Семинар 3

Бизнес-аналитика и ИИ как инструмент эффективного управления

Что сегодня будем делать

- Познакомимся с машинным обучением
- Поймем как выглядит общая постановка задачи для модели
- Обучим простые модели

Ключевые выводы EDA

1. **Общие закономерности:**
   что мы узнали?
2. **Проблемные места:**
   выбросы, пропуски, дисбаланс классов?
3. **Направления для ML:**
   какие гипотезы проверять, какие модели могут быть релевантны
4. **Связь с бизнесом:**
   какие решения можно принять на основе этих выводов

EDA хорош для

- *ответа на вопрос «Что происходит в данных?»*
- *понимания текущей ситуации и формулировки гипотез*
- *анализа простых зависимостей и ярких инсайтов*

Но

- *EDA описывает прошлое, а не предсказывает будущее*
- *тяжело учесть десятки признаков и их взаимодействия «на глаз»*
- *решения часто остаются ручными и субъективными*

Машинное обучение – это

Способ научить алгоритм находить закономерности в исторических данных
и использовать их для прогнозов и рекомендаций

Примеры вокруг нас

- *Системы кредитного рейтингования в банках*
- *Рекомендации товаров/фильмов*
- *Динамическое ценообразование*
- *и еще очень много всего*

Как выглядят данные

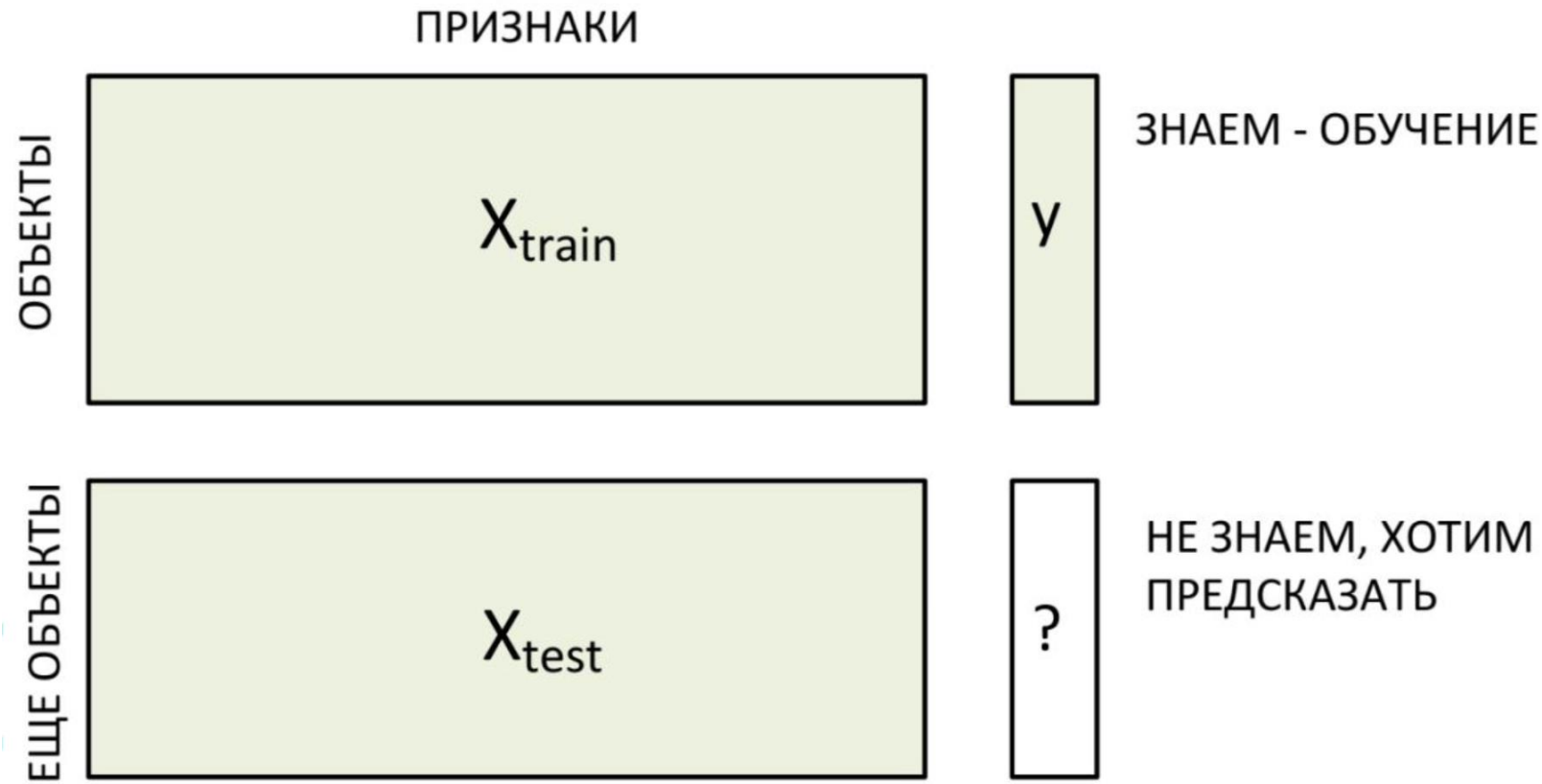
Имеющиеся признаки/features
(по столбцам)

Имеющиеся наблюдения
(по строкам)

	Признак 1	Признак 2	…	Признак m	Target
Наблюдение 1	…	…		…	…
Наблюдение 2	…	…		…	…
…	…	…		…	…
Наблюдение n	…	…		…	…

Целевая переменная,
которую хотим
предсказывать

Как выглядят данные



Базово: **нельзя** оценивать модель на тех же данных, на которых она обучалась

Основные типы признаков

- Числовые
  - Доход
  - Кол-во покупок
  - Сумма чека
  - …
- Категориальные
  - Тип устройства (IOS / Android / Web)
  - Город
  - Канал привлечения
  - …
- Бинарные
  - Курильщик / не курильщик
  - …
- Временные
  - Дата регистрации
  - Время последнего визита
  - …

Важно:

1. Не все признаки одинаково полезны
2. Некоторые признаки нельзя использовать
3. Тип признака влияет на сложность внедрения и качество модели

2 основных типа задач в ML

Регрессия

Хотим предсказывать число

Примеры:

Прогноз выручки на следующий месяц

Прогноз LTV клиента

Прогноз стоимости страховки

Прогноз времени доставки

Классификация

Хотим отнести объект к одному из классов

Примеры:

Уйдет клиент или останется

Фрод / Не фрод

Откликнется на оффер или нет

Сегмент клиента

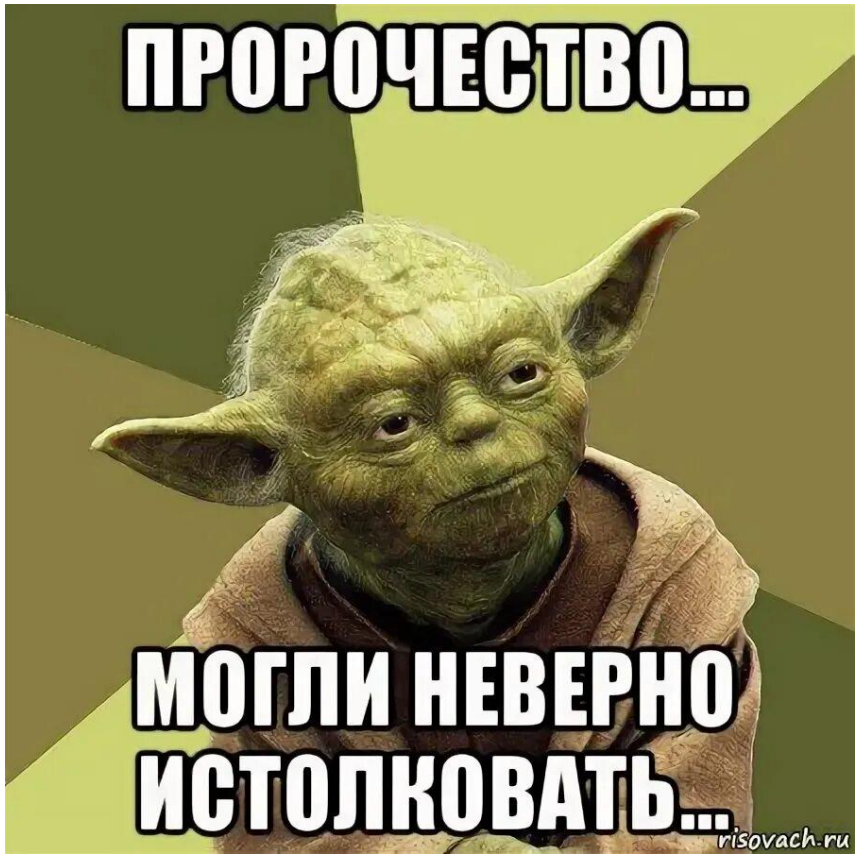
Регрессия

Определение:
прогнозирование числового значения (пример: будущие продажи)

