Дизайн систем машинного обучения

2. Основы проектирования ML-систем

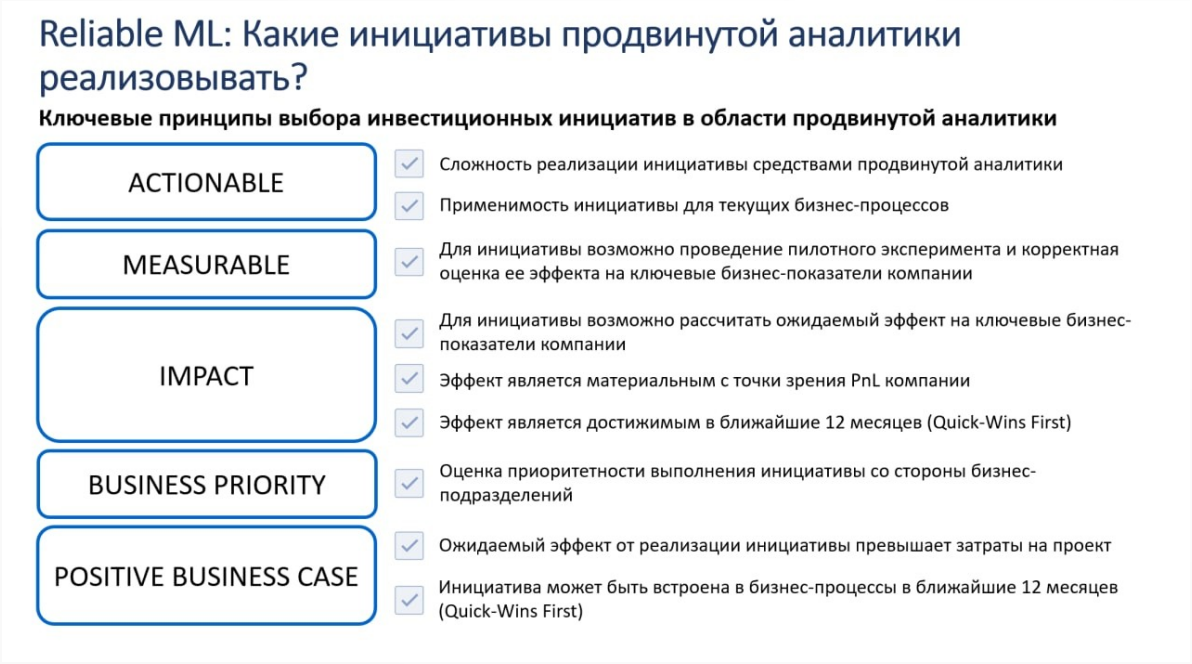
План курса

1) ~~Практическое применение машинного обучения~~
2) **Основы проектирования ML-систем — Вы находитесь здесь**
3) Обучающие данные
4) Подготовка и отбор признаков
5) Выбор модели, разработка и обучение модели
6) Оценка качества модели
7) Развертывание систем
8) Диагностика ошибок и отказов ML-систем
9) Мониторинг и обучение на потоковых данных
10) Жизненный цикл модели
11) Отслеживание экспериментов и версионирование моделей
12) Сложные модели: временные ряды, модели над графами
13) Непредвзятость, безопасность, управление моделями
14) ML инфраструктура и платформы
15) Интеграция ML-систем в бизнес-процессы

Погружаемся

- Бизнес-анализ — зачем?
- Дизайн системы — как?
- Проектные ограничения — чем?
- Юридические ограничения — можно ли?
- Технические ограничения — что придется использовать?
- Формулировка задачи ML — чему будем учить машину?

Стоит ли запускать проект



#reliable_ml

Инвестиционный цикл

#reliable_ml

Как проект может увеличить прибыль

- Явно:
  - увеличить продажи
  - снизить затраты
- Неявно:
  - увеличить удовлетворенность пользователей
  - увеличить удовлетворенность персонала
- Совсем неявно:
  - Увеличить инвестиционную привлекательность
  - Перенести операционные затраты в капитальные

Например, ML в интернет-магазине

- Как увеличить прибыль?
- Чего мы хотим от пользователя?
- Что мы помогаем ему сделать?
- Что мы мешаем ему сделать?
- Какая метрика надежнее?
- Какая метрика удобнее?

