Введение в машинное обучение

Лекция 1. Введение. kNN. Naïve Baies

Осень 2025

Формат курса

- Лекционные занятия: онлайн/оффлайн
- Семинарские занятия (по группам?): онлайн/оффлайн
- Домашние задания с фиксированным дедлайном

Оценка за курс: семинары / домашние задания / экзамен

Программа курса

1. Naive Bayes, kNN
2. Линейные модели
3. Логистическая регрессия
4. SVM, PCA
5. BVD, k
6. Деревья решений. Методы ансамблирования моделей
7. Градиентный бустинг
8. Введение в нейронные сети
9. Методы кластеризации и понижения размерности
10. Неградиентная оптимизация
11. Задачи ранжирования и матчинга

Машинное обучение -

«Компьютерная программа говорит, что она учится на основе опыта E в отношении некоторого класса задач T и меры качества P, если её результативность при решении задач T, измеряемая с помощью P, улучшается с опытом E.»

Том Митчел

Пример:

- Задача T: например, классификация писем как «спам/не спам»
- Опыт E: примеры писем, которые уже были помечены как спам или нет.
- Метрика P: процент правильных классификаций.

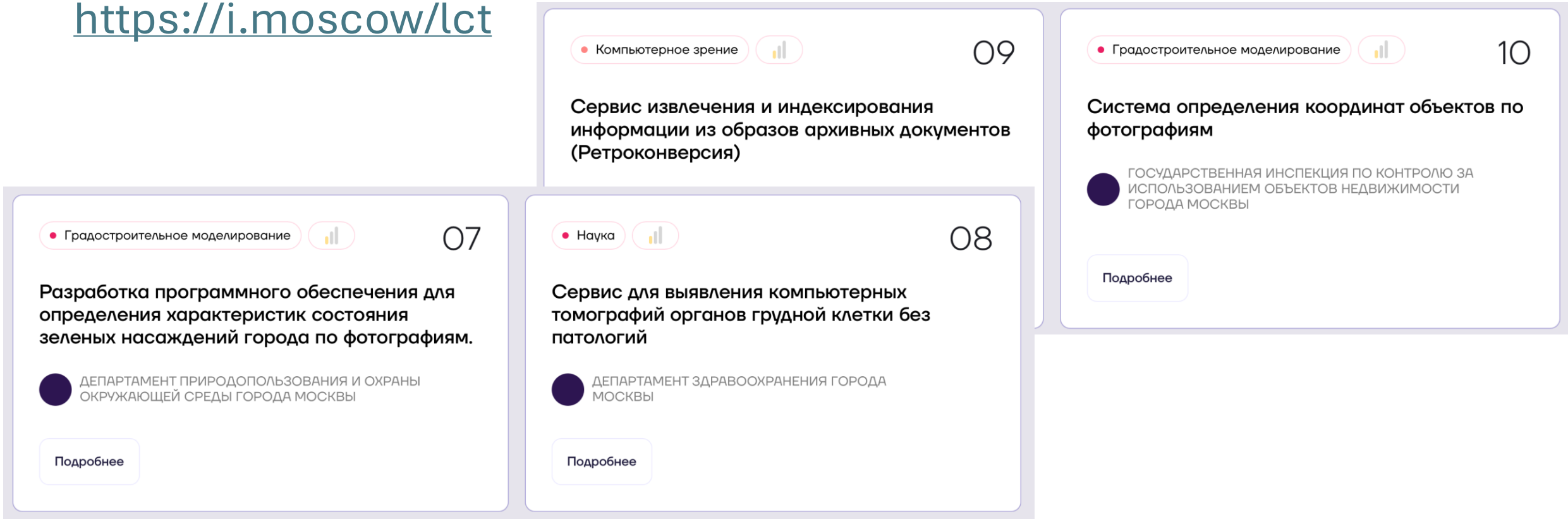
Если программа со временем (с обучением) повышает точность – она «обучается»