Примеры в бизнесе:

1. Прогнозирование спроса, выручки, доходности
2. Оценка стоимости недвижимости

Основные метрики
MAE (Mean Absolute Error),
MSE (Mean Squared Error), RMSE и т.д.



https://education.yandex.ru/handbook/ml

Классификация

Определение:
прогнозирование категории (пример: понравится фильм или нет)

Примеры в бизнесе:

1. Фрод-мониторинг (мошенничество)
2. Отток клиентов (churn)
3. Сегментация по отклику на рекламную кампанию

Основные метрики
Acuracy, Precision, Recall, F1, ROC-AUC

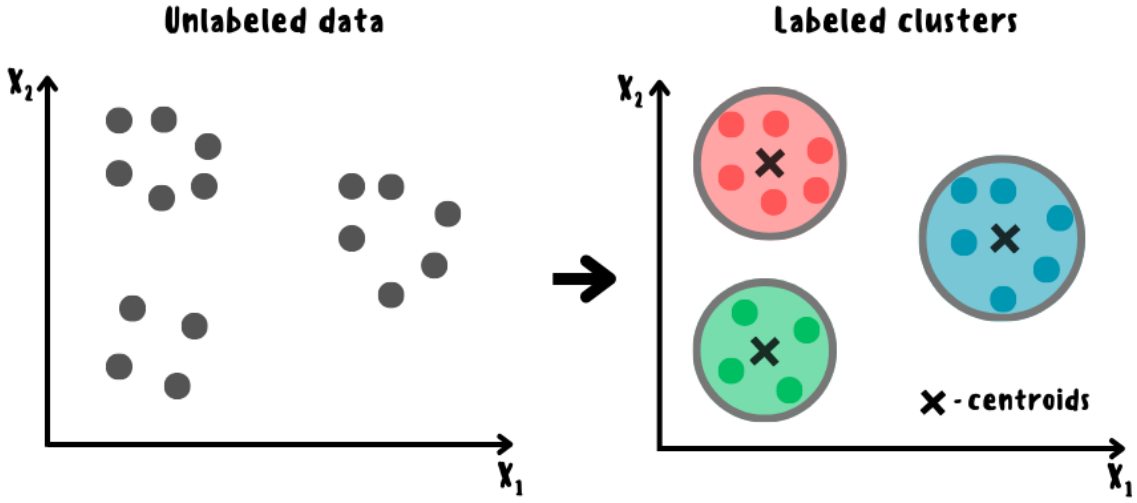
https://education.yandex.ru/handbook/ml

Кластеризация

Определение:
группировка объектов по схожести признаков (пример: сегментация клиентов)

Примеры в бизнесе:

1. Сегментация клиентов для персонализированных предложений каждой группе
2. Группировка товаров по схожести спроса
3. Анализ аномалий в транзакциях



Основные метрики
Внутрикластерное расстояние (Inertia)
Коэффициент силуэта (Silhouette Score)
и т.д.

https://education.yandex.ru/handbook/ml

Выбор подхода под бизнес-задачу

Связь с бизнес-целями:
прогнозирование числового значения (пример: будущие продажи)

Примеры в бизнесе:

1. Нужно предсказать факт (да/нет)?
   → Классификация
2. Нужно предсказать число (продажи)?
   → Регрессия
3. Нужно сгруппировать клиентов?
   → Кластеризация

Важно:
всегда начинаем с вопроса “зачем?”,
а не с “какой метод круче”

https://education.yandex.ru/handbook/ml

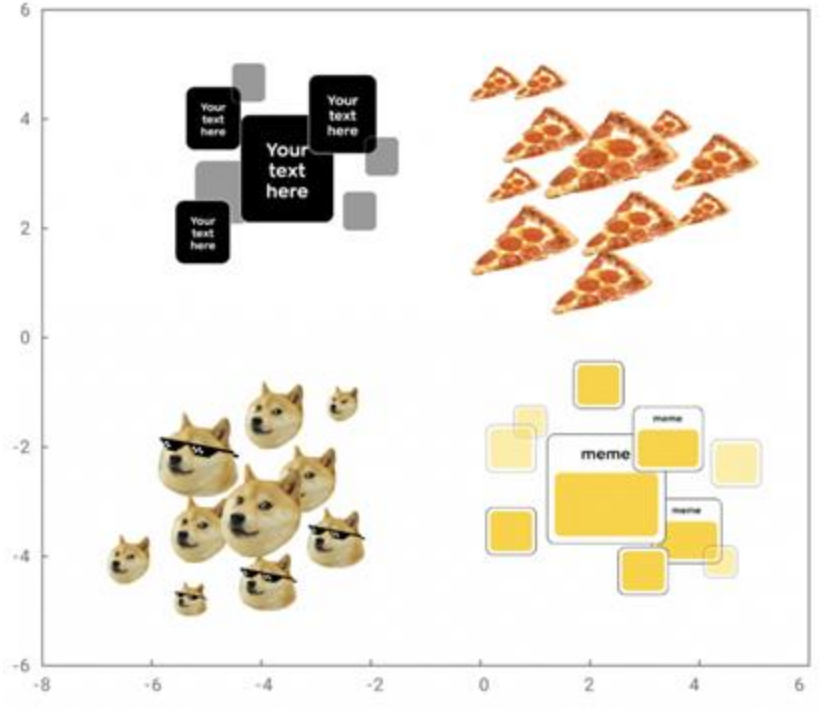
Классификация основных задач ML

1. **Обучение с учителем:** классификация, регрессия
2. **Обучение без учителя:** кластеризация, понижение размерности
3. **Обучение с подкреплением:** более сложные кейсы (роботы, игры и т.п.)

Обучение с учителем

	Regression	Classification
Outcome	Continuous	Class
Examples	Linear regression	Logistic regression, SVM, Naive Bayes

Обучение без учителя



https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-supervised-learning

Линейная регрессия

Идея:

- Есть точки (наблюдения) на плоскости или в пространстве
- Мы проводим через них линию/плоскость, которая в среднем лучше всего их описывает
- Модель учится: как изменится результат, если изменить признак

Примеры бизнес-кейсов

	ЦИАН	Пятёрочка	Альфа Банк
Тех задача	Прогнозирование цены продажи квартиры	Прогнозирование спроса на товары (сколько купят)	Прогноз доходов клиентов в банковской сфере
Бизнес задача	Поставить в объявлении цену, за которую купят и захотят продать	Привезти столько товара, сколько нужно	Сприоритизировать обработку самых платежеспособных

Линейная регрессия

Идея:

- Есть точки (наблюдения) на плоскости или в пространстве
- Мы проводим через них линию/плоскость, которая в среднем лучше всего их описывает
- Модель учится: как изменится результат, если изменить признак

