Прикладной Data Mining

Крот Александр, Lead Data Scientist

Замечания

- Лекция носит обзорный и ознакомительный характер для людей, впервые знакомящихся с методами прикладного анализа данных
- Информация в этом курсе не претендует на научную строгость и полноту изложения
- Часть информации к моменту рассказа может быть немного устаревшей, за более подробной и актуальной информацией лучше обращаться к документации

План

Теория

- Примеры прикладных задач анализа данных
- Основы машинного обучения
- Инструменты анализа данных
- Разработка продукта: бизнес-процесс
- Подготовка специалистов в областях Data Science и Big Data

Практические кейсы

- Примеры решения прикладных задач анализа данных
  - Предсказание судьбы пассажиров Титаника
  - Кластеризация городов России по социально-экономическим показателям
  - Классификация отзывов к фильмам
  - Поиск необычных моделей Playboy

Примеры прикладных задач анализа данных

Примеры прикладных задач анализа данных

- **Сферы**
  - Финансовые организации, ритейл, телеком, медиа, социальные сети, медицина, урбанистика, Интернет - компании
- **Данные**
  - Обработка естественного языка, анализ социальных сетей, анализ изображений и видео, анализ аудио сигналов, рекомендательные системы, поиск ассоциативных правил
- **Кейсы**
  - Поиск спама, рекомендательные системы, сегментация потребителей, выявление фрода, прогнозирование оттока, распознавание речи, классификация изображений

Примеры прикладных задач анализа данных: сферы

- **Финансовые организации** (Bank of America, Citigroup, Сбербанк, HomeCredit)
- **Ритейл** (Amazon, Target, Metro, Лента)
- **Телеком** (Vodafone, China Mobile, Вымпелком, МТС, Мегафон)
- **Социальные сети** (Facebook, Baidu, ВКонтакте, Одноклассники)
- **Медицина** (Enlitic, Lumiata, Numerate)
- **Урбанистика** (Uber, ГенПлан, РЖД, ДИТ Правительства Москвы)
- **Интернет-компании** (Google, Facebook, Яндекс, Mail.ru,)

Примеры прикладных задач анализа данных: данные

- **Обработка естественного языка (Natural Language Processing)**
  - Машинный перевод, анализ отзывов, выделение названий, логические выводы
- **Анализ социальных сетей (Social Network Analysis)**
  - Рекомендация друзей, поиск сообществ, выделение лидеров мнения
- **Анализ изображений и видео (Computer Vision)**
  - Выделение лиц на изображениях, извлечение номеров, названий с камер

- **Анализ аудио сигналов (Signal Processing)**
  - Распознавание речи, классификация музыки, рекомендация плейлиста
- **Рекомендательные системы (Recommended Systems)**
  - Рекомендация товаров, друзей, прогнозирование оценок к фильмам
- **Поиск ассоциативных правил (Association Rule Learning)**
  - Построение логических правил, анализ чеков

Примеры прикладных задач анализа данных: кейсы

- **Поиск спама (Spam Detection)**
  - Gmail, Mail.ru, Яндекс.Почта, …
- **Рекомендательные системы (Product Recommendation)**
  - Netflix, Amazon, Ozon, RetailRocket, Facebook, …
- **Сегментация потребителей (Customer Segmentations)**
  - Facebook, Google, Яндекс, Вымпелком, …
- **Выявление фрода (Fraud Detection)**
  - Google, Facebook, Вымпелком, Сбербанк, …

- **Прогнозирование оттока (Churn Prediction)**
  - Amazon, Netflix, Вымпелком, МТС, Мегафон, МГТС, …
- **Распознавание речи (Speech Understanding)**
  - Apple (Siri),
- **Классификация изображений (Image Understanding)**
  - Facebook, Google, Instagram, Яндекс, Mail.ru, …

А также множество других задач, многие из которых в стадии Research

Основы машинного обучения

Основы машинного обучения

- Понятия объекта и признаков
- 3 классические задачи машинного обучения
- Извлечение, отбор и преобразование признаков
- Алгоритмы машинного обучения
- Метрики качества алгоритмов классификации

