

# Требования к NLP-специалистам в индустрии

Мурат Апишев (mel-lain@yandex.ru)

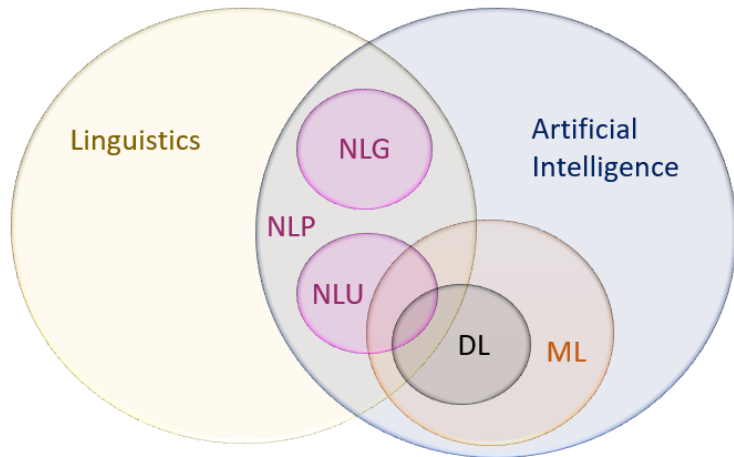
Февраль, 2022

# О чем пойдет речь

- ▶ Предметная область и профессия:
  - ▶ профессия NLP-специалиста, задачи
  - ▶ смежные специальности, отличия
- ▶ Требования к специалистам:
  - ▶ Стажер / джуниор
  - ▶ Мидл
  - ▶ Синьор
  - ▶ Тимлид
  - ▶ Развитие специалиста
- ▶ Тренды отрасли
- ▶ Вопросы из собеседований:
  - ▶ Примеры вопросов с ответами из различных секций
  - ▶ Прочие вопросы, типичные ошибки

# Профессия NLP-специалиста

- ▶ С развитием отрасли специализации становятся все уже и глубже
- ▶ Специалист в NLP решает различные задачи получения, анализа, валидации и обработки текстовых данных



# Задачи обработки естественного языка

Можно формулировать задачи с технической и продуктовой точек зрения

## Технические задачи:

- ▶ Классификация
- ▶ POS-теггинг
- ▶ Извлечение сущностей
- ▶ Разрешение кореференции
- ▶ Лемматизация
- ▶ Сегментация
- ▶ Разметка семантических ролей
- ▶ Векторизация
- ▶ Машинный перевод
- ▶ ...

## Продуктовые задачи:

- ▶ Анализ тональности
- ▶ Машинный перевод
- ▶ Генерация подписей к изображениям
- ▶ Генерация сниппетов для новостей
- ▶ Ведение диалога
- ▶ Исправление опечаток
- ▶ Поисковое и рекомендательное ранжирование
- ▶ Анализ трендов
- ▶ ...

# Что нужно работодателю чаще всего

- ▶ Решать относительно простые и понятные задачи:
  - ▶ классификация (анализ тональности)
  - ▶ выделение сущностей (адреса, даты)
  - ▶ кластеризация (выделение интенгов из логов)
  - ▶ трендовая аналитика
  - ▶ чат-боты
- ▶ Применять текстовые модели для задач предиктивной аналитики
- ▶ Создавать специализированные решения:
  - ▶ чит-чаты
  - ▶ машинный перевод
  - ▶ парафразирование
  - ▶ суммаризация
- ▶ В разных ситуациях как создавать собственное решение, так и искать + тестировать + внедрять готовое

## Смежные специальности

- ▶ Разработчик ПО на Python/C++/Java/...
- ▶ Data-инженер
- ▶ MLops
- ▶ ML-инженер
- ▶ Специалист по машинному обучению
- ▶ Специалист по анализу данных
- ▶ Аналитик
- ▶ Product-менеджер
- ▶ Project-менеджер

# Требования к NLP-специалистам

- ▶ Разнообразии задач и навыков порождает разнообразие вакансий и требований
- ▶ В ряде случаев работодатели не до конца представляют, какие именно задачи предстоит решать, и пишут в описание вакансии все подряд
- ▶ Иногда требования намеренно завышаются, что выбрать наиболее сильных и универсальных кандидатов
- ▶ Иногда, наоборот, описываются общими словами с минимальной конкретикой в описании инструментов и навыков
- ▶ Общеприняты три основных грейда: джуниор (junior), мидл (middle) и сеньор (senior)
- ▶ В разных компаниях их понимание может отличаться радикально, но можно вывести некое усредненное представление