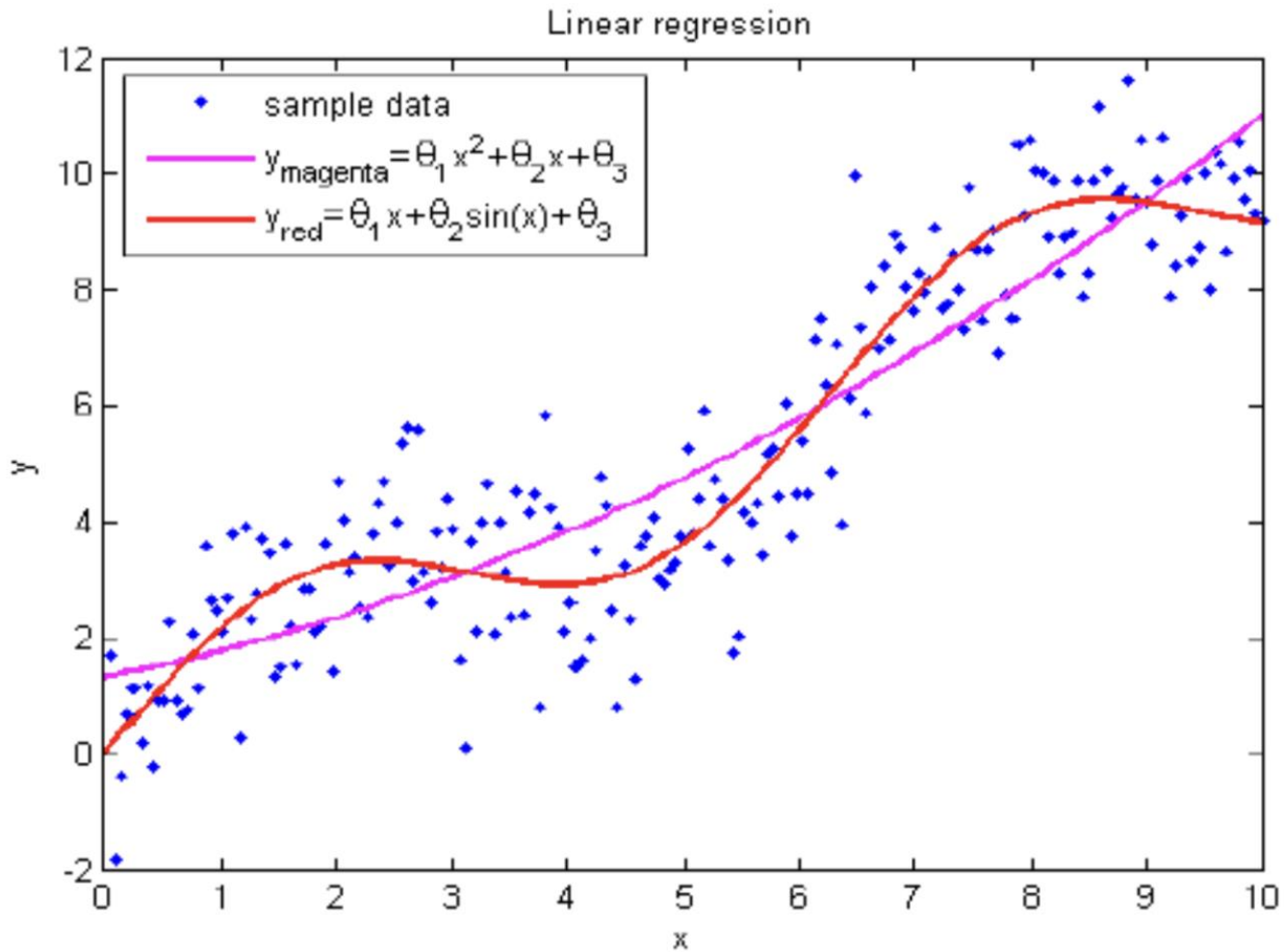
$$Y = a_0 + a_1 * feature_1 + a_2 * feature_2 + \cdots + a_m * feature_m$$

Коэффициенты a – как раз то, что модель «подбирает» на этапе обучения

Бизнес-смысл:

Коэффициент a_i показывает как в среднем меняется целевая переменная, при увеличении $feature_i$ на 1

Линейная регрессия



Логистическая регрессия

Идея:

- Тоже регрессия, но результат – вероятность события
- Модель считает число от 0 до 1 => это вероятность (ну примерно)
- Далее мы выбираем некоторый порог для принятия решения:

Если значение > порога => относим наблюдение к классу 1

Если значение < порога => относим наблюдение к классу 0

Почему логистическая:

Внутри лежит S-образная кривая, которая переводит любой результат

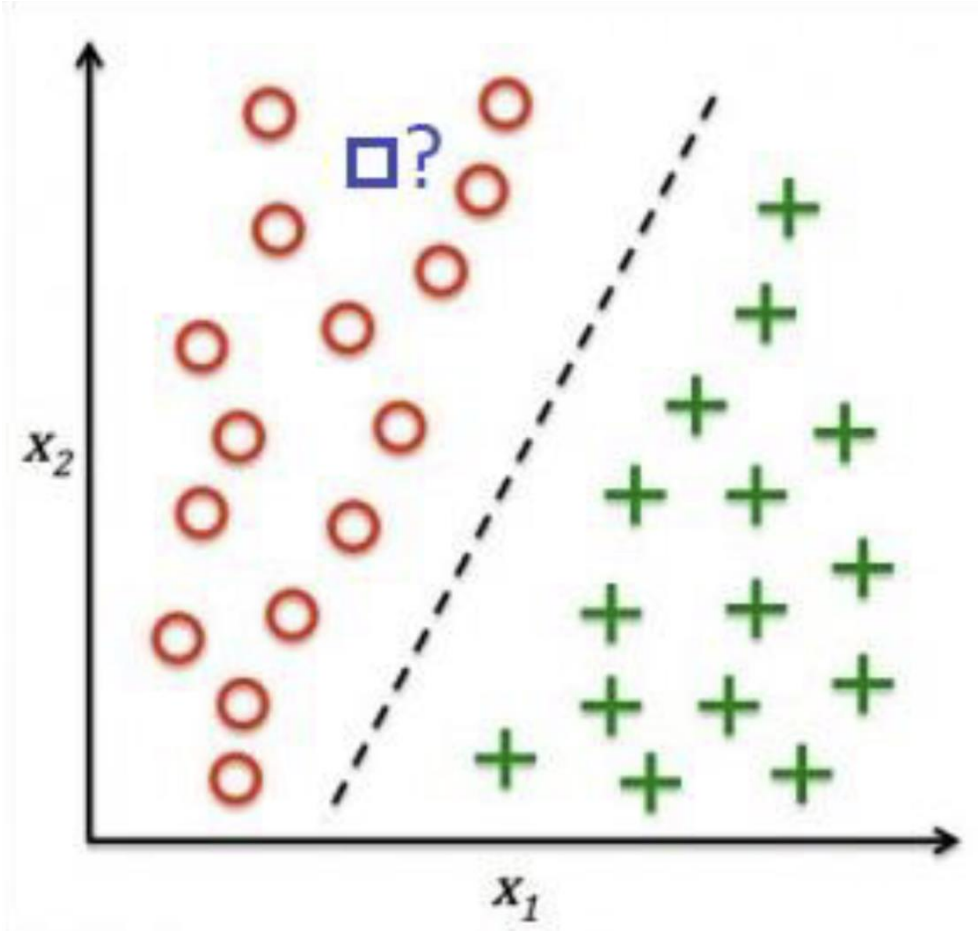
$$Y = a_0 + a_1 * feature_1 + a_2 * feature_2 + \cdots + a_m * feature_m$$

в значения от 0 до 1, где меньшее исходное значение Y ближе к 0, а большее – к 1

Примеры бизнес-кейсов

	Альфа Банк	Beeline™	OZON
Тех задача	Кредитный скоринг	Определение фродовых звонков	Определение клиентов склонных к оттоку
Бизнес задача	Не дать кредит тому, кто его не вернет или поставить % окупающий риск	Не пропустить спам / мошенников до абонентов	Удержать клиентов (например дать скидку)

Логистическая регрессия



Хотим тоже построить линию/плоскость,
но теперь она должна наилучшим образом
отделять объекты одного класса от
другого

Как именно они обучаются

1) Инициализируем какие-то коэффициенты a_i

2) Модель делает предсказание для каждой строки

3) Сравниваем предсказания модели с реальными данными

4) Обучение = минимизация суммарной ошибки по всем наблюдениям

- считаем ошибку по всем наблюдениям
- на основе производных двигаем коэффициенты в сторону наибольшего убывания ошибки

5) Повторяем шаги 2-4 до тех пор, пока не получим стабильные и точные параметры

Семинар 4

Бизнес-аналитика и ИИ как инструмент эффективного управления

Что сегодня будем делать

- Познакомимся с метриками, которыми оперируют Data Scientist’ы
- Поймем в каких задачах, что используем
- Начнем знакомство с более сложными ML-моделями

Суть

Метрики нужны не ради красивых чисел, а чтобы понять, полезна ли модель в конкретных ситуациях

Есть две группы метрик:

1. *ML-метрики — качество предсказаний как таковых*
   (MAE, RMSE, Accuracy, ROC-AUC, F1…) – об этом сегодня и поговорим

2. *Бизнес-метрики — деньги и эффект*
   (выручка, маржа, ROI кампании, уменьшение оттока, экономия колл-центра…)

- *Улучшение ML-метрики имеет смысл, только если оно ведёт к улучшению бизнес-метрики.*
- *При тестировании на исторических данных – почти никогда не можем оценивать бизнес-метрики. Оперируем ML-метриками. В онлайне (A/B-тест, работающая модель) – считаем и то, и другое*

Метрики задачи регрессии: MAE

MAE — Mean Absolute Error (средняя абсолютная ошибка)

Интуиция:

- MAE = «в среднем на сколько мы промахиваемся».
- Считаем абсолютную разницу **между прогнозом и фактом** для каждого объекта и берём среднее.
- Почему хороша для бизнеса: Измеряется в тех же единицах, что и таргет (Прогнозируем чек в рублях → MAE в рублях)

Метрики задачи регрессии: MAE

MAE — Mean Absolute Error (средняя абсолютная ошибка)

Когда хороша:

- Важно понимать средний промах на клиента:
  - *Прогноз среднего заказа в e-commerce*
  - *Прогноз месячной выручки по магазину/региону*
  - *Прогноз LTV клиента для расчёта допустимого CAC.*
- Когда ошибка в плюс и в минус одинаково болезненна.