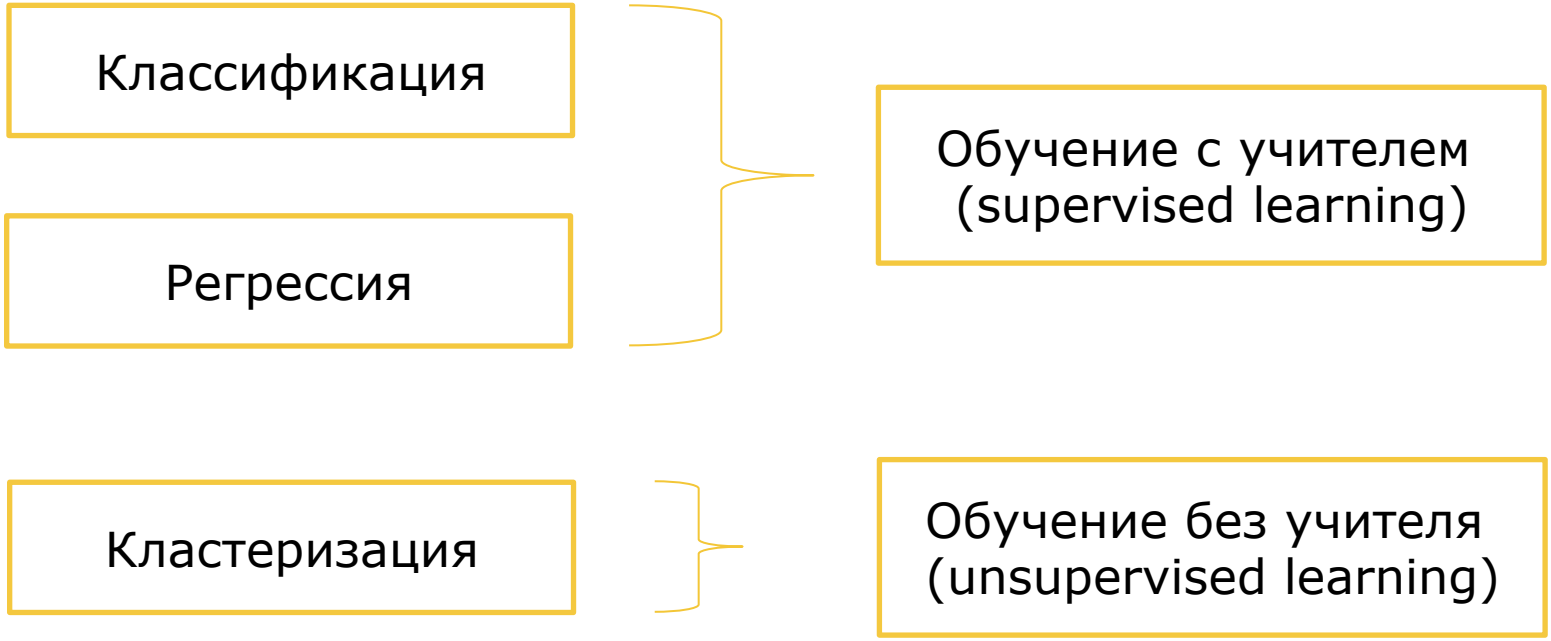
Объекты и признаки

- **Объекты и признаки зависят от рассматриваемого контекста (задачи)**
  - **B2C абонент сотового оператора**
    - ARPU, модель телефона, количество финансовых блокировок, количество входящих звонков, средняя сумма пополнения баланса
  - **Пара сим-карт**
    - Количество общих контактов, модели телефонов, количество совместных регистраций на базовой станции
  - **Электронное сообщение**
    - Длина сообщения, наличие цифр, количество слов, предложений, сами слова
  - **Ресторан**
    - Средний чек, количество посетителей за месяц, район, количество официантов