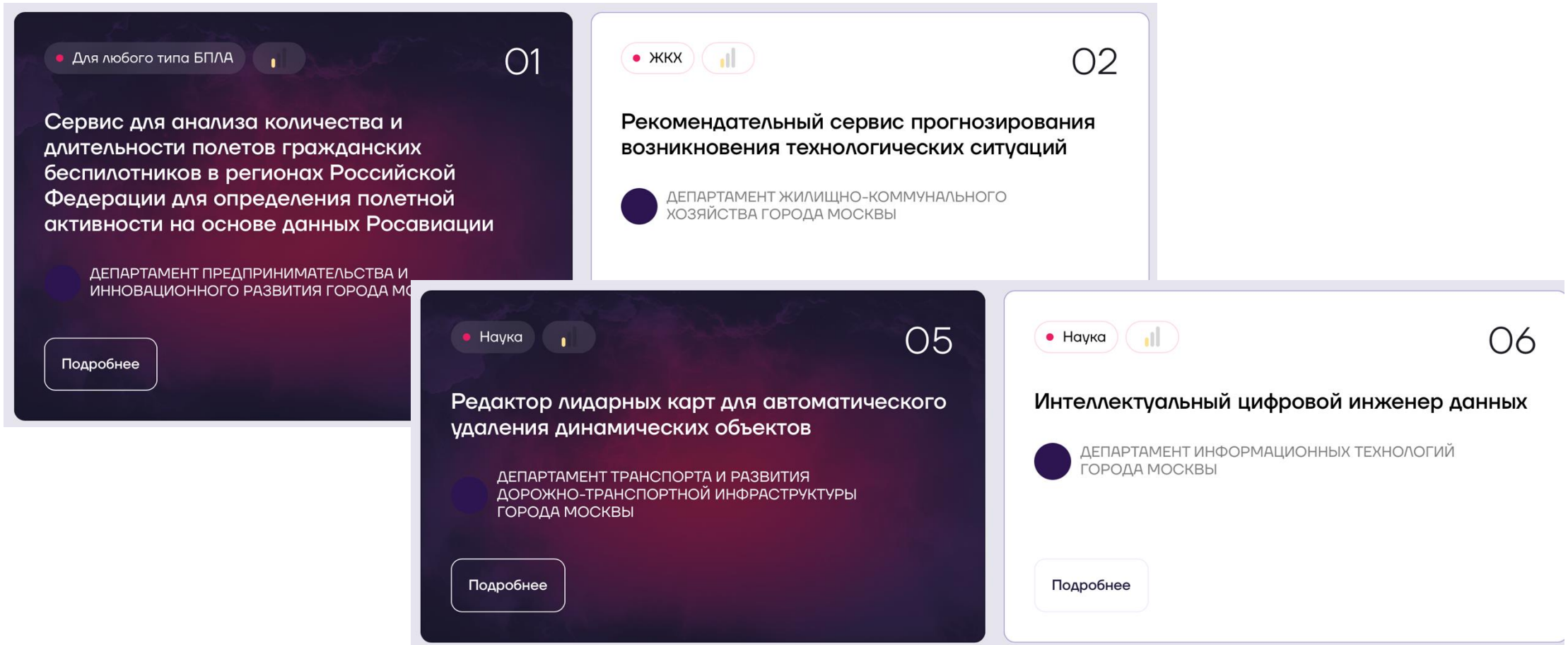
Примеры

- Рекомендации фильмов в Netflix или Spotify.
- Распознавание речи в Siri или Google Assistant.
- Автоматический перевод в Google Translate.
- Медицинские системы поддержки принятия решений.
- Финансовые алгоритмы для выявления мошенничества.
- Распознавание объектов.