Когда не очень:

- Если по данным есть очень большие “хвосты”: несколько экстремальных значений могут искажать картину (но не так сильно, как в некоторых других метриках)
- Не показывает, хорош ли прогноз относительно масштаба:
  - *MAE = 1000 ₽ — это много или мало? Если средний чек 1500 ₽ — ужас. Если прогнозируем годовой LTV 300 000 ₽ — ерунда.*

Метрики задачи регрессии: MSE / RMSE

MSE — Mean Squared Error (средний квадрат ошибки)
RMSE — Root Mean Squared Error (корень из MSE)

Интуиция:

- MSE — средний квадрат ошибки (более техническая метрика)
- RMSE — корень из MSE, возвращает единицы обратно (рубли, доллары).
- Крупные ошибки штрафуются сильнее, т.к. ошибка возводится в квадрат.

Если для бизнеса очень плохо сильно промахнуться (например, недозаказать или сильно переплатить), RMSE более чувствителен к таким промахам, чем MAE.

Метрики задачи регрессии: MSE / RMSE

MSE — Mean Squared Error (средний квадрат ошибки)
RMSE — Root Mean Squared Error (корень из MSE)

Когда хороши:

- Прогноз спроса или загрузки склада, где большие ошибки влекут серьёзные издержки:
  - Недозаказ → потерянная выручка.
  - Перезаказ → неликвид, логистика, списание.
- Планирование маркетинговых бюджетов по каналам, где сильные промахи приводят к перерасходу/недоиспользованию

Когда не очень:

- Если данные шумные, с редкими большими «выбросами», RMSE может казаться плохим, даже если модель в целом разумна.
- Менее интуитивен, нежели MAE
- Не показывает, хорош ли прогноз относительно масштаба

Метрики задачи регрессии: MAPE

MAPE — Mean Absolute Percentage Error (средняя относительная ошибка)

Интуиция:

- MAPE говорит: на сколько процентов в среднем мы ошибаемся.

  Пример: MAPE = 10% → в среднем наш прогноз на 10% выше/ниже факта.

Метрики задачи регрессии: MAPE

MAPE — Mean Absolute Percentage Error (средняя относительная ошибка)

Когда хорошо:

- Показывает, хорош ли прогноз относительно масштаба
- Удобно сравнивать качество прогнозов на разных рынках/продуктах:
    - В одном регионе выручка 1 млн, в другом 100 млн; MAPE нормализует масштаб.

Когда не очень:

- Плохо работает, когда фактическое значение ≈ 0 (деление на маленькие числа → огромные проценты).
- Может вводить в заблуждение, если важны абсолютные деньги, а не проценты.

Метрики задачи регрессии: R^2

R^2 — R squared (коэффициент детерминации)

Интуиция:

- Показывает, какую долю разброса в данных модель «объясняет».
- Лежит в пределах от 0 до 1 (иногда может быть < 0, но это совсем плохо).

Метрики задачи регрессии: R^2

R^2 — R squared (коэффициент детерминации)

Когда хорошо:

- Когда нужно сравнить модели между собой или с тривиальными предсказаниями (константный прогноз)
- Удобно выставлять пороги качества, когда уже есть экспертиза, т.к. значения всегда от 0 до 1
- Показывает, хорош ли прогноз относительно масштаба

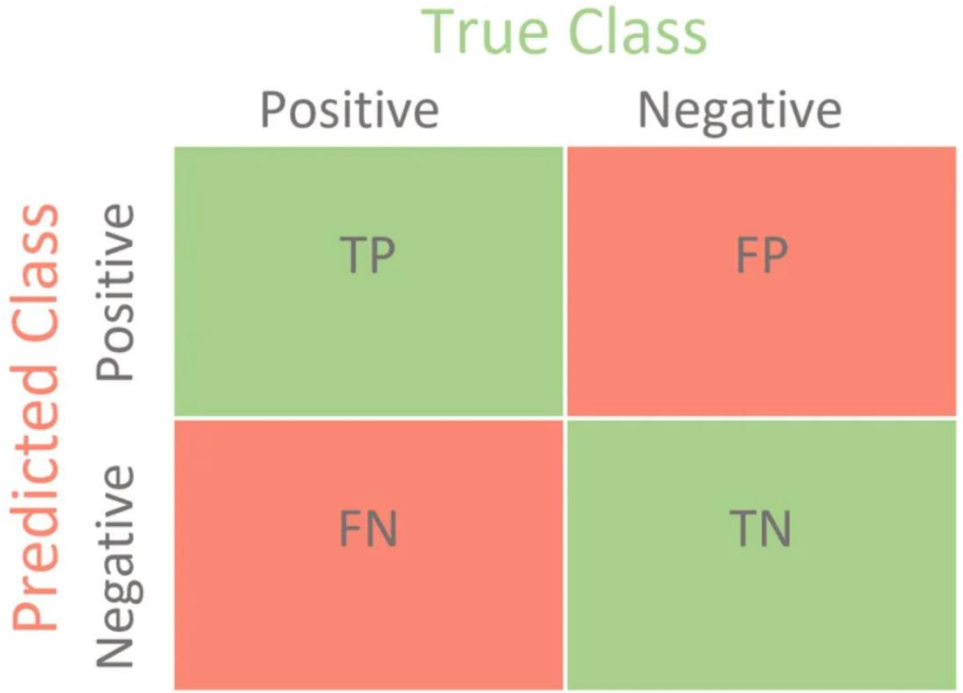
Когда не очень:

- Абсолютное значение R^2 ничего не говорит о бизнес-качестве
- Этой метрикой можно манипулировать просто за счет добавления в модель большего кол-ва признаков, что приводит к снижению устойчивости и долгосрочного качества.

Метрики задачи классификации: матрица ошибок

База: confusion matrix (матрица ошибок)

- TP (True Positive) — предсказали «да» и в реальности «да»
- FP (False Positive) — предсказали «да», но в реальности «нет»
- FN (False Negative) — предсказали «нет», а в реальности «да»
- TN (True Negative) — предсказали «нет» и в реальности «нет»



Все метрики классификации — разные способы посмотреть на эту матрицу

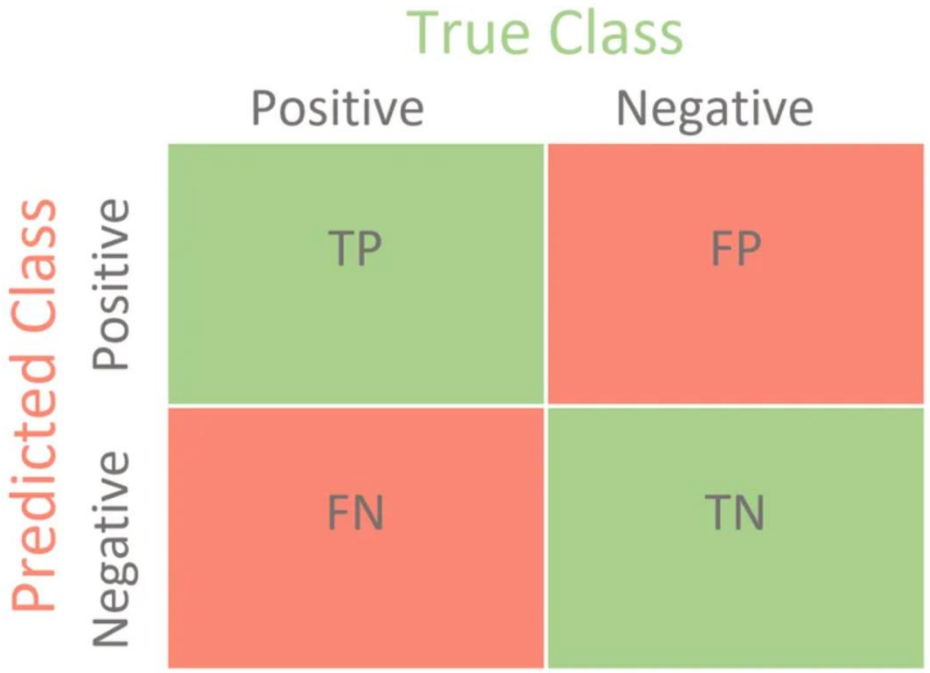
Помним: прогнозом модели классификации является вероятность принадлежности объекта к классу 1. Изменяя порог принятия решения (после какой вероятности мы считаем это наблюдение за 1), мы почти всегда меняем и метрику