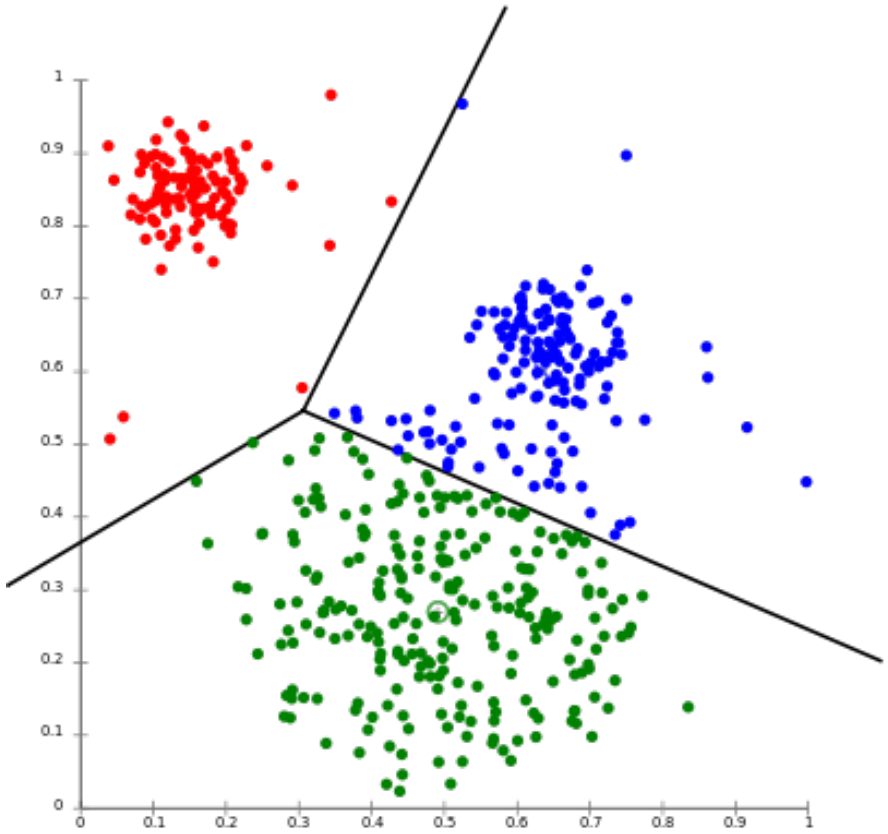
3 классические задачи

- Машинное обучение **умеет**:
  - Получив **объекты** (человек, изображение, сим-карта, цветок, пара людей)
  - **Получить** или **извлечь признаковое описание** объектов (рост, цвет волос, размер одежды, количество детей, образование, наличие смартфона)
  - Посмотрев на объекты, научиться:
    - **Классифицировать** (Мужчина/Женщина)
    - **Прогнозировать значения для объектов** (Возраст, Доход, Рост)
    - **Группировать** (Школьники, Бизнесмены, Политики, Любители Чая)

3 классические задачи



Классификация



- **Дано:**
  - **Обучающая выборка**, состоящая из **признакового описания** объектов и **метки класса** для каждого объекта
- **Найти:**
  - **Алгоритм**, который бы для каждого нового объекта по его признаковому описанию прогнозировал класс этого объекта

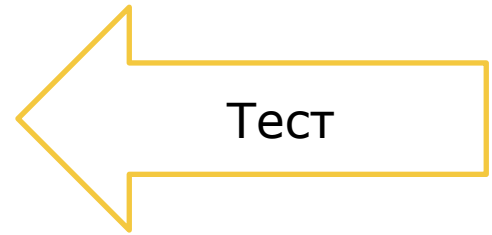
Геометрически алгоритм строит **разделяющую** поверхность, отделяющую объекты разных классов

Классификация

ID	ARPU	Кол-во блокировок	Кол-во входящих звонков	ОТТОК
9034948911	123.5	5	15	1
9034948912	245.6	10	124	0
9034948913	890.4	0	23	1
9034948914	50.3	101	0	0

АЛГОРИТМ: **ВХОД** - [ID, ARPU, Кол-во блокировок, Кол-во входящих звонков], **ВЫХОД** – [ОТТОК]

903494895	12.6	8	12	?
9034948916	1012.2	10	256	?
9034948917	132.9	112	10	?

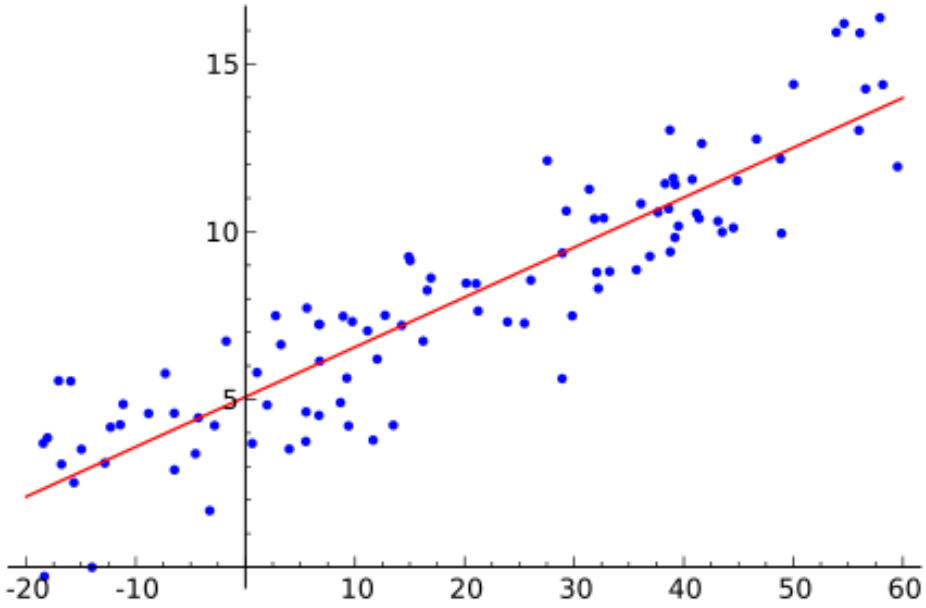


Классификация

- **Важно помнить:**
  - Классификация – это **обучение с учителем (supervised learning)**, в роли учителя выступает **обучающая выборка**
  - Классификация прогнозирует **метку (класс)** для объекта, который может принимать набор **дискретных** значений
  - В результате получается **алгоритм**, который на вход принимает **признаковое описание** объекта, а на выходе выдает его **класс**