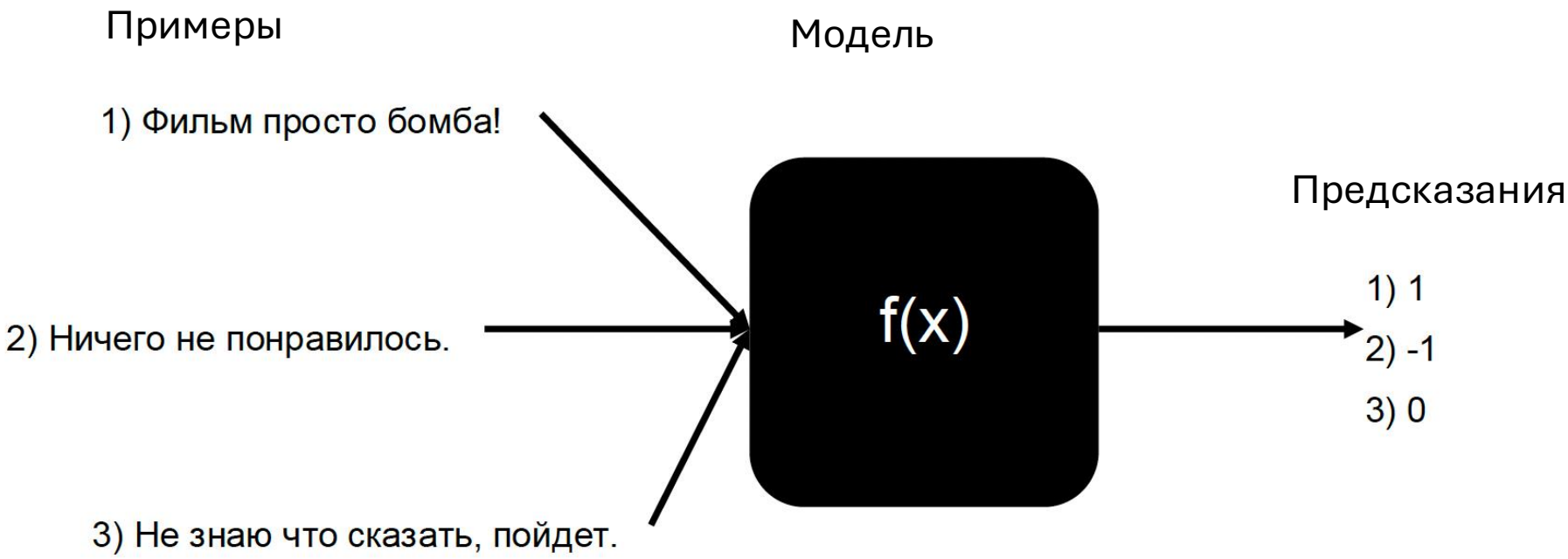
Примеры из бизнеса

Вот некоторые АКТУАЛЬНЫЕ на сегодняшний момент задачи с хакатона «Лидеры цифровой трансформации»
https://i.moscow/lct




Для любого типа БПЛА
01
Сервис для анализа количества и длительности полетов гражданских беспилотников в регионах Российской Федерации для определения полетной активности на основе данных Росавиации
ДЕПАРТАМЕНТ ПРЕДПРИНИМАТЕЛЬСТВА И ИННОВАЦИОННОГО РАЗВИТИЯ ГОРОДА МО
Подробнее
ЖКХ
02
Рекомендательный сервис прогнозирования возникновения технологических ситуаций
ДЕПАРТАМЕНТ ЖИЛИЩНО-КОММУНАЛЬНОГО ХОЗЯЙСТВА ГОРОДА МОСКВЫ
Наука
05
Редактор лидарных карт для автоматического удаления динамических объектов
ДЕПАРТАМЕНТ ТРАНСПОРТА И РАЗВИТИЯ ДОРОЖНО-ТРАНСПОРТНОЙ ИНФРАСТРУКТУРЫ ГОРОДА МОСКВЫ
Подробнее
Наука
06
Интеллектуальный цифровой инженер данных
ДЕПАРТАМЕНТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ГОРОДА МОСКВЫ
Подробнее

Что общего у задач?



Основные понятия

- Данные — информация, с которой мы работаем. Обычно это таблицы или изображения.
- Объект (sample, instance) — одна запись, например, один пациент или один дом.
- Признаки (features) — характеристики объекта. У пациента это может быть возраст, давление, вес. У дома — площадь, район, количество комнат. Признаки категориальные (пол…) и числовые (возраст, цена), бинарные.
- Выборка (dataset) — множество объектов.

  Обучающая выборка (training set) — для обучения.

  Тестовая выборка (test set) — для проверки.

  Валидационная выборка (validation set) — для настройки параметров.
- Модель — математическая структура, которая описывает данные (например, линейная функция).
- Гипотеза — конкретное предположение о закономерности.
- Ошибка — разница между предсказанием модели и реальностью.

- Переобучение (overfitting): модель слишком хорошо запомнила обучающие данные и плохо работает на новых.
- Недообучение (underfitting): модель слишком простая и не улавливает закономерности.

Направления

Обучение с учителем (Supervised Learning)

У нас есть данные с правильными ответами.

- Пример: есть фото животных и подписи «кот», «собака».
- Цель: научить модель предсказывать метку для новых данных.
- Подзадачи: классификация, регрессия.

Обучение без учителя (Unsupervised Learning)

Данные без меток. Модель сама ищет закономерности.

- Пример: сегментация клиентов банка на группы.
- Подзадачи: кластеризация, снижение размерности.

Обучение с подкреплением (Reinforcement Learning)

Агент взаимодействует со средой и получает награду или штраф.

- Пример: обучение робота ходить.

Дополнительно:

- Semi-supervised learning — частично размеченные данные.
- Self-supervised learning — обучение на огромных неразмеченных данных (важно для современных нейросетей).

Типы задач

- **Классификация**
- - Задача: присвоить объекту категорию.
- - Примеры: фильтрация спама, определение болезни по симптомам, распознавание цифр.
- **Регрессия**
- - Задача: предсказать числовое значение.
- - Примеры: прогноз цен на жильё, предсказание температуры, оценка продаж.
- **Кластеризация (Без учителя)**
- - Задача: разделить объекты на группы по сходству.
- - Примеры: сегментация клиентов банка, группировка новостей, выделение тем в текстах.
- **Снижение размерности**
- - Задача: уменьшить число признаков, сохранив суть.
- - Пример: сжатие изображений, визуализация многомерных данных.
- **Ранжирование**
- - Задача: упорядочить объекты по релевантности (...в поисковой выдаче, определение схожести).
- - Пример: поисковые системы, рекомендательные алгоритмы.