Метрики задачи классификации: Accuracy

Accuracy – доля правильных ответов

Интуиция:

- Просто сколько предсказали верно среди всех прогнозов
- Accuracy = (TP + TN) / все объекты



Метрики задачи классификации: Accuracy

Accuracy – доля правильных ответов

Когда хорошо:

- Когда неточность прогноза 0 и 1 одинаково важна
- Для сбалансированных задач, где классы примерно 50/50
- Часто отражает необходимую бизнес-метрику

Когда не очень:

- При дисбалансе классов:

Пример:

  - 99% клиентов не уйдут в отток.
  - Если модель всегда говорит «не уйдёт», Accuracy = 99%, но модель бесполезна — она никого не спасает
  - А в маркетинге почти всегда есть дисбаланс

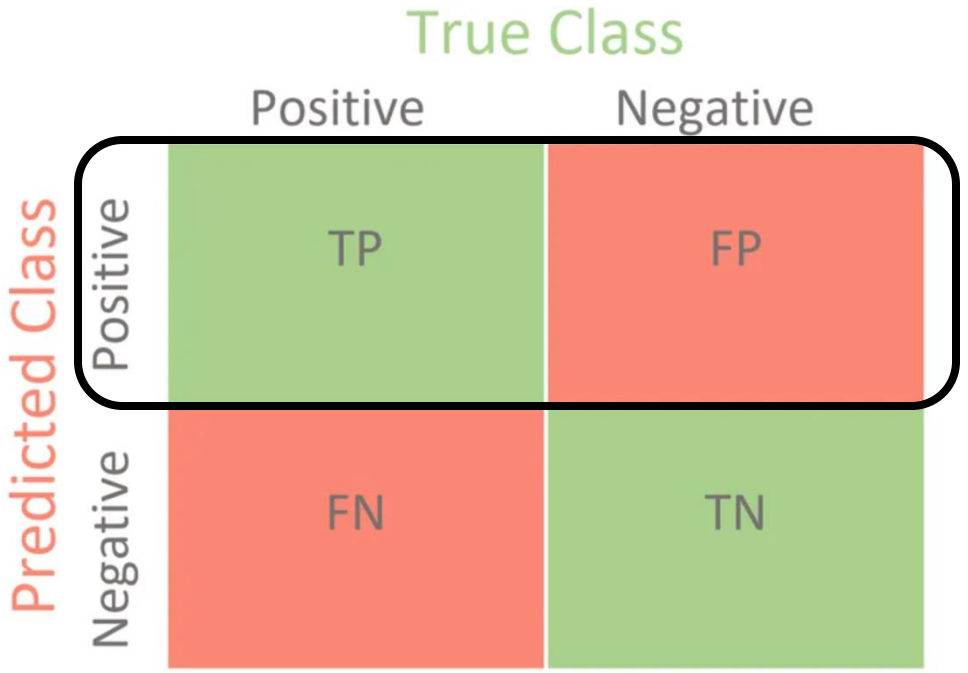
- Не учитываем, что неправильный 0 может стоить дороже неправильного 1

Метрики задачи классификации: Precision

Precision — точность (доля «правильных да»)

Интуиция:

- Из всех, кому модель сказала «да», сколько действительно “да”
- Precision = TP / (TP + FP)



Метрики задачи классификации: Precision

Precision — точность (доля «правильных да»)

Когда хорошо:

- Когда дорого “стрелять по пустым”:
  - Звонки колл-центра по «горячим» лидам
  - Дорогие персональные предложения (крупная скидка, подарок, кредит)

Когда не очень:

- Когда важно не потерять как можно больше реальных 1
- Хорошей моделью может стать та, которая угадает только самую очевидную 1, а все остальные предскажет 0

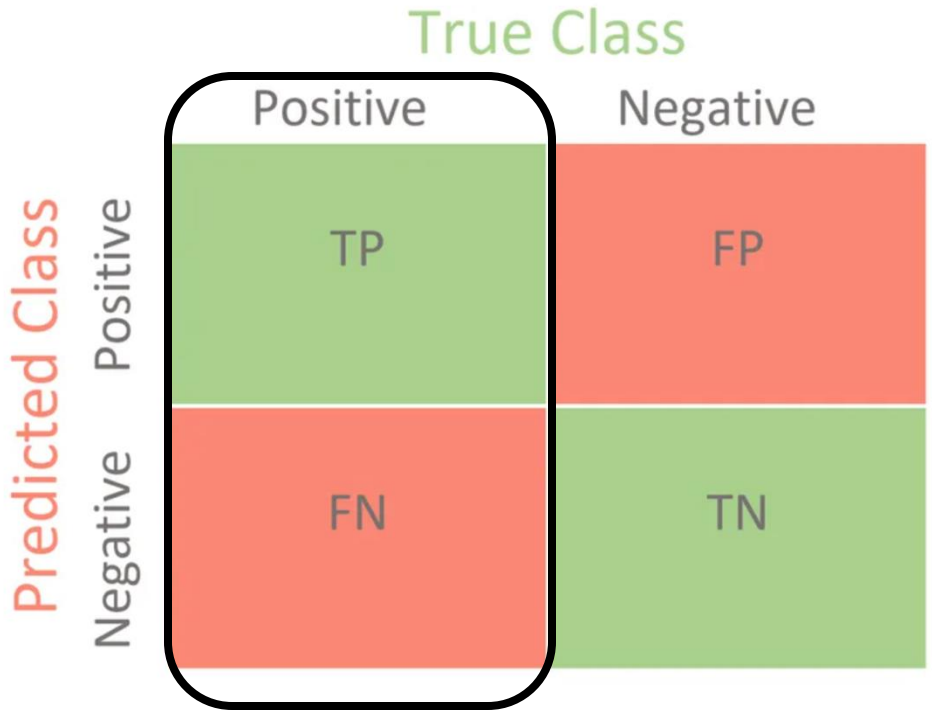
(Эта крайность может достигаться при неправильном подходе к моделированию)

Метрики задачи классификации: Recall

Recall — полнота (доля пойманных «да»)

Интуиция:

- Из всех реальных «да» сколько мы поймали?
- Recall = TP / (TP + FN)



Метрики задачи классификации: Recall

Recall — полнота (доля пойманных «да»)

Когда хорошо:

- Когда очень дорого кого-то пропустить
  - Классический пример из жизни: медицинские модели, предсказывающие наличие болезней у пациента
  - В бизнесе:
    - Потерять уходящего крупного клиента
    - Не заметить фрод
    - Проигнорировать ценного лида

Когда не очень:

- Кейсы, когда нужен Precision
- Хорошей моделью может стать та, которая константно прогнозирует всем 1

(Эта крайность может достигаться при неправильном подходе к моделированию)

Метрики задачи классификации: F1-score

F1-score — баланс Precision и Recall

Интуиция:

- F1 — гармоническое среднее Precision и Recall.
- Высокий F1 = и Precision, и Recall приемлемые, без сильного перекоса.

Когда хорошо:

- Когда важен компромисс между precision и recall
- Сравнение моделей между собой

Когда не очень:

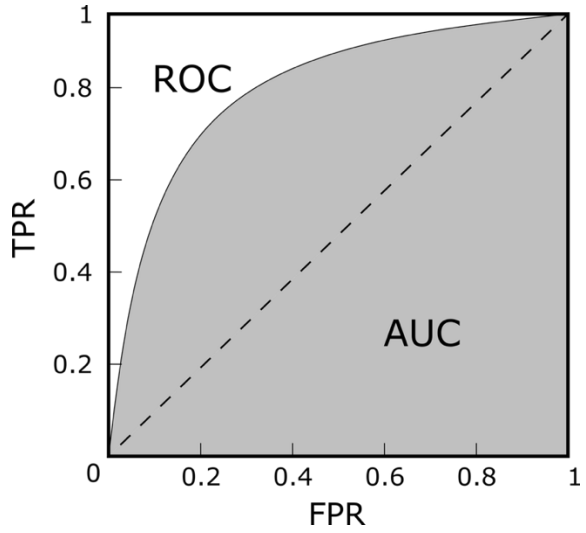
- Непонятно как интерпретировать результат и связать с бизнес-метрикой