- **Примеры задач:**
  - Классификация абонентов по полу, классификация спама, классификация абонентов на наличие второго устройства

Регрессия



Геометрически, алгоритм восстанавливает **зависимость** между **признаками** и **целевой переменной**

- **Дано:**
  - **Обучающая выборка**, состоящая из **признакового описания** объектов и **значения целевой переменной** для каждого объекта
- **Найти:**
  - **Алгоритм**, который бы для каждого нового объекта по его признаковому описанию прогнозировал **целевую переменную** этого объекта

Регрессия

ID	ARPU	Модель телефона	Интернет-трафик	ДОХОД
9034948911	123.5	Samsung	1500.4	10000
9034948912	245.6	iPhone 6	1124.7	25000
9034948913	890.4	Nokia	2312.6	135000
9034948914	50.3	Samsung	1321.3	90000

Обучение

АЛГОРИТМ: **ВХОД** - [ID, ARPU, Модель телефона, Интернет-трафик], **ВЫХОД** - [ДОХОД]

903494895	12.6	iPhone 5S	12123.6	?
9034948916	1012.2	HTC	13256.9	?
9034948917	132.9	Samsung	101333.1	?

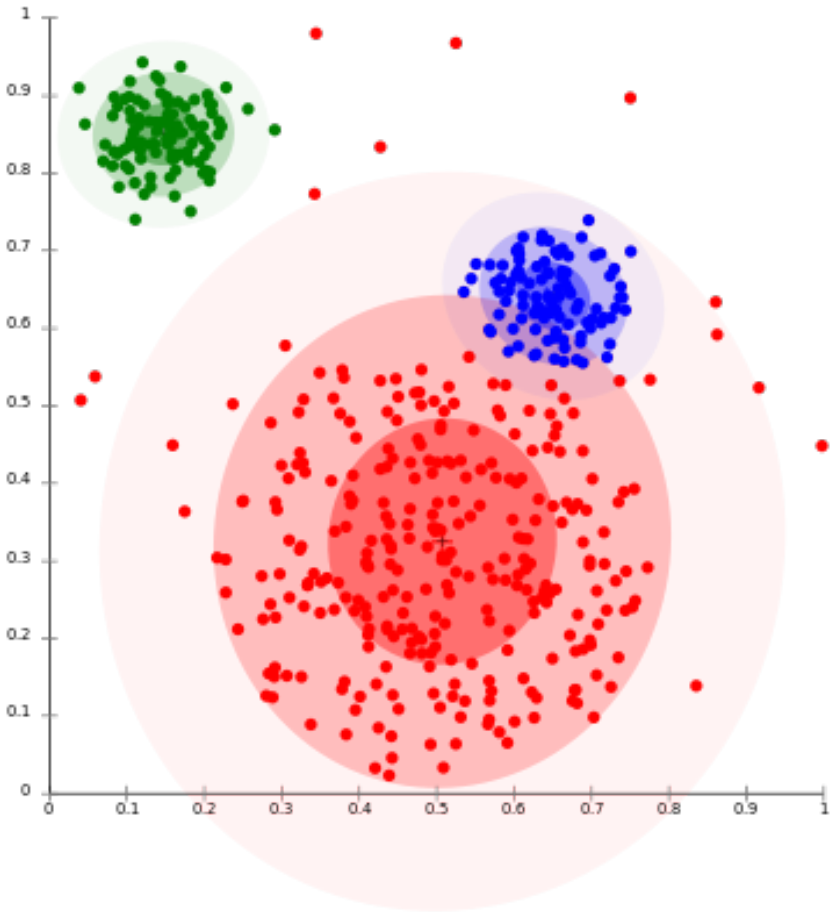
Тест

Регрессия

- **Важно помнить:**
  - Регрессия – это **обучение с учителем (supervised learning)**, в роли учителя выступает **обучающая выборка**
  - Регрессия прогнозирует **значение целевой переменной** для объекта, которая может принимать любое **действительное** значение
  - В результате получается **алгоритм**, который на вход принимает **признаковое описание** объекта, а на выходе выдает **значение целевой переменной**

- **Примеры задач:**
  - Прогнозирование дохода абонента, прогнозирование нагрузки на колл-центр, прогнозирование прибыли ресторана

Кластеризация



- **Дано:**
  - **Обучающая выборка**, состоящая из **признакового описания** объектов
- **Найти:**
  - Разделение всех объектов на **кластеры**