# Базовые обязательные требования

- ▶ Технический английский язык (как минимум на уровне чтения)
- ▶ Базовая математическая подготовка:
  - ▶ линейная алгебра
  - ▶ математический анализ
  - ▶ теория вероятностей
  - ▶ математическая статистика
  - ▶ методы оптимизации
- ▶ Навыки программирования (как минимум на Python, в т.ч. в Jupyter Notebook)
- ▶ Базовые навыки работы в Unix/Linux-системах
- ▶ Умение самостоятельно искать нужную информацию в интернете



# Требования к Junior

- ▶ Понимание постановок основных задач
- ▶ Умение работать с данными (поиск, предобработка, подсчёт статистик, выявление особенностей, поиск аномалий)
- ▶ Умение использовать методы ML и не-нейросетевые подходы (регулярные выражения, словари, правила, анализаторы)
- ▶ Понимание основных DL-архитектур, умение воспользоваться готовыми решениями по туториалам
- ▶ Понимание технических метрик качества и применимости их в различных задачах
- ▶ Навыки работы с репозиториями (версионирование, код ревью, CI)
- ▶ Желательно: владение основами компьютерной лингвистики

# Требования к Middle

- ▶ Опыт решения реальных промышленных задач (хотя бы из базовых)
- ▶ Уверенное понимание основных задач в индустрии и подходов к их решению
- ▶ Использование DL-подходов, обучение и дообучение собственных моделей
- ▶ Понимание современных DL-архитектур, принципов их работы, преимуществ и недостатков
- ▶ Понимание основ статистической проверки гипотез и А/Б-тестов
- ▶ Умение работать с краудсорсингом и ассессорами
- ▶ Уверенные навыки разработки на Python, владение ООП
- ▶ Знания об основных алгоритмах и структурах данных, их устройстве и сложностях
- ▶ Желательно: опыт разработки на не-скриптовых языках

# Требования к Senior

- ▶ Многократный опыт решения разнообразных NLP-задач
- ▶ Глубокое понимание используемых подходов и моделей, как ML, так и DL
- ▶ Понимание предметной области, в которой ведется работа
- ▶ Умение для поставленной верхнеуровневой задачи сразу выстраивать поэтапный план ее решения, выбирать оптимальные гипотезы для проверки и оценивать предполагаемые результаты
- ▶ Понимание как технических, так и бизнес-метрик, умение выбирать нужные метрики и сопоставлять их друг с другом
- ▶ Привычка регулярно читать статьи о новых исследованиях и достижениях в области NLP, доносить их до коллег
- ▶ Желательно: умение взаимодействовать не только с коллегами из команды, но и со смежными специалистами, в т.ч. не техническими
- ▶ Желательно: иметь экспертизу в смежных областях (CV, RL, предиктивная аналитика)

# Требования к Team Lead

- ▶ Умение организовать работу команды, состоящей из людей с различными уровнями и наборами компетенций
- ▶ Развитые софт-скиллы при общении с коллегами и с внешними контрагентами
- ▶ Умение представлять результаты работы команды перед вышестоящим начальством/заказчиками на понятном им языке и, наоборот, переводить их запросы в конкретные задачи для подчинённых
- ▶ Умение оценить перспективность решения/продукта «на берегу» с разных сторон
- ▶ Понимание устройства технических и бизнес процессов в компании хотя бы в рамках круга потребностей и задач своей команды

# Направления роста NLP-специалиста

- ▶ За счёт широкого набора кругозора можно выбирать разные направления, например, переход
  - ▶ в исследовательские подразделения
  - ▶ в смежную DS/ML область
  - ▶ в MLOps
  - ▶ в разработку, в т.ч. на низкоуровневых языках
  - ▶ на руководящие посты
  - ▶ в project/product-менеджеры
  - ▶ в собственный бизнес
- ▶ Выбор не окончательный и может меняться многократно
- ▶ Иногда специалист может освоить сразу несколько направлений и одновременно решать в компании широкий спектр задач

# Текущий статус индустрии обработки текстов

- ▶ В отличие от CV, в NLP после DL-революции классические подходы не потеряли полностью своей актуальности
- ▶ На локальном рынке много компаний, которые успешно используют подобные методы для решения базовых задач (напр. классификации)
- ▶ Но глобально лидирует подход Transfer Learning на основе больших предобученных моделей Transformer
- ▶ Архитектура Transformer из NLP оказалась настолько успешной, что активно модели на ее основе используются в CV и обработке сигналов
- ▶ В 2021 году мировой тренд на рост моделей развернулся:
  - ▶ модели становятся меньше в десятки раз
  - ▶ показатели качества остаются аналогичными за счет добавления возможности работы с базами данных для поиска нужной информации
  - ▶ пример: модель RETRO от DeepMind