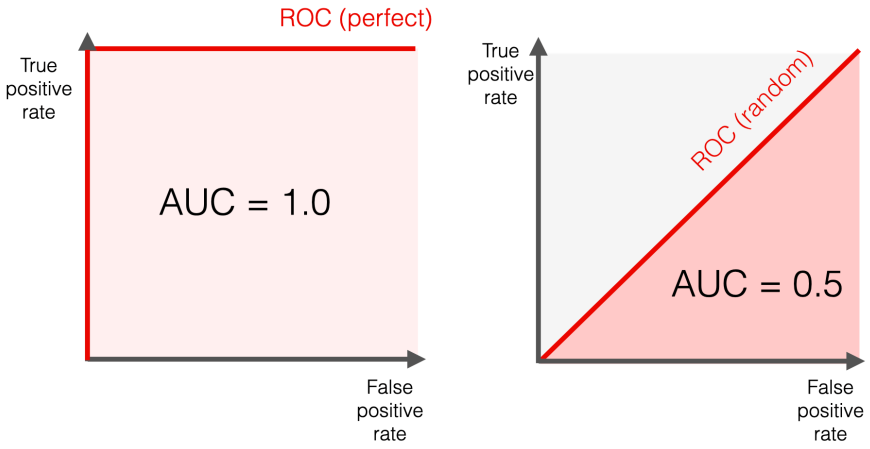
Метрики задачи классификации: ROC-AUC

ROC-AUC — площадь под ROC-кривой

Интуиция:

- ROC-кривая показывает, как меняются TPR (Recall) и FPR (FP / (FP + TN)) при разных порогах отсечения вероятности
- AUC — площадь под этой кривой от 0 до 1
- Отражает, насколько хорошо модель умеет сортировать 0 и 1 между собой





Метрики задачи классификации: ROC-AUC

ROC-AUC — площадь под ROC-кривой

Когда хорошо:

- Когда важно качество ранжирования клиентов:
  - Мы всё равно будем выбирать топ-N клиентов по скору
    (нам не важна конкретная вероятность, мы хотим понять, насколько модель действительно толкает наблюдения с таргет=1 наверх в ранжировании)
- Не зависит от конкретного порога: хорош как общая метрика для сравнения моделей.
- Дисбаланс классов не так сильно влияет

Когда не очень:

- Непонятно как интерпретировать
  результат и связать с бизнес-метрикой

Метрики задачи классификации: считаем руками

Задача: модель прогнозирует вероятность отклика на маркетинговый оффер

Клиент	Вероятность отклика	Факт отклика
A	0.95	1
B	0.90	1
C	0.85	0
D	0.80	1
E	0.70	0
F	0.60	1
G	0.55	0
H	0.40	1
I	0.30	0
J	0.20	0

Цель: выбрать, кому отправить платный оффер

Таргет: откликнется ли клиент (да = 1)

Метрики задачи классификации: считаем руками

Порог 0.5: агрессивная стратегия

Клиент	Вероятность отклика	Факт отклика
A	0.95	1
B	0.90	1
C	0.85	0
D	0.80	1
E	0.70	0
F	0.60	1
G	0.55	0
H	0.40	1
I	0.30	0
J	0.20	0

Кто получит оффер:

Вероятность больше чем 0.5 у клиентов A, B, C, D, E, F, G => им предсказываем 1

Метрики задачи классификации: считаем руками

Порог 0.5: агрессивная стратегия

Клиент	Вероятность отклика	Факт отклика
A	0.95	1
B	0.90	1
C	0.85	0
D	0.80	1
E	0.70	0
F	0.60	1
G	0.55	0
H	0.40	1
I	0.30	0
J	0.20	0

Кто получит оффер:

Вероятность больше чем 0.5 у клиентов A, B, C, D, E, F, G => им предсказываем 1

Метрики задачи классификации: считаем руками

Порог 0.5: агрессивная стратегия

Клиент	Вероятность отклика	Факт отклика
A	0.95	1
B	0.90	1
C	0.85	0
D	0.80	1
E	0.70	0
F	0.60	1
G	0.55	0
H	0.40	1
I	0.30	0
J	0.20	0

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{4 + 2}{4 + 2 + 3 + 1} = 0.60$$

$$Precision = \frac{TP}{TP + FP} = \frac{4}{4 + 3} = 0.57$$

$$Recall = \frac{TP}{TP + FN} = \frac{4}{4 + 1} = 0.80$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * 0.57 * 0.8}{0.57 + 0.8} = 0.67$$

Метрики задачи классификации: считаем руками

Порог 0.8: осторожная стратегия

Клиент	Вероятность отклика	Факт отклика
A	0.95	1
B	0.90	1
C	0.85	0
D	0.80	1
E	0.70	0
F	0.60	1
G	0.55	0
H	0.40	1
I	0.30	0
J	0.20	0

Кто получит оффер:

Вероятность больше чем 0.5 у клиентов A, B, C, D => им предсказываем 1

Метрики задачи классификации: считаем руками

Порог 0.8: осторожная стратегия

Клиент	Вероятность отклика	Факт отклика
A	0.95	1
B	0.90	1
C	0.85	0
D	0.80	1
E	0.70	0
F	0.60	1
G	0.55	0
H	0.40	1
I	0.30	0
J	0.20	0

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{3 + 4}{3 + 4 + 1 + 2} = 0.70$$

$$Precision = \frac{TP}{TP + FP} = \frac{3}{3 + 1} = 0.75$$

$$Recall = \frac{TP}{TP + FN} = \frac{3}{3 + 2} = 0.60$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * 0.75 * 0.6}{0.75 + 0.6} = 0.67$$

Метрики задачи классификации: считаем руками

Клиент	Вероятность отклика	Факт отклика
A	0.95	1
B	0.90	1
C	0.85	0
D	0.80	1
E	0.70	0
F	0.60	1
G	0.55	0
H	0.40	1
I	0.30	0
J	0.20	0

Кто получит оффер (порог 0.5): A, B, C, D, E, F, G

Кто получит оффер (порог 0.5): A, B, C, D

Допустим:

- если клиент откликнулся (TP), то чистая прибыль с него = $1000
- Каждый контакт стоит $100

Прибыль:

$$Profit = TP * 1000 - (TP + FP) * 100$$

Метрики задачи классификации: считаем руками

Клиент	Вероятность отклика	Факт отклика
A	0.95	1
B	0.90	1
C	0.85	0
D	0.80	1
E	0.70	0
F	0.60	1
G	0.55	0
H	0.40	1
I	0.30	0
J	0.20	0

Прибыль (порог 0.5): 3300

Прибыль (порог 0.8): 2600

При дешевом контакте выгоднее более низкий порог:

- стреляем шире, ловим больше True Positive, высокий Recall дает свои плюсы

Если меняем стоимость контакта = $400:

Прибыль (порог 0.5): 1200

Прибыль (порог 0.8): 1400

- более строгий порог приносит больше денег, т.к. цена ошибки высока

Переходим к моделям

Зачем нам модели сложнее регрессий

Линейная / Логистическая регрессия – это “умная линия”. Но в реальной жизни зависимости часто:

- Пороговые
- Ступенчатые
- Сильно нелинейные

Иногда регрессия реально дает нам 70-80% качества.

