



МЕЖДУНАРОДНАЯ ОЛИМПИАДА
ПО ПРОМЫШЛЕННОЙ РАЗРАБОТКЕ



СКОЛЬКО ТЫ ЗАРАБАТЫВАЕШЬ?

Задача

Нужно сделать сервис, который по данным резюме показывает, сколько человек потенциально может зарабатывать на рынке (вилка дохода).

Сервис также должен подсказать, как улучшить резюме, чтобы увеличить эту вилку.

Идея проста:

Заполнил резюме → получил вилку дохода → сделал правки → получил новую вилку.

Проблема

Многие люди не знают, на какую зарплату они могут рассчитывать.

Резюме часто написаны плохо: нет нужных навыков, мало конкретики, опыт описан невнятно.

Из-за этого:

- человек просит меньше, чем стоит;
- или наоборот, сильно завышает ожидания;
- а резюме выглядит слабым.

Задача кейса — помочь человеку понять свою рыночную стоимость и улучшить резюме.

Как считается доход

Если команда без ML

Мы даем API — вы отправляете туда данные, API возвращает вилку дохода.

Если команда с ML

Сами тренируете модель и выдаете вилку самостоятельно.

Функциональные требования (MVP)

Frontend/Mobile

- Страница с формой резюме.
- Экран результата с вилкой дохода.
- Список рекомендаций.
- Кнопка «Пересчитать».

Backend

- Приём формы.
- Поиск подходящих вакансий в датасете.
- Вызов API или ML-модели.
- Генерация рекомендаций.
- Возврат результата фронтенду.

ML

- Предсказание вилки.

Основной пользовательский сценарий

Сценарий 1 — первая оценка

Пользователь заполняет роль, опыт, навыки и регион.

Нажимает «Оценить».

Сервис ищет похожие вакансии, вызывает API/модель и показывает вилку дохода + рекомендации.

Сценарий 2 — улучшение

Пользователь добавляет недостающий навык или уточняет опыт.

Нажимает «Пересчитать».

Вилка растёт, рекомендации обновляются.

Пользователь делает несколько итераций, пока результат его не устроит.



МЕЖДУНАРОДНАЯ ОЛИМПИАДА ПО ПРОМЫШЛЕННОЙ РАЗРАБОТКЕ



Тариф, я выбираю тебя!

Задача

Нужно сделать сервис, который помогает выбрать лучший вариант доставки из Москвы в любой город.

Сервис должен:

- показывать разные типы доставки («Экспресс», «Сборный груз», «Терминал-Дверь» и т.п.);
- рассчитывать цену для нужного веса;
- учитывать сроки доставки;
- подсказывать, какие данные были восстановлены (если в датасете чего-то не хватало);
- показывать лучший вариант: дешевле, быстрее или «баланс».

Выбираешь город и вес → получаешь варианты доставки → выбираешь лучший.

Проблема

У транспортных компаний разные сайты, разные тарифы и разные способы показывать цены.

Из-за этого:

- сложно понять, сколько реально стоит доставка;
- непонятно, какой тип доставки выбрать;
- данные часто неполные — где-то нет цены для нужного веса, где-то нет сроков;

- приходится вручную проверять 3–5 сайтов, тратить много времени и всё равно легко запутаться.

Задача кейса — собрать все варианты доставки в одном месте, аккуратно восстановить недостающие данные и помочь человеку сделать лучший выбор.

Функциональные требования (MVP)

Frontend/Mobile

- Поле выбора города.
- Поле ввода веса.
- Кнопка «Найти тариф».
- Экран результата:
 - список тарифов,
 - цена для указанного веса,
 - сроки доставки,
 - тип тарифа,
 - метка `real/restored`,
 - стратегия сортировки.

Backend

- Прием данных.
- Поиск подходящих строк в датасете.
- Расчет цены.
- Восстановление пропусков.
- Сортировка по стратегии.
- Возврат результата.

ML

- Предсказание стоимости или сроков.

Пользовательские сценарии

Сценарий 1 — первая подборка

Пользователь выбирает город и вводит вес.

Нажимает «Найти тариф».

Сервис показывает:

- варианты доставки разных компаний,
- стоимость именно для его веса,
- сроки доставки,
- типы тарифов (Экспресс / Сборный груз),
- какие данные реальные, а какие спрогнозированные.

Он выбирает лучший вариант.

Сценарий 2 — другой вес или другой город

Пользователь меняет вес или город.

Нажимает «Пересчитать».

Сервис снова считает стоимость, сроки и сортирует все по стратегии.

Пользователь сравнивает и делает информированный выбор.



МЕЖДУНАРОДНАЯ ОЛИМПИАДА
ПО ПРОМЫШЛЕННОЙ РАЗРАБОТКЕ



Базовые критерии Product (10 баллов)

A. Проблема и пользователь (2 балла)

- A1. Проблема, которую продукт решает для пользователя сформулирована ясно и конкретно. (Да/Нет)
- A2. Участники конкретно понимают кто пользователь продукта. Понимают как продукт решает проблему пользователя. (Да/Нет)

B. Сценарии и соответствие реализации (2 балла)

- B1. Участники отрисовали и презентовали пользовательский путь. Проработали основные User-Story. (Да/Нет)
- B2. Реализованный продукт полностью поддерживает основные сценарии. (Да/Нет)

C. Ценность и приоритизация (3 балла)

- C1. Посчитана и обоснована бизнес-ценность реализованного продукта (Да/Нет)
- C2. Реализован план разработки продукта с обоснованием приоритетов (Да/Нет)
- C3. Реализованная функциональность позволяет пройти основной пользовательский путь. (Да/Нет)

D. Понятность UX/UI (2 балла)

- D1. Реализованный продукт очевиден и понятен пользователю. Не требуется дополнительных пояснений, чтобы пройти основной пользовательский путь. (Да/Нет)

D2. Визуальный стиль и продукт соответствует времени и выполняемым задачам (Да/Нет)

E. Метрики и проверка гипотез (2 балла)

E1. Озвучена и посчитана NordStar (главная) метрика продукта. (Да/Нет)

E2. Команда назвала и объяснила хотя бы одну метрику успеха реализованного продукта и способ проверки (MVP/тест/интервью). (Да/Нет)