Классификация

бинарная классификация (либо «да», либо «нет»)

Например, мы можем предсказывать, кликнет ли пользователь по рекламному объявлению, вернёт ли клиент кредит в установленный срок.

многоклассовая (multiclass) классификация

- это задача, где объект принадлежит строго одному классу – распознавание объектов.
- на выходе вектор вероятностей для класса (сумма равна 1)

Пример: [«Планшет», «Смартфон», «Ноутбук»]

Результат модели: [0.1, 0.0, 0.9]

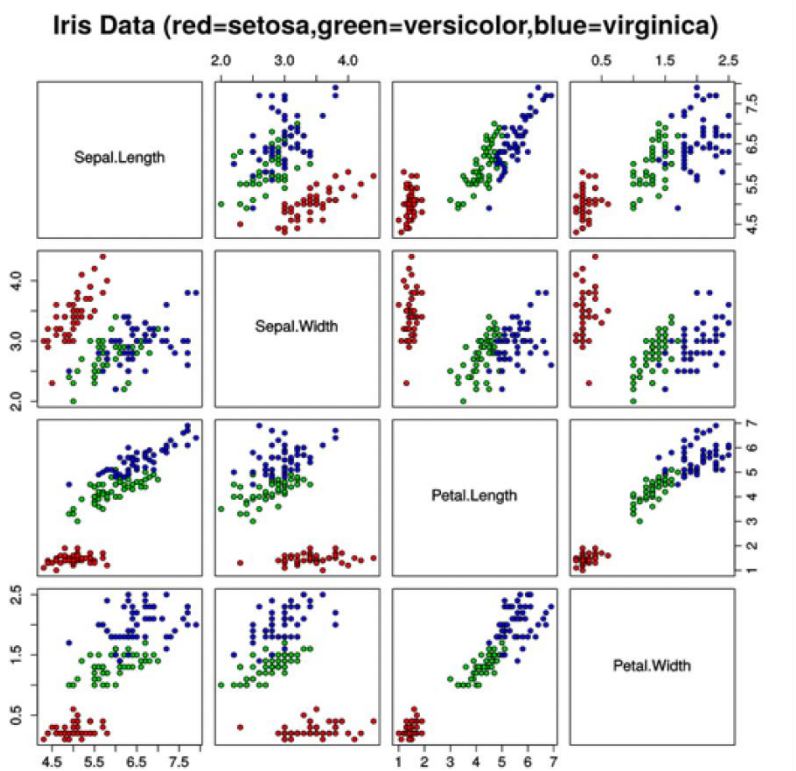
Классификация

Многоклассовая классификация с пересекающимися классами (multilabel classification)

Например, задача автоматического проставления тегов для статей, распознавания объектов и тд.

Поэтому на выходе вектор вероятностей для класса и сумма НЕ равна 1

Примеры задач (Ирисы Фишера)



Какая задача: классификация

Признаки: числовые

Примеры задач (стоимость дома)

Нужно предсказать стоимость дома. Есть обучающий датасет со следующими признаками:

- ✓ Удаленность от метро;
- ✓ Оценка состояния дома (плохое, среднее, хорошее, отличное);
- ✓ Количество комнат;
- ✓ Площадь;
- ✓ Год строительства;
- ✓ Название района, в котором находится дом.

Какая задача: регрессия

Признаки: числовые, порядковые, категориальные

Примеры задач (поиск страницы в Интерне

Получив запрос от пользователя нужно найти наиболее полезные документы из некоторой базы.

Что нам известно:

- ✓ Запрос пользователя;
- ✓ Текст документа;
- ✓ Какие ключевые слова есть в каждом документе;
- ✓ Насколько каждый документ популярен.

Какая задача: ранжирование

Признаки: ???