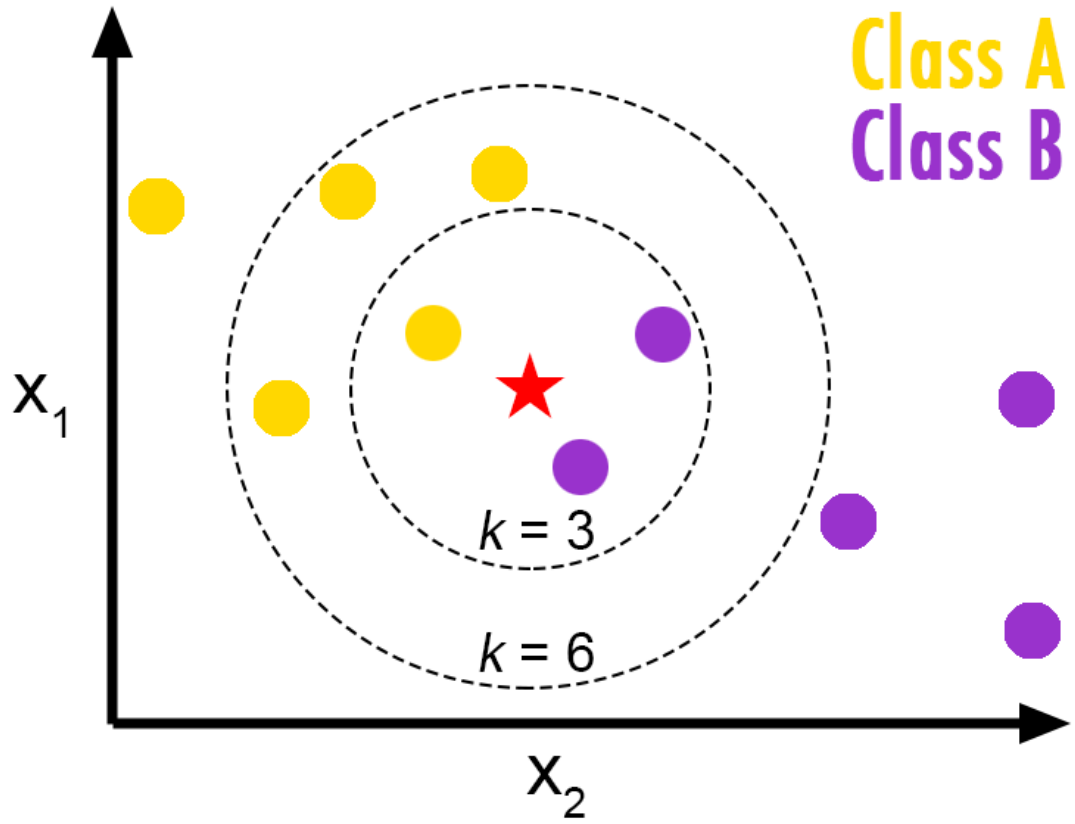
Остальное – мы добираем более сложными моделями.

Цена более сложных моделей: меньшая гибкость, сложнее интерпретация и (часто) дороже разработка и внедрение

KNN – метод k ближайших соседей

Интуиция:

Модель ищет в пространстве признаков k наиболее похожих наблюдений на то, которое нам надо предсказать.



Наша задача:

- Подобрать признаки для пространства
- Подобрать число k, на которое будет смотреть модель

KNN – метод k ближайших соседей

Преимущества

- Очень простая идея — интуитивно понятная
- Не требует долгого обучения (фактически «обучение» = просто хранение данных)
- Может хорошо работать, даже если мало признаков или данных

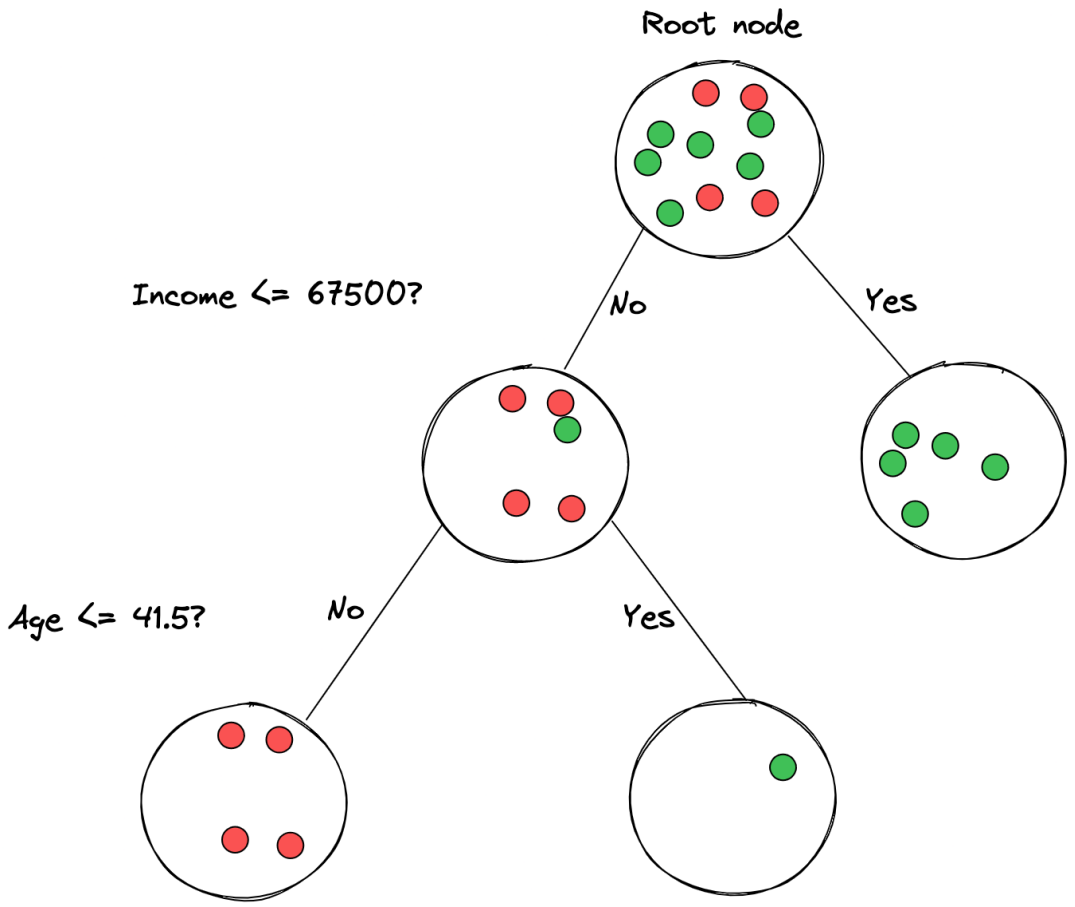
Недостатки

- Медленный на предсказании: чтобы предсказать для одного клиента, надо «пройтись» по многим другим
- Плохо масштабируется на большие объёмы данных
- Не любит много признаков (проклятие размерности).
- Интерпретация для бизнес-задачи: трудно объяснить, почему модель решила так, кроме «у похожих так же» (не всегда этого достаточно)

Decision Tree — деревья решений

Интуиция:

Строит иерархию if-else правил.
На каждом шаге выбирается признак и порог, которые лучше всего разделяют клиентов



Наша задача:

- Подобрать признаки для пространства
- Подобрать оптимальную глубину дерева

Decision Tree — деревья решений

Преимущества

- Легко интерпретируемые правила принятия решения
- Может ловить нелинейности, пороги, сложные комбинации признаков
- Быстро обучается и предсказывает

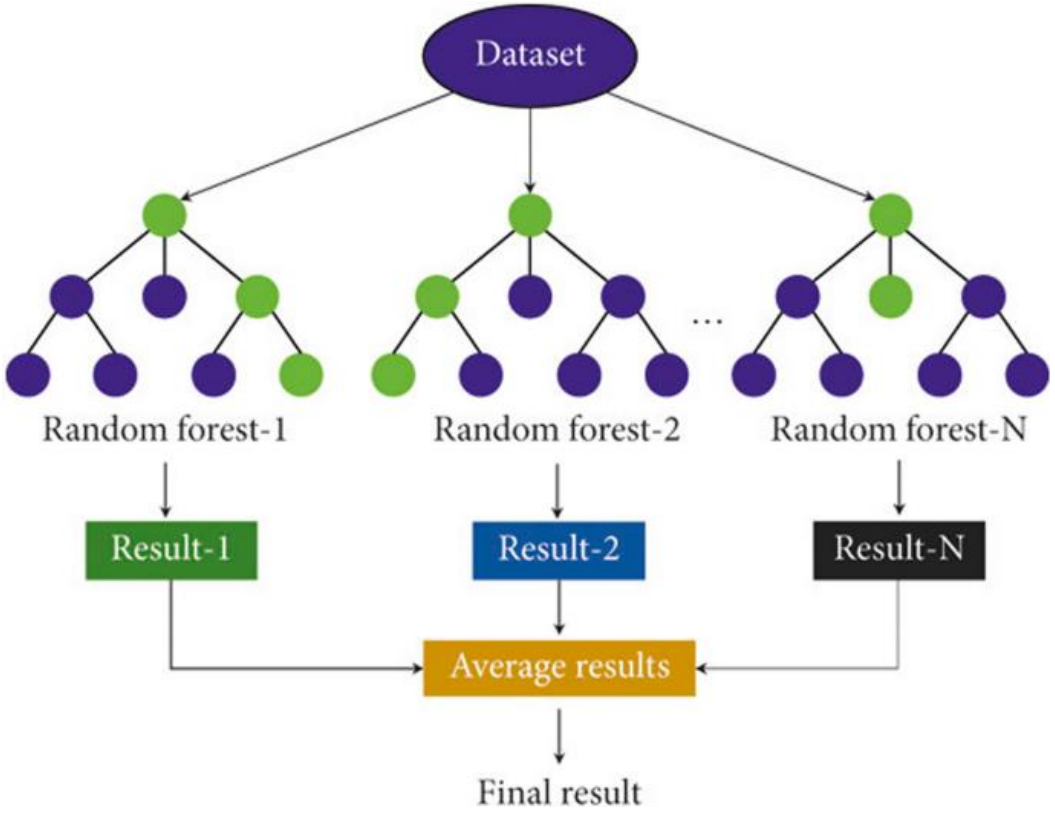
Недостатки

- Качество не сильно лучше регрессии
- Дерево может стать слишком глубоким и подогнанным под шум (переобучение)
- Малые изменения в данных → дерево может сильно поменяться (нестабильность структуры). Не всегда умеет экстраполировать свою логику на новые примеры