Метрики проекта

- Бизнес-метрики: влияние на ключевые показатели бизнеса
  - Прибыль
  - Доля рынка
  - Привлекательность для инвесторов
  - F1, ROC-AUC, Accuracy, Precision, Recall — интересно, но не нужно
- Инженерные метрики:
  - Задержка
  - Пропускная способность
  - F1, ROC-AUC, Accuracy, Precision, Recall — не всегда можем измерить

Прокси-метрика

- Если что-то не можем измерить
- Если не можем измерить в нужное время
- Дополнительные материалы:
  - Модель сломалась, выручка выросла
- Прокси-метрика из знаний о бизнесе
- Прокси-метрика из корреляций в данных
- Комбинировать ML и знания о бизнесе

https://habr.com/ru/company/retailrocket/blog/591205/

Дизайн систем машинного обучения

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

Надежность Reliability

- Свойство объекта сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных условиях применения, технического обслуживания, хранения и транспортирования →
- Как мы поймем, что все работает правильно?
- Как мы отличим ошибочный предикт от хорошего?
- Подробнее — в лекции про мониторинг.

Масштабируемость Scalability

- Способность системы, сети или процесса справляться с увеличением рабочей нагрузки (увеличивать свою производительность) при добавлении ресурсов (обычно аппаратных) →
- Если повезет, вам потребуется масштабируемость
- Растет трафик и объем данных
- Растет количество моделей и сценариев использования
- Обсудим в дальнейших лекциях

Обслуживаемость Maintainability

- Приспособленность к восстановлению работоспособного состояния после отказа или повреждения →
- Аварии серверов и сетей связи
- Проблемы с данными
- Проблемы с персоналом
- Обновление оборудования и схемы данных

Адаптируемость Adaptability

- Способность адаптироваться к меняющимся обстоятельствам →
- Меняются:
  - Бизнес-требования
  - Структура данных
  - Доступность данных
  - Оборудование
  - Инфраструктурные сервисы
  - Распределение данных (data shift, target drift)

Проектные ограничения

- Время
  - Примерно 20% на разработку решения, 80% на доводку
- Бюджет
  - Данные
  - Деньги
  - Оборудование
  - Люди

Купить немного времени

- Мощнее сервера — дороже, быстрее
- Больше людей на разметке — дороже, быстрее
- Больше людей в разработке — дороже, быстрее
- Купить существующее решение — дороже, быстрее

Compliance & Privacy

- Можем ли мы отдать данные на разметку?
- Можем ли мы хранить данные в облаке?
- Можем ли мы собирать данные пользователей?
- Обязаны ли мы шифровать данные?
- Можем ли мы использовать сторонние сервисы?
- Какие законы мы должны соблюдать?

Технические ограничения

- Проблема интеграции в существующие системы
  - Получить данные
  - Загрузить данные обратно
  - Превзойти качество существующей системы
- Использование имеющейся инфраструктуры
- Обучение пользователей и операторов системы

Этапы внедрения ML

- Решение без ML, если это возможно
- Простая ML модель
- Оптимизация простой модели
- Сложная ML модель

Упрощенная модель ML-разработки

1) Определение границ проекта
2) Подготовка данных
3) Разработка модели
4) Развертываение
5) Мониторинг и дообучение
6) Бизнес-анализ
7) Повторить

Взгляд исследователя на ML

- Дано:
  - Что у нас есть?
- Найти:
  - Что нужно оптимизировать?
- Критерий:
  - Как мы поймем, что нашли то, что нужно

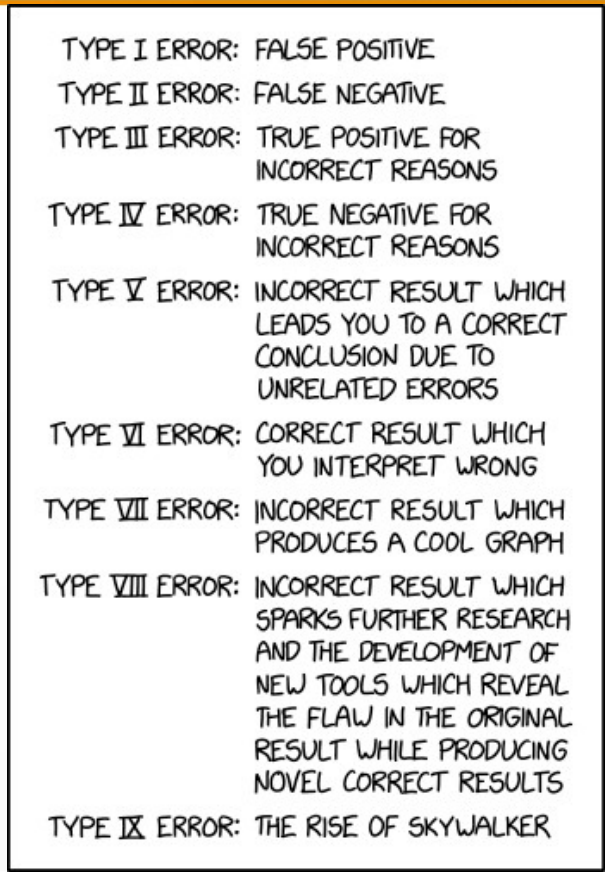
В индустрии все сложнее:

- Критерий надо найти и согласовать

ML-метрики

- Что будем измерять
- Как будем измерять
- Baseline — с чем сравнивать
- Цена ошибки 1 и 2 рода
- Требования к качеству
- Оценка достоверности
- Надежность, масштабируемость

https://www.explainxkcd.com/wiki/index.php/2303:_Error_Types



Baseline

- Прежде чем улучшать, найдите, с чем сравнить
  - Существующее на рынке решение
  - Простое решение на правилах
  - Качество решений человека
  - Решения конкурентов

Разная цена ошибки

- False positive
  - Нашли то, чего нет
- False negative
  - Не нашли то, что есть
- Как мы измерим цену ошибки?
- Сколько стоит убитый пациент?
- Как измерить удобство?
- Unknown Unknowns — не знаем, что это нужно измерять

Требования к качеству

- Самоуправляемый автомобиль:
  - Высокие требования к качеству
- Подсказки при вводе текста
  - Не такие высокие требования
- Рекомендации в интернет-магазине
  - Даже если совсем не то, все равно найдут поиском и купят
- Как формализовать?
  - Например, попробовать выразить в деньгах
  - Насколько вырастет прибыль, если увеличить accuracy на 1%

Итеративно ищем критерий

- Выбрать метрику
- Найти или собрать данные и разметку
- Подготовить признаки
- Обучить модель
- Найти ошибки в данных, переобучить модель
- Выгрузить модель, найти ошибки в метрике
- Вернуться к выбору метрики

Формулируем задачу для ML

- Одна и та же проблема может быть решена по-разному
- Например — нужно ускорить работу службы поддержки
- Классификация — отправлять обращение нужному специалисту
- Рекомендательная система — рекомендовать оператору варианты ответа
- Регрессия/классификация — оценивать важность пользователя и срочность его обращения
- Отвечать пользователю статьей из базы знаний — NER, информационный поиск, генерация текста, суммаризация и т. д.

Типичные ML задачи

- Классификация
  - Бинарная
  - Мультиклассовая
    - Низкой кардинальности
    - Высокой кардинальности
  - Многотемная классификация
- Регрессия
- Детекция объектов
- Детекция аномалий
- Извлечение структурированной информации
- Преобразование объектов

Classification vs regression

- Можем заменить регрессию классификацией, разбив диапазон на несколько классов
- Не делить на одинаковые диапазоны
- Лучше разбивать на квантили
  - pandas.qcut
- Или приводите распределение к равномерному

  sklearn.preprocessing.QuantileTransformer

Binary vs multiclass classification

- При малой кардинальности
  - one vs all
  - Последовательный chaining классификатор
- При большой — иерархические классификаторы
  - Сначала выбираем группу
  - Потом подгруппу
  - Потом элемент в группе
- При большой — нейронные сети

Multiclass vs multilabel classification

- Например, для каждого типа меток — отдельная бинарная классификация. Например, если у нас 3 возможных метки, у нас будет 3 модели.
- Или группировать встречающиеся вместе метки в один класс. Если у нас 3 возможных метки, у нас будет 8 моделей.
- В многотемной (multilabel) классификации сложно стратифицировать выборку и измерять качество
- Тут нейронные сети удобнее всего

Целевая функция

- Loss Function, оптимизируемый функционал
- Как правило, бизнес-метрики не могут быть использованы для оптимизации:
  - Недифференцируемые
  - Зависящие от внешних данных
  - Вычислительно дорогие
- https://paperswithcode.com/methods/category/loss-functions

Декомпозиция задачи

- Часто задачу можно разбить на подзадачи,
  каждая из которых будет решаться своей моделью
- Обычно это позволяет учить модель
  на меньших объемах данных
- Такие модели легче тестировать и отлаживать

Декомпозиция функции потерь

- Иногда мы хотим оптимизировать одновременно несколько показателей (продажи и прибыль)
- Мы можем использовать взвешенную сумму
  - A * loss1 + (1 - A) * loss2
- Мы можем ограничить какую-то из метрик
  - Минимизировать loss1, удерживая loss2 <= threshold

Дополнительные материалы

- The Netflix Recommender System:
  Algorithms, Business Value, and Innovation
- Trustworthy Online Controlled Experiments:
  Five Puzzling Outcomes Explained
- Top Challenges from the first
  Practical Online Controlled Experiments Summit