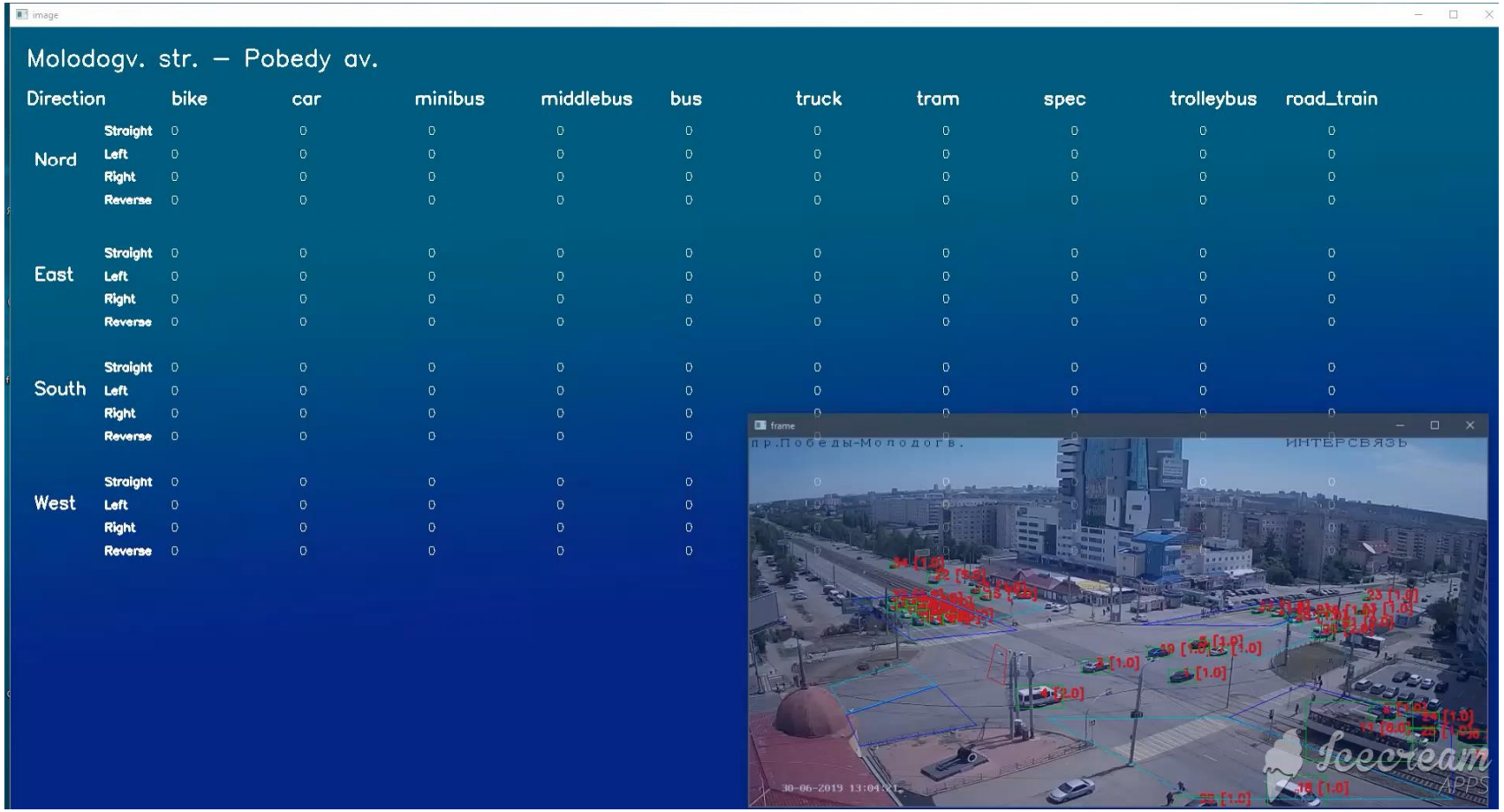


image
Molodogv. str. — Pobedy av.
Direction bike car minibus middlebus bus truck tram spec trolleybus road_train
Nord Straight 0 0 0 0 0 0 0 0 0 0
Left 0 0 0 0 0 0 0 0 0 0
Right 0 0 0 0 0 0 0 0 0 0
Reverse 0 0 0 0 0 0 0 0 0 0
East Straight 0 0 0 0 0 0 0 0 0 0
Left 0 0 0 0 0 0 0 0 0 0
Right 0 0 0 0 0 0 0 0 0 0
Reverse 0 0 0 0 0 0 0 0 0 0
South Straight 0 0 0 0 0 0 0 0 0 0
Left 0 0 0 0 0 0 0 0 0 0
Right 0 0 0 0 0 0 0 0 0 0
Reverse 0 0 0 0 0 0 0 0 0 0
West Straight 0 0 0 0 0 0 0 0 0 0
Left 0 0 0 0 0 0 0 0 0 0
Right 0 0 0 0 0 0 0 0 0 0
Reverse 0 0 0 0 0 0 0 0 0 0
frame
пр.Победы-Молодогв.
ИНТЕРСВЯЗЬ
30-06-2019 13:04:
Icecream APPS

Этапы проекта

1. Сбор данных

2. Подготовка данных. Очистка, нормализация, работа с пропущенными значениями.

3. Выбор модели. Решаем, какую математическую схему использовать.

4. Обучение. Подбор параметров на обучающей выборке.

5. Оценка. Проверяем модель на тестовой выборке.

6. Внедрение. Используем модель в реальном приложении.