Random Forest — случайный лес

Интуиция:

Строим много деревьев на разных поднаборах данных и признаков. Результат усредняем для прогноза



Наша задача:

- Подобрать признаки для пространства
- Подобрать оптимальную глубину дерева и кол-во этих деревьев

Random Forest — случайный лес

Преимущества

- Значительно точнее, чем одно дерево
- Меньше проблем с переобучением, чем у одного дерева
- Быстро обучается и предсказывает

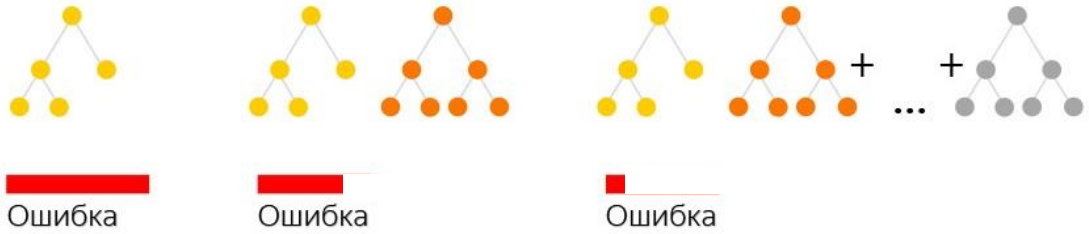
Недостатки

- Практически полностью теряем интерпретируемость (что-то как-то влияет)
- Требует большого кол-ва данных для обучения
- Для очень больших данных может быть дорогим по ресурсу хранения и предсказания (нужно внедрять более сложную систему и хранить больше данных)

Gradient Boosting — градиентный бустинг

Интуиция:

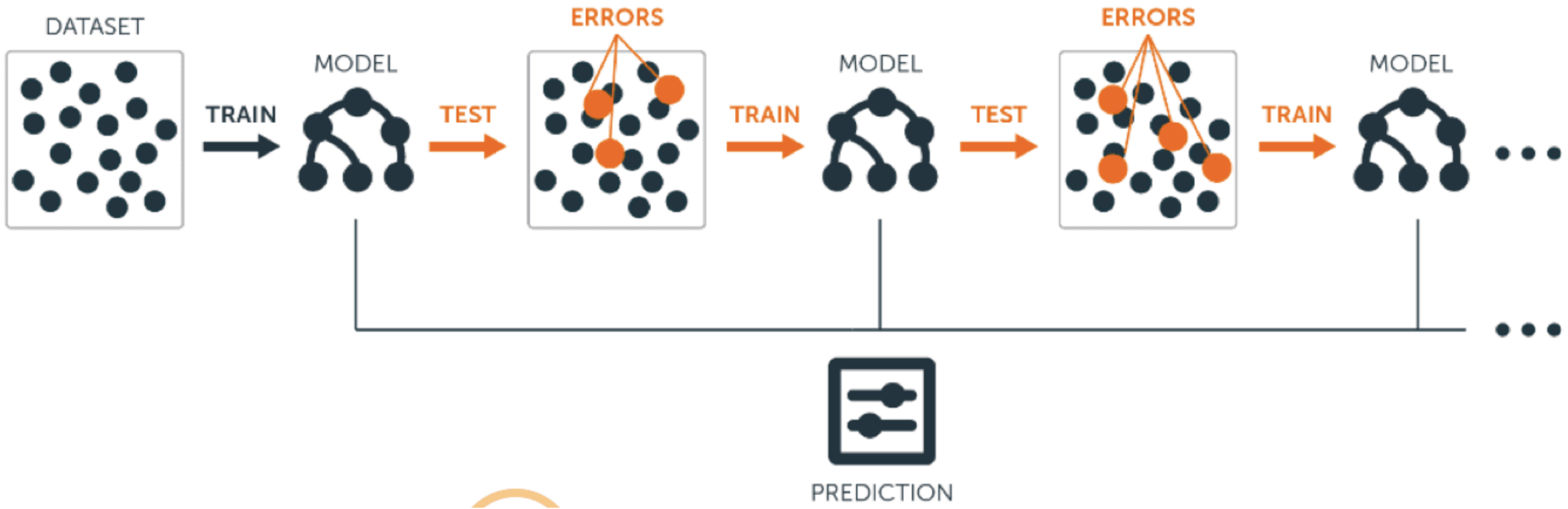
много маленьких деревьев,
исправляющих ошибки друг друга



Алгоритм:

1) Первое дерево делает очень грубый прогноз
2) Следующее дерево учится уже не на таргет, а на отклонение прогноза первого дерева от таргета
3) Делаем так много раз, складываем и получаем очень точный прогноз

Детализация по градиентному бустингу



Gradient Boosting — градиентный бустинг

Преимущества

- Часто самая высокая точность на табличных бизнес-данных
- Может хорошо работать с разными типами признаков

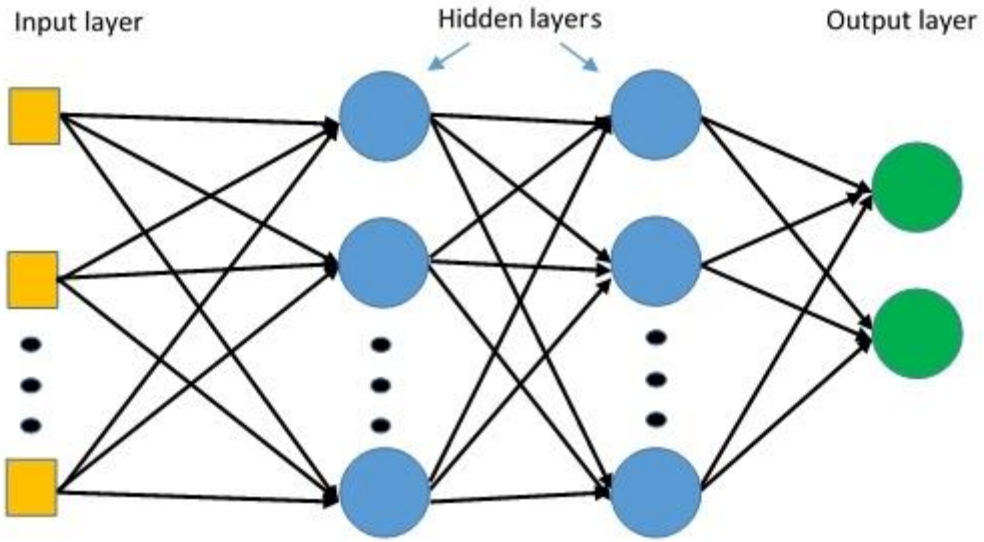
Недостатки

- Практически полностью теряем интерпретируемость (что-то как-то влияет)
- Требует большого кол-ва данных для обучения
- Может быть дорогим по ресурсу хранения и предсказания
- Дольше обучается и применяется (некритично)
- Есть высокий риск переобучения модели
- Сложность настройки и внедрения: здесь уже по-хорошему нужен Data Scientist

MLP — многослойный перцептрон (простая нейросеть)

Интуиция:

Будем обучать много линейных регрессий, но между ними положим нелинейную связку, чтобы ловить более сложные закономерности



Алгоритм:

1) Берёт входные признаки и преобразует их линейно
2) Пропускает через нелинейную функцию
3) Несколько таких слоёв подряд дают возможности представлять очень сложные зависимости

MLP — многослойный перцептрон (простая нейросеть)

Преимущества

- Очень гибкая модель, может приближать очень сложные зависимости
- Очень высокая точность на табличных бизнес-данных
- В некоторых задачах обошел бустинг и продолжает улучшаться (много модных статей)

Недостатки

- **Полностью** теряем интерпретируемость (что-то как-то влияет)
- Требует **очень** большого кол-ва данных для обучения
- **Самый дорогой** по ресурсу хранения и предсказания
- Обучается и применяется **дольше всех** (бывает очень критично)
- Есть высокий риск переобучения модели
- **Сложность** настройки и внедрения: здесь уже точно нужен Data Scientist