## Прогнозы на ближайшее будущее

Исходя из текущих тенденций, на ближайшее будущее намечается несколько трендов (список далеко не исчерпывающий):

- ▶ активное внедрение мультязычных моделей, где перенос обучения позволяет уменьшить требования к объему данных на менее распространенных языках
- ▶ массовое распространение чат-ботов разного уровня в большом числе бизнесов, демократизация процесса их создания
- ▶ рост возможностей и большая персонализация сервисов типа голосовых ассистентов
- ▶ развитие интеграции NLP с технологиями виртуальной и дополненной реальности

# Типы собеседований для NLP-специалистов

- ▶ Кодовая секция:
  - ▶ придумать алгоритм решения задачи
  - ▶ реализовать его на одном из подходящих языков программирования
- ▶ ML-секция:
  - ▶ ответить на набор вопросов про ML и DS
  - ▶ решить задачи на теорию вероятностей, логику или комбинаторику
  - ▶ часто проходит в формате «Представьте, что нужно создать такой-то сервис, как будете подходить к решению задачи?»
- ▶ DL/NLP-секция:
  - ▶ ответить на набор вопросов про DL, архитектуры, обучение
  - ▶ ответить на набор вопросов про нейросетевые модели (в т.ч. NLP)
  - ▶ часто проходит в формате «Расскажите про задачи, которые решали, чем пользовались?»



# Кодовая секция. Вопрос 1

## ▶ Вопрос:

- ▶ Какая структура данных лежит в основе контейнера `set` в Python?
- ▶ Какова сложность операции добавления элемента?
- ▶ Как еще можно реализовать такой контейнер?

# Кодовая секция. Вопрос 1

## ▶ Вопрос:

- ▶ Какая структура данных лежит в основе контейнера `set` в Python?
- ▶ Какова сложность операции добавления элемента?
- ▶ Как еще можно реализовать такой контейнер?

## ▶ Ответ:

- ▶ В основе лежит хэш-таблица
- ▶ Сложность добавления в обычном случае  $O(1)$ , в худшем —  $O(n)$
- ▶ Реализовать можно с помощью бинарного дерева поиска

## Кодовая секция. Вопрос 2

- ▶ **Задача:** для данной строки найти длину самой большой подпоследовательности из одного и того символа  
(aaadddaaabbbb -> 4)

## Кодовая секция. Вопрос 2

- ▶ **Задача:** для данной строки найти длину самой большой подпоследовательности из одного и того же символа (aaadddaaabbbb -> 4)
- ▶ **Решение:**
  - ▶ Пройти по всей строке
  - ▶ Запоминать на каждом шаге длину текущего повторяющегося символа и длину самого длинного повторения
  - ▶ Обновлять наибольшую длину по мере необходимости
- ▶ **Комментарий:** это простая задача, но ее можно немного усложнить: то же условие, но символов два (aaadddaaabbbb -> 9)

## Кодовая секция. Вопрос 3

- ▶ **Задача:** написать декоратор функции на Python, который печатает число раз, которое эта функция была вызвана

## Кодовая секция. Вопрос 3

- ▶ **Задача:** написать декоратор функции на Python, который печатает число раз, которое эта функция была вызвана
- ▶ **Решение:**

```
1 def print_num_calls(f):  
2     i = 0  
3     def _f(*args, **kwargs):  
4         nonlocal i  
5         print(i := i + 1)  
6         return f(*args, **kwargs)  
7     return _f
```

## Кодовая секция. Вопрос 4

- ▶ **Вопрос:** является ли Python языком со строгой типизацией или нет?

## Кодовая секция. Вопрос 4

- ▶ **Вопрос:** является ли Python языком со строгой типизацией или нет?
- ▶ **Ответ:**
  - ▶ Да, Python – строго типизированный язык
  - ▶ Важно не путать объекты и переменные, которые являются ссылками на объекты

```
1 a = 1
2 a = 's'
```

- ▶ Переменные могут легко присваиваться разным объектам
- ▶ Объект же имеет один тип и не меняет его на протяжении всего жизненного цикла



# ML секция. Вопрос 1

- ▶ **Вопрос:** представим, что требуется создать сервис для кредитного скоринга
  - ▶ Как подходить к решению этой задачи?
  - ▶ Какие данные могут понадобиться?
  - ▶ Как будет измеряться качество решения?

