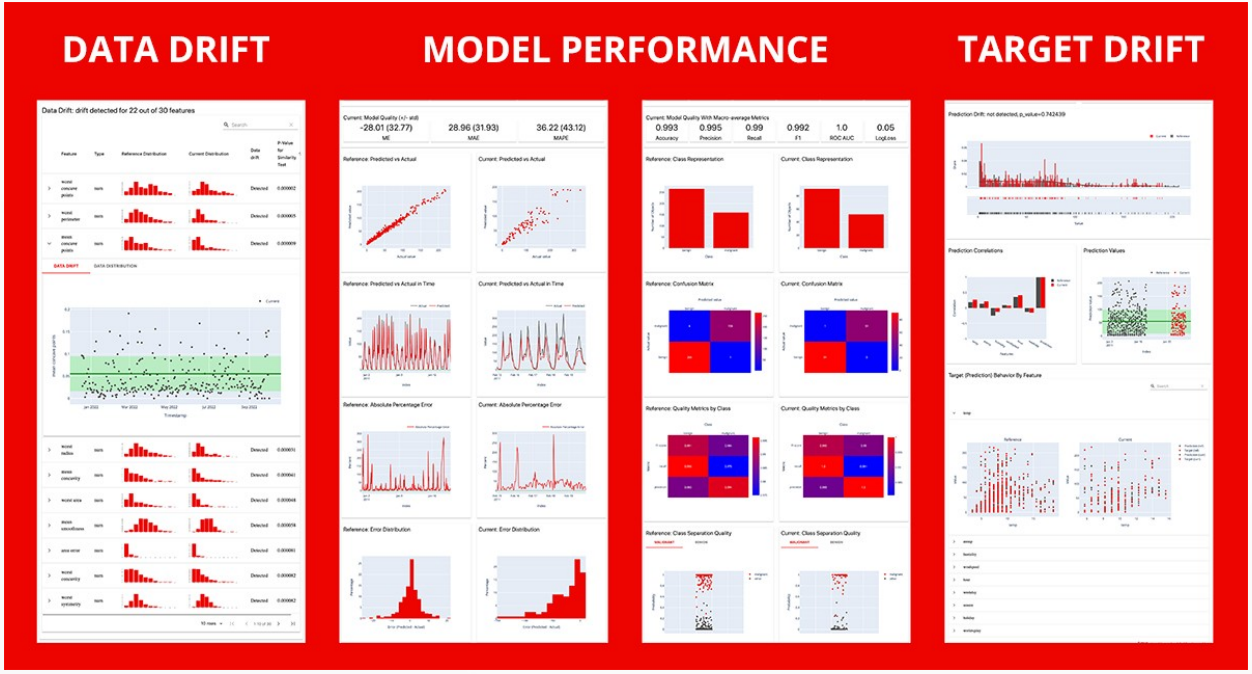

https://www.youtube.com/watch?v=06-AZXmwHjo

# Данные важнее моделей




https://www.youtube.com/watch?v=06-AZXmwHjo

# Данные меняются




https://github.com/evidentlyai/evidently

# Управление проектом

- Трудно управлять качеством
- Трудно управлять сроками
- Непонятны границы возможного
- Дефекты возникают не в коде
- Но править их приходится в коде
- Двухфазные проекты:
  - Discovery: фиксированное время
  - Delivery: фиксированные задачи

# Мониторинг

- Незаметно сломалась
- Или сразу не заработала
- Проблемы с данными
  - аномалии, выбросы, редкие значения, пропуски.
  - поменялось распределение данных
- Проблемы с метриками
- Проблемы с обратной связью
- Программные, аппаратные, организационные отказы

# Вроде как правила

- 50% эффекта ML-модели можно достичь без ML, готовым API, или очень простой моделью
- Если эксперт не видит в данных закономерностей, ML-модель тоже не увидит
- Большинство проблем — на границах системы
- Основные затраты — данные
- ML-метрики не важны заказчику
- Метрики, важные заказчику, придется поискать

# Капитан очевидность

- Хороший программист быстро научится запускать модели с HuggingFace
- Хороший исследователь редко умеет хорошо программировать
- Оба не сделают ничего полезного без DevOps инженера

# Дополнительные материалы

- Rules of ml
- Hidden Technical Debt in Machine Learning Systems
- How to avoid machine learning pitfalls