Геометрически, алгоритм **группирует** данные объекты в **кластеры** наилучшим образом

Кластеризация

ID	ARPU	Модель телефона	Интернет-трафик	Кол-во блокировок
9034948911	123.5	Samsung	1500.4	5
9034948912	245.6	iPhone 6	1124.7	10
9034948913	890.4	Nokia	2312.6	0
9034948914	50.3	Samsung	1321.3	101
9034948915	12.6	iPhone 5S	12123.6	8
9034948916	1012.2	HTC	13256.9	10
9034948917	132.9	Samsung	101333.1	2
9034948918	152.0	Nokia	1498.2	76
9034948919	14.6	Samsung	4135.7	54

Кластер 1

Кластер 2

Кластер 3

Кластеризация

- **Важно помнить:**
  - Кластеризация – это **обучение без учителя (unsupervised learning)**, размеченная (обучающая) выборка не нужна
  - Кластеризация **группирует** данное множество объектов на кластеры **наилучшим образом**
  - В результате получается **алгоритм**, который на вход принимает **признаковое описание** набора объектов и на выходе выдает разбиение объектов на группы

- **Примеры задач:**
  - Выделение домохозяйств среди абонентской базы, выделение сообществ, определение архетипа абонента

Извлечение, отбор и преобразование признаков

- Признаки **для простых объектов** (человек, сим-карта) берутся на основе целевой переменной:
  - Опытным путем (наверное, на доход влияет ARPU)
  - Из статей (если задача ранее решалась)

- Для **сложных объектов** (лицо на изображении, слова в тексте, номер на видео) признаки извлечь очень тяжело:
  - Либо из статей (то, что ученые придумали)
  - Либо извлекать автоматически (**Deep Learning** подход)

Человеко-читаемые, извлекаются сразу

НЕ человеко-читаемые, извлекаются с помощью сложных алгоритмов

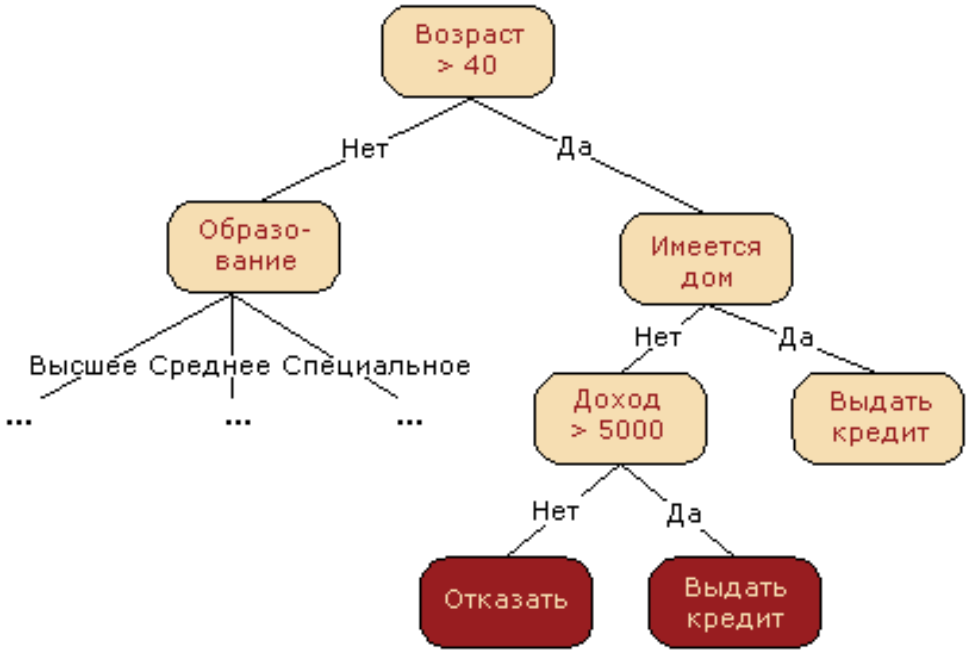
Извлечение, отбор и преобразование признаков

- **Машина – не человек:**
  - Если в качестве признака есть дата, то машина не понимает время суток
  - Если дано имя – машина не понимает, что оно женское
  - Если дан числовой признак – машина не понимает, много это или мало
  - Машина не может группировать признаки
  - Машина не различает «много» или «мало»
- **Примеры преобразования признаков:**
  - При прогнозировании спроса на вело прокат дату можно преобразовать в признаки - «утро», «день», «вечер»
  - При прогнозировании цены квартиры «длину» и «ширину» нужно преобразовать в площадь