# ML секция. Вопрос 1

- ▶ **Вопрос:** представим, что требуется создать сервис для кредитного скоринга
  - ▶ Как подходить к решению этой задачи?
  - ▶ Какие данные могут понадобиться?
  - ▶ Как будет измеряться качество решения?
- ▶ **Ответ:**
  - ▶ Общая идея: собираем исторические данные и обучаем ML-модель
  - ▶ Объемы кредитов, признаки заемщиков, успешность возвращения
  - ▶ В простейшем случае доля верных предсказаний

## ML секция. Вопрос 2

- ▶ **Вопрос:** какие именно признаки заемщиков могут оказаться полезными?

## ML секция. Вопрос 2

- ▶ **Вопрос:** какие именно признаки заемщиков могут оказаться полезными?
- ▶ **Ответ:**
  - ▶ Объем средств на счетах
  - ▶ Суммарный объем и число кредитов
  - ▶ Регулярность и объем пополнений счетов
  - ▶ Город и район прописки/проживания
  - ▶ Пол, возраст
  - ▶ ...

## ML секция. Вопрос 3

- ▶ **Вопрос:** в данном случае у нас задача бинарной классификации, поэтому можно измерять долю верных ответов. А в каких случаях эта метрика качества плоха? Какие еще есть метрики для этой задачи? А для регрессии?

## ML секция. Вопрос 3

- ▶ **Вопрос:** в данном случае у нас задача бинарной классификации, поэтому можно измерять долю верных ответов. А в каких случаях эта метрика качества плоха? Какие еще есть метрики для этой задачи? А для регрессии?
- ▶ **Ответ:**
  - ▶ Доля верных ответов для классификации подойдет плохо, если выборка существенно не сбалансирована
  - ▶ Альтернатива — точность/полнота и их гармоническое среднее (F-мера)
  - ▶ Для регрессии есть множество метрик, наиболее популярные MAE, MSE, RMSE
- ▶ **Комментарий:** обычно еще требуется расписать эти метрики и объяснить, чем одна отличается от другой

## ML секция. Вопрос 4

- ▶ **Вопрос:** было сказано об ML-моделях. Какие модели стоит попробовать использовать?

## ML секция. Вопрос 4

- ▶ **Вопрос:** было сказано об ML-моделях. Какие модели стоит попробовать использовать?
- ▶ **Ответ:**
  - ▶ В первую очередь стоит обучить лог-регрессию, это простой и сильный бейзлайн
  - ▶ Сильным решением получится градиентный бустинг над решающими деревьями
- ▶ **Комментарий:** обязательно спросят про то, что представляют собой оба подхода



## ML секция. Вопрос 5

- ▶ **Вопрос:** что такое «регуляризация»? Какие виды регуляризации и используются при обучении лог-регрессии и почему? К какому эффекту они приводят?

## ML секция. Вопрос 5

- ▶ **Вопрос:** что такое «регуляризация»? Какие виды регуляризации и используются при обучении лог-регрессии и почему? К какому эффекту они приводят?
- ▶ **Ответ:**
  - ▶ Регуляризация ограничивает область возможных значений весов модели
  - ▶ В лог-регрессии используются математические подходы ( $L_1$ ,  $L_2$  и ElasticNet) и ранний останов
  - ▶ В отсутствие регуляризации лог-регрессия сильно переобучается, устремляя все веса в  $\pm\infty$
  - ▶  $L_1$ -регуляризация зануляет часть весов,  $L_2$  делает все веса небольшими по модулю
  - ▶  $L_2$ -регуляризация лучше справляется с коррелированными признаками (веса распределяются пропорционально, а не случайно, как в  $L_1$ )

## ML секция. Вопрос 6

- ▶ **Вопрос:** каким образом можно распараллелить обучение градиентного бустинга?

## ML секция. Вопрос 6

- ▶ **Вопрос:** каким образом можно распараллелить обучение градиентного бустинга?
- ▶ **Ответ:**
  - ▶ Градиентный бустинг нельзя распараллелить по деревьям
  - ▶ Но можно распараллелить по ядрам процесс обучения одного дерева, например, по признакам внутри одного узла дерева

## ML секция. Вопрос 7

- ▶ **Вопрос:** Предположим, что модель была обучена и отправлена в работу. Как проверить, что она работает лучше, чем люди?