Алгоритм kNN

Алгоритм k-ближайших соседей (k nearest neighbors)

Чтобы предсказать ответ для нового объекта, kNN просто ищет **k** самых похожих объектов в обучающей выборке и «спрашивает у них мнение». При классификации – целевой объект принимает значение большинства, при регрессии — усредняет численные значения.

Не имеет фазы обучения

Характеризует целевой объект исходя из «качеств» ближайших k объектов

Алгоритм поиска ответа

1. Храним обучающие данные: признаки XX и ответы yy. Никакого «обучения» в привычном смысле нет — это ленивый алгоритм: он просто запоминает данные.

2. Выбираем число соседей **k** и **метрику расстояния** (как измерять «похожесть»).

3. Для целевого объекта O:

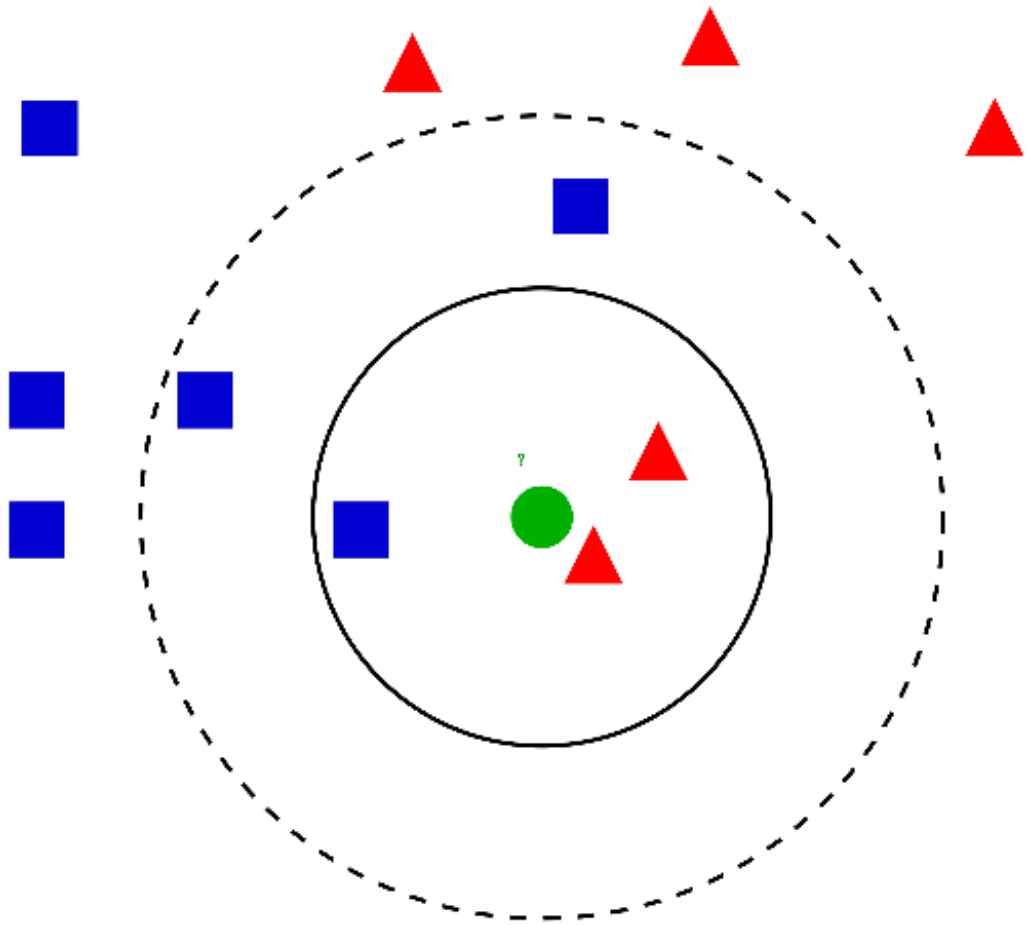
    - считаем расстояния от O до **всех** известных объектов

    - находим **k** с наименьшим расстоянием — это соседи.

4а) классификация: выбираем класс, который встречается среди соседей чаще (можно с весами).

4б) регрессия: берём среднее (или взвешенное среднее) их целевых значений.

Классификация



https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

Расстояния

- **Евклидово**:

  $d(\mathbf{x}, \mathbf{z}) = \sqrt{\sum_i (x_i - z_i)^2}$ — «обычная» геометрическая дистанция.

- **Манхэттенское**:

  $d(\mathbf{x}, \mathbf{z}) = \sum_i |x_i - z_i|$ — как если идти по кварталам.