Наиболее простые подходы к задачам машинного обучения

- **Классификация**
  - Деревья решений (**Decision Trees**), метод ближайшего соседа (**kNN**), метод опорных векторов (**SVM**)
- **Регрессия**
  - Линейная регрессия (**Linear Regression**)
- **Кластеризация**
  - **KMeans**, иерархическая кластеризация (**Hierarchical Clustering**)

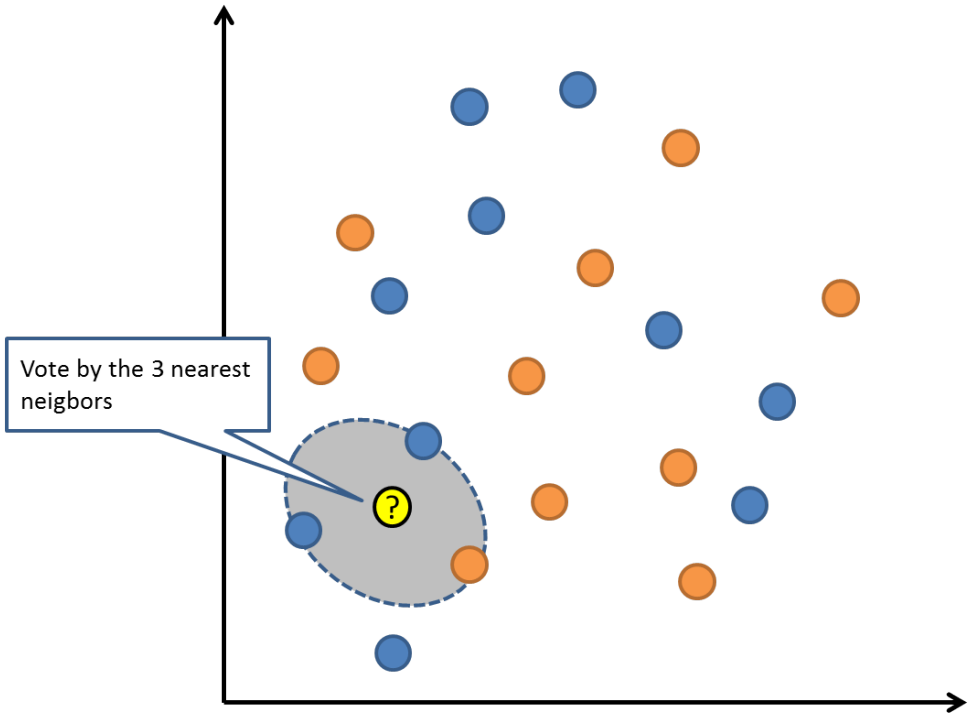
Классификация: деревья решений



Наиболее часто используемый алгоритм в медицине и банковском скоринге **ввиду человеко-читаемости**

- **Идея:**
  - Пытаемся оптимальным образом построить **дерево** так, чтобы объекты **обучающей выборки** классифицировались **максимально правильно**
- **Результат:**
  - Для каждого **нового** объекта сможем пройтись по дереву и классифицировать объект, выдав при этом **причину** классификации

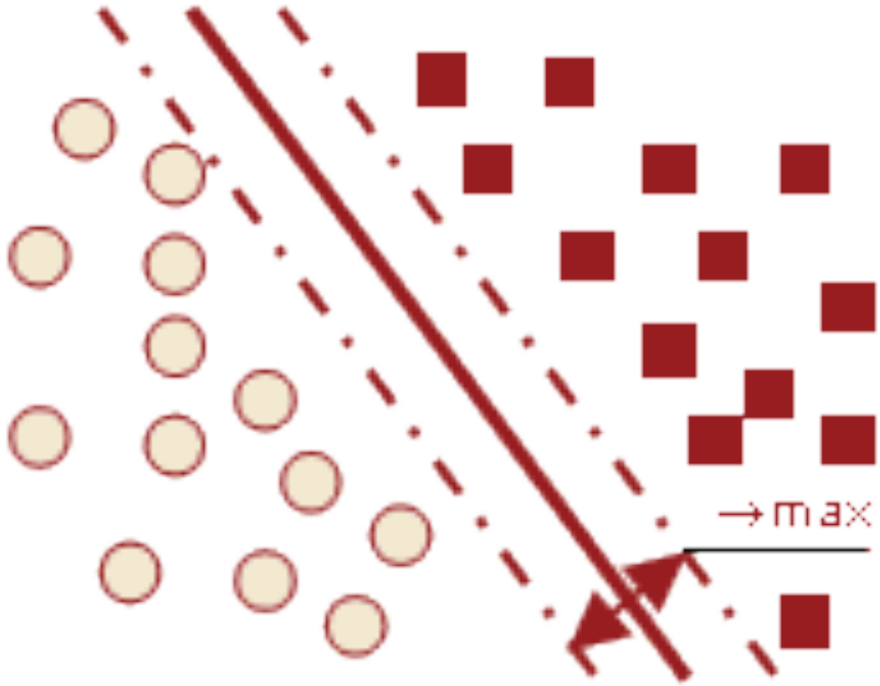
Классификация: метод ближайшего соседа



- **Идея:**
  - Наверное, новый объект такого же класса, как и его окружение
- **Результат:**
  - Для каждого **нового** объекта смотрим его окружение и говорим, на кого он больше похож