## ML секция. Вопрос 7

- ▶ **Вопрос:** Предположим, что модель была обучена и отправлена в работу. Как проверить, что она работает лучше, чем люди?
- ▶ **Ответ:**
  - ▶ В идеале нужно провести А/Б-тест, выделив репрезентативные выборки под оба метода
  - ▶ С учётом того, что кредит будет выплачиваться долго, можно пробовать более краткосрочные метрики, например, просрочку по первым 2-3 платежам
- ▶ **Комментарий:** скорее всего спросят про то, как сравнить результаты двух подходов с помощью статистических критериев

# NLP секция. Вопрос 1

- ▶ **Вопрос:** из чего состоит чат-бот?

# NLP секция. Вопрос 1

- ▶ **Вопрос:** из чего состоит чат-бот?
- ▶ **Ответ:**
  - ▶ Большинство чат-ботов состоит из классификатора интенгов, NER и сценарного графа
  - ▶ В случае чит-чата могут использоваться генеративные и IR-модели
- ▶ **Комментарий:** здесь могут уточнить про разницу между NER и slot-filling, а также про детали работы сценарного графа



## NLP секция. Вопрос 2

- ▶ **Вопрос:** предположим, что ставится задача классификации текстов, и выдана выборка размеченных текстов. Что будете делать с данными до обучения моделей?

## NLP секция. Вопрос 2

- ▶ **Вопрос:** предположим, что ставится задача классификации текстов, и выдана выборка размеченных текстов. Что будете делать с данными до обучения моделей?
- ▶ **Ответ:**
  - ▶ Первый шаг – анализ. Подсчет основных статистик по объектам и классам, анализ словаря
  - ▶ Второй шаг – предобработка и генерация признаков
- ▶ **Комментарий:** имеет смысл уточнить, какие именно статистики и зачем считаются, когда нужна предобработка, какие признаки можно генерировать

## NLP секция. Вопрос 3

- ▶ **Вопрос:** предположим, что для классификации текстов есть только неразмеченная выборка. Как можно получить разметку?

## NLP секция. Вопрос 3

- ▶ **Вопрос:** предположим, что для классификации текстов есть только неразмеченная выборка. Как можно получить разметку?
- ▶ **Ответ:**
  - ▶ самостоятельная ручная разметка (обязательна!)
  - ▶ разметка с помощью правил, словарей и регулярных выражений
  - ▶ разметка с помощью краудсорсинга (например, Яндекс.Толока)
  - ▶ для краудсорсинга может потребоваться активное обучение, его можно сделать с помощью простой модели
  - ▶ можно использовать аугментацию данных (синонимы, замены/удаления, SMOTE)
- ▶ **Комментарий:** в случае задачи с большим числом классов может потребоваться последовательное повторение шагов для разных уровней агрегации классов

## NLP секция. Вопрос 4

- ▶ **Вопрос:** рассказать про постановку задачи NER и существующие подходы к ее решению

## NLP секция. Вопрос 4

- ▶ **Вопрос:** рассказать про постановку задачи NER и существующие подходы к ее решению
- ▶ **Ответ:**
  - ▶ NER — задача извлечения именованных сущностей (адреса, телефоны, геолокации, имена, числа)
  - ▶ Базовый метод решения — правила, словари и регулярные выражения
  - ▶ В нейросетевой постановке это задача разметки последовательности, раньше ее решали с помощью RNN, теперь основной подход – модели на основе кодировщика трансформера

## NLP секция. Вопрос 5

- ▶ **Вопрос:** как устроена сверточная сеть для классификации текстов?

## NLP секция. Вопрос 5

- ▶ **Вопрос:** как устроена сверточная сеть для классификации текстов?
- ▶ **Ответ:**
  - ▶ Для классификации текстов обычно используют однослойные CNN с одномерными свертками
  - ▶ Вход сети — матрица из эмбеддингов (слово за словом)
  - ▶ Одномерные свертки разных размеров двигаются по ней сверху вниз (вторая размерность такая же, как у эмбеддингов)
  - ▶ На выходе слоя набор векторов-карт признаков разного размера
  - ▶ К ним применяется пулинг для получения итогового вектора фиксированного размера
  - ▶ Этот вектор передается в полносвязные слои и softmax
- ▶ **Комментарий:** можно рассказать про смысл использования сверток разных размеров и типы пулинга



## NLP секция. Вопрос 6

- ▶ **Вопрос:** в чём заключается преимущество модели векторных представлений FastText перед word2vec или GloVe?

## NLP секция. Вопрос 6

- ▶ **Вопрос:** в чём заключается преимущество модели векторных представлений FastText перед word2vec или GloVe?
- ▶ **Ответ:**
  - ▶ FastText одновременно с векторами слов и словарных n-грамм обучает векторы символьных n-грамм
  - ▶ Это позволяет получать векторы для OOV-слов с помощью суммы векторов входящих в них символьных n-грамм
  - ▶ В более ранних моделях проблема OOV обычно решалась введением UNK-токена, что существенно хуже
  - ▶ Также у FastText есть очень эффективная параллельная реализация на CPU

## NLP секция. Вопрос 7

- ▶ **Вопрос:** как устроен слой self-attention в архитектуре Transformer? Есть ли в нем нелинейность?

## NLP секция. Вопрос 7

- ▶ **Вопрос:** как устроен слой self-attention в архитектуре Transformer? Есть ли в нем нелинейность?
- ▶ **Ответ:**
  - ▶ Слой параметризован тремя весовыми матрицами
  - ▶ Каждый вектор входной последовательности перемножается со всеми тремя матрицами
  - ▶ Для каждого токена получается три вектора: ключи, запросы и значения
  - ▶ Набор коэффициентов для токена получается умножением его вектора запроса на все векторы ключей (+ softmax и нормировка)
  - ▶ Итоговый вектор для токена получается как взвешенная с этими коэффициентами сумма векторов значений
  - ▶ Нелинейности в нем нет (она есть дальше после полносвязного слоя)

## NLP секция. Вопрос 8

- ▶ **Вопрос:** Как в слое self-attention использовать авторегрессионность?

## NLP секция. Вопрос 8

- ▶ **Вопрос:** Как в слое self-attention использовать авторегрессионность?
- ▶ **Ответ:** Достаточно занулять коэффициенты, связанные с маскированными токенами

## NLP секция. Вопрос 9

- ▶ **Вопрос:** Какое потребление памяти на этапе инференса у RNN и BERT?

## NLP секция. Вопрос 9

- ▶ **Вопрос:** Какое потребление памяти на этапе инференса у RNN и BERT?
- ▶ **Ответ:**
  - ▶ RNN –  $O(1)$
  - ▶ BERT –  $O(n^2)$ ,  $n$  – длина последовательности
- ▶ **Комментарий:** в стандартном self-attention нужно хранить матрицу коэффициентов для каждой пары токенов входа



## NLP секция. Вопрос 10

- ▶ **Вопрос:** в чем главные идеи моделей ALBERT и ELECTRA?

## NLP секция. Вопрос 10

- ▶ **Вопрос:** в чем главные идеи моделей ALBERT и ELECTRA?
- ▶ **Ответ:**
  - ▶ Модель ALBERT экономит память за счет разделения весов между блоками кодировщика трансформера
  - ▶ В модели ELECTRA (в основе – BERT) реализуется идея GAN: обучаются две Transformer-подобные модели, генератор и дискриминатор
  - ▶ Генератор – небольшая модель, вставляющая на место токенов [MASK] последовательности свои варианты
  - ▶ Дискриминатор – основная модель, которая учится предсказывать, какие токены являются замененными
  - ▶ Такая процедура обучения позволяет улучшить качество предобучения по сравнению с обычным MLM

## Какие еще бывают вопросы

- ▶ Есть важные вопросы, не касающиеся напрямую технических навыков, с ростом грейда их количество растет:
  - ▶ Есть ли опыт руководства командой, какой численности?
  - ▶ Как была устроена работа в командах, какие процессы?
  - ▶ Был ли опыт построения решения end-to-end — от бизнес постановки до внедрения и сопровождения? Если был, то какой?
  - ▶ Есть ли опыт взаимодействия с внутренними или внешними бизнес-заказчиками?
  - ▶ Какой самый интересный/сложный проект удавалось реализовать в роли исполнителя/руководителя?

## Какие ошибки встречаются на NLP-собеседованиях

- ▶ Отсутствие части компетенций (математика, программирование)
- ▶ Написание существенно неоптимального кода при решении задач
- ▶ Перекос знаний и навыков в сторону DL (или, наоборот, в сторону классических подходов и моделей)
- ▶ Kaggle-подход: игнорирование при проектировании решения технических проблем, связанных с будущей эксплуатацией решения
- ▶ непонимание того, что работа с данными является часто важнее и затратнее непосредственного обучения моделей
- ▶ Неумение описать свой вклад в сделанные ранее проекты