- **Минковского** (обобщает оба):

  $$d_p(\mathbf{x}, \mathbf{z}) = \left(\sum_i |x_i - z_i|^p \right)^{1/p}.$$

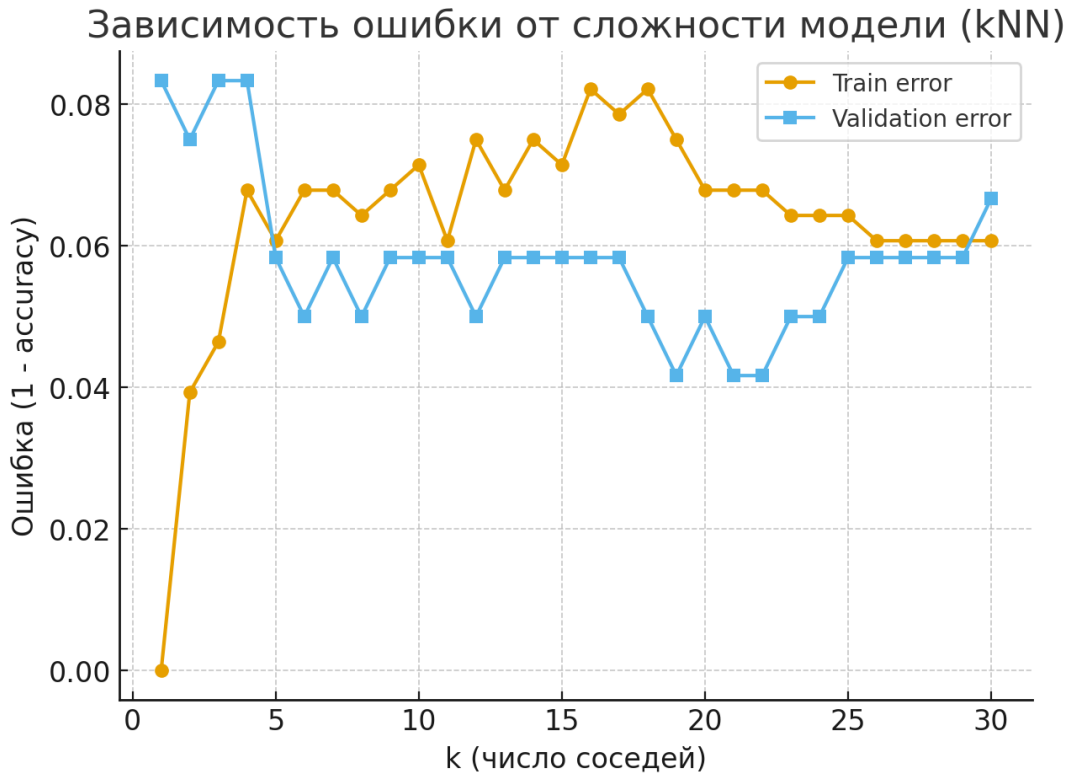
И не только....

Выбор количества соседей

- малое число – высокая вариативность, чувствительность к шуму
- большое число – «размытость» границ между классами

Практический подход: перебор вариантов, оценка результатов.

Какой k брать? Четный или нечетный?



Предобработка данных

- Масштабирование
- Обработка категориальных
- Пропуски
- Работа с шумами

Наивный байесовский классификатор

Мы предсказываем класс объекта, **вычисляя вероятность** того, что объект принадлежит каждому классу на основе **формулы Байеса**, и выбираем класс с наибольшей вероятностью.

Формула Байеса связывает:

- то, что мы *хотим узнать* (**апостериорная вероятность**),
- с тем, что мы *уже знаем* (**априорная вероятность** и вероятности признаков).

Формула:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

- $P(A \mid B)$ —вероятность гипотезы A(например, «письмо — спам») при условии наблюдения B(например, «в письме есть слово *discount*»).
- $P(B \mid A)$ —вероятность увидеть наблюдение B, если гипотеза Aверна.
- $P(A)$ —априорная вероятность (насколько часто встречается Aвообще).
- $P(B)$ —нормализатор: вероятность наблюдать Bпри любых обстоятельствах.

Алгоритм решения

Задача: определить является ли письмо спамом

- рассчитаем оценку для каждого класса и выберем максимальную по формуле:

$$arg\max[P(Q_k)\prod_{i=1}^{n}P(x_i|Q_k)] \qquad P(Q_k)=\frac{\text{число документов класса } Q_k}{\text{общее количество документов}}$$

$$P(x_i|Q_k)=\frac{\alpha+N_{ik}}{\alpha M+N_k}$$ вхождение слова x в документа класса Q

- Nk –количество слов, входящих в документ класса Q
- Mik –количество слов из обучающей выборки
- N – количество вхождений слова *x* в документ класса Q
- α —параметр для сглаживания; мы не можем обучить алгоритм всем словам, и если его не применять, то оценка будет равна 0; $0 < \alpha \leq 1$ (сглаживание Лапласа)