Алгоритм **не используется в продуктивных задачах**, т.к. для каждого нового объекта мы должны искать ближайших – **это долго**

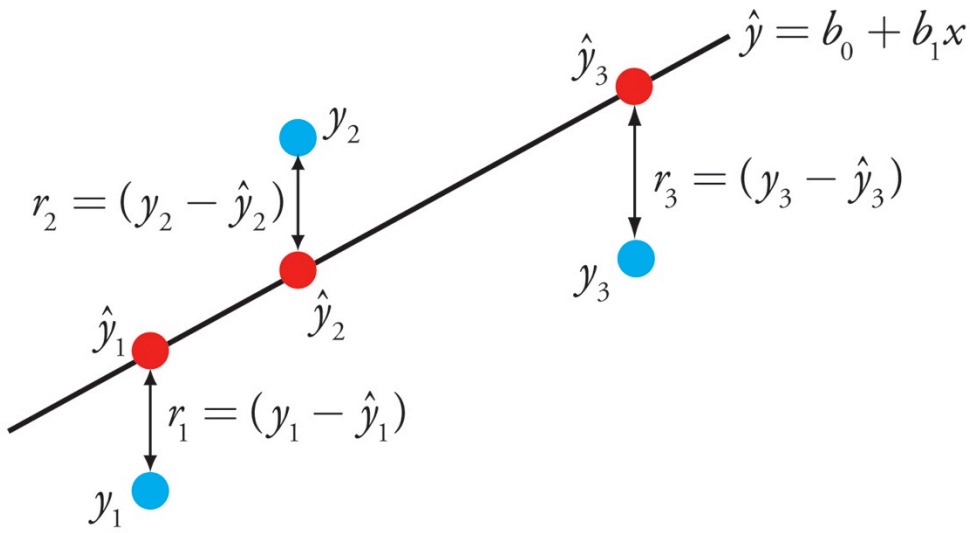
Классификация: метод опорных векторов



Один из самых **распространенных** алгоритмов классификации, ввиду своей **гибкости**

- **Идея:**
  - Пытаемся провести **разделяющую поверхность** так, чтобы максимизировать зазор между объектами **обучающей выборки** разных классов
- **Результат:**
  - Для каждого **нового** объекта смотрим, с какой стороны от разделяющей поверхности он лежит, тем самым, классифицирую объект

Регрессия: линейная регрессия



Линейные модели допускают **гибкую настройку и большое количество эвристик**. Настраивать **сложно**, но алгоритмы наиболее подходят для **продуктивных решений** ввиду своей **простоты**

- **Идея:**
  - Метод наименьших квадратов, известный со школы
  - Ищем значение целевой переменной в виде **линейной комбинации** признаков
- **Результат:**
  - Для каждого **нового** объекта смотрим по восстановленной зависимости (формуле) считаем значение целевой переменной

Кластеризация: KMeans

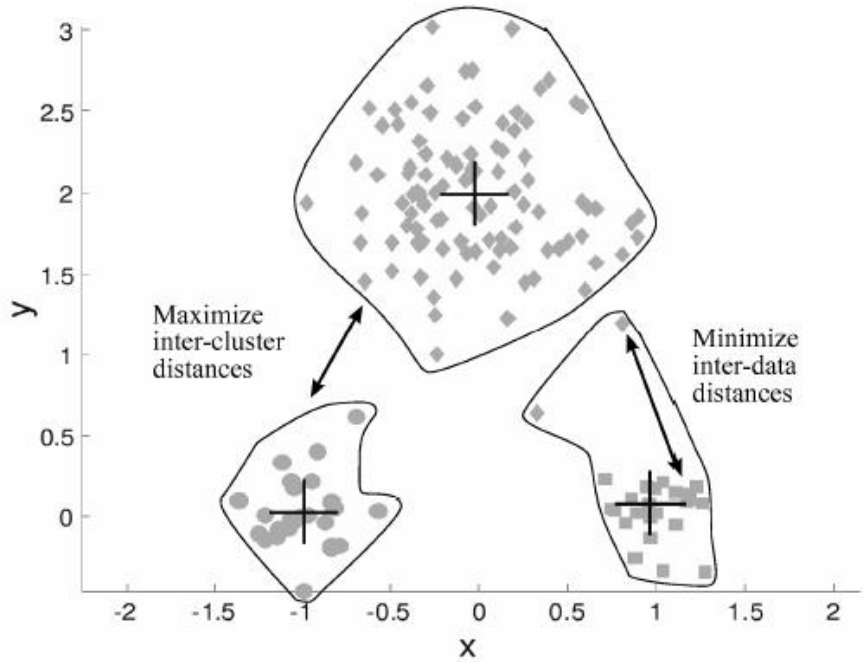
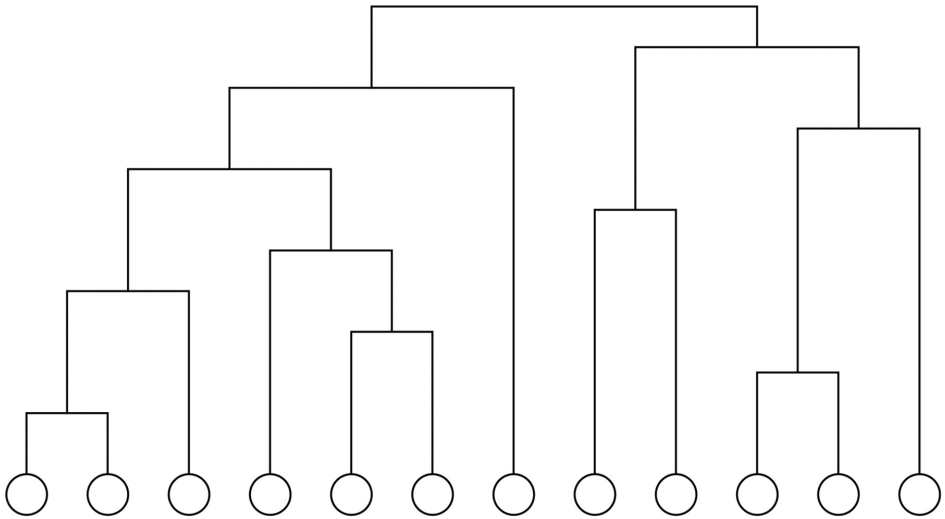


Fig. 2. Grouping of observations by k-means algorithm

KMeans является **наиболее распространенным** методом кластеризации, однако важно правильно определить **метрику** – расстояние между объектами

- **Идея:**
  - Задаем **количество кластеров**
  - Задаем **центры кластеров**
  - Каждый объект принадлежит к тому кластеру, центр которого ближе
  - Уточняя центры кластеров находим оптимальное разбиение
- **Результат:**
  - Наиболее оптимальное разбиение **данных объектов** на кластеры

Кластеризация: иерархическая кластеризация



Иерархическая кластеризация **более гибкая**, чем KMeans в бизнес приложениях, но и **более чувствительна к настройкам** параметров

- **Идея:**
  - Изначально каждый объект – отдельный кластер
  - Постепенно **объединяем** похожие кластеры между собой **на основе метрики схожести**
- **Результат:**
  - **Дендрограмма** – иерархическое древовидное представление кластеризации

Метрики качества алгоритмов классификации

Бинарная классификация		Истинные значения	
		1	0
Результат алгоритма	1	TP	FP
	0	FN	TN

Доля

$$\text{Accuracy} = \frac{TP + TN}{TP + FP}$$

Точность

$$\text{Precision} = \frac{TP}{TP + FP}$$

Полнота

$$\text{Recall} = \frac{TP}{TP + FN}$$

F-мера

$$F = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **True Positive**
  – верно угадали 1
- **True Negative**
  – верно угадали 0
- **False Positive**
  – ошибка первого рода
- **False Negative**
  – ошибка второго рода

Инструменты анализа данных

Инструменты для анализа данных

- Инструменты анализа **Small Data**
  - Интерактивные инструменты: **RapidMiner**, **Weka**
  - Скриптовые языки: **Python** (**pandas**, **scikit-learn**), **R** (**DataFrame**, **caret**)
- Инструменты анализа **Big Data**
  - Введение в стек технологий **Hadoop**
  - Первичный анализ данных: **Hive**, **Pig**
  - Машинное обучение: **Apache Spark**, **Vowpal Wabbit**
- Порядок использования инструментов в конкретных задачах

Инструменты анализа Small Data: интерактивные инструменты - RapidMiner

Подготовка данных (ETL)

Первичный анализ (Data Mining)

Машинное обучение

- Хороший GUI
- Упрощает ETL - операции
- Расширяемость (R-scripts)
- Open-source решение
- Реализовано множество алгоритмов машинного обучения
- Минимальный порог входа для аналитика

Инструменты анализа Small Data: интерактивные инструменты - Weka

Первичный анализ (Data Mining)

Машинное обучение

- Относительно хороший GUI