Задача

У нас есть обучающая выборка:

Спам:

- «Путевки по низкой цене»
- «Акция! Купи шоколадку и получи телефон в подарок»

Не спам:

- «Завтра состоится собрание»
- «Купи семь килограмм и шоколадку»

Нужно классифицировать фразу:
«Купи килограмм яблок и шоколадку»

Шаг 1. Составляем словарь

Выделим все уникальные слова (игнорируем пунктуацию, приводим к нижнему регистру):

спам: {путевки, по, низкой, цене, акция, купи, шоколадку, и, получи, телефон, в, подарок}

не спам: {завтра, состоится, собрание, купи, семь, килограмм, и, шоколадку}

Словарь (уникальные слова всего):
{путевки, по, низкой, цене, акция, купи, шоколадку, и, получи, телефон, в, подарок, завтра, состоится, собрание, семь, килограмм}

Итого: **17 слов**.

Шаг 2. Вероятности классов

Всего писем = 4

Спам = 2 → $P(Спам) = 2/4 = 0.5$

Не спам = 2 → $P(Не\ спам) = 2/4 = 0.5$

Шаг 3. Считаем условные вероятности слов (с Лапласовским сглаживанием)

Метод:

$$P(\textit{слово} \mid \textit{Класс}) = \frac{\textit{частота слова в классе} + 1}{\textit{общее число слов в классе} + \textit{словарь}}$$

- Для **спама**:
  Общее число слов = 11
  Размер словаря = 17
  → знаменатель = $11 + 17 = 28$
- Для **не спама**:
  Общее число слов = 8
  Размер словаря = 17
  → знаменатель = $8 + 17 = 25$

Для предложения «Купи килограмм яблок и шоколадку»

(ключевые слова: **купи, килограмм, яблок, и, шоколадку**)

В классе «Спам» (знаменатель = 28):

Слово «купи» встречается 1 раз в спаме (1+1)/28=2/28=0.071.

Аналогично для остальных 5 слов

Перемножаем:

P("слова" |"Спам")=0.071·0.036·0.036·0.071·0.071≈4.7×10^(−6)

P("Спам" |"слова") 0.5·4.7×10^(−6)≈2.35×10^(−6)

В классе «Не спам» (знаменатель = 25):

«купи» встречается 1 раз в не спаме

$$(1 + 1)/25 = 2/25 = 0.08$$

Перемножаем:

$$P(\textit{слова} \mid \textit{Не спам}) = 0.08 \cdot 0.08 \cdot 0.04 \cdot 0.08 \cdot 0.08 \approx 1.64 \times 10^{-5}$$

$$P(\textit{Не спам} \mid \textit{слова}) \propto 0.5 \cdot 1.64 \times 10^{-5} \approx 8.2 \times 10^{-6}$$

Метрики качества

1 Самый верхний уровень – это бизнес-метрики, например, будущий доход сервиса. Их трудно измерить в моменте, они сложным образом зависят от совокупности всех наших усилий, не только связанных с машинным обучением.

2 Онлайн (online) метрики – это характеристики работающей системы, с помощью которых мы надеемся оценить, что будет с бизнес-метриками. Например, это может быть:

– Медианная длина сессии в онлайн-игре. Можно предположить, что пользователь, который долго сидит в игре – это довольный пользователь.

– Среднее количество бананов на полках во всех магазинах торговой сети в конце дня.

3 Субъективное восприятие. Оценка специально нанятыми людьми – асессорами. Например, так можно оценивать, получилось ли у нас улучшить качество машинного перевода или релевантность выдачи в поисковой системе.

4 Офлайн (offline) метрики могут быть измерены до введения модели в эксплуатацию, например, по историческим данным. В задачах, в которых нужно предсказывать какой-то конкретный таргет, офлайн метрики обычно оценивают отклонение предсказаний модели от истинных значений таргета. Например, это может быть точность предсказания, то есть число верно угаданных значений, или среднеквадратичное отклонение.

Требования к моделям

Это не только точность, но и другие параметры, важные для реальных условий

- например, работа в реальном времени. Заметим, что это требование не только к модели, но и к её реализации, а также к тому железу или к тем серверам, на которых она работает.

- модель достаточно компактна, чтобы помещаться на мобильном телефоне или другом устройстве.

- понимание ответов модели. Например, дадут ли кредит или будет ли согласовано дорогостоящее лечение. Такое требование является частным случаем более общего понятия интерпретируемости модели.

- Предсказания модели не дискриминируют какую-либо категорию пользователей. Например, если двум людям с одинаковой и достаточно длинной историей просмотров онлайн-кинотеатр рекомендует разные фильмы только из-за того, что у них разный пол, то это не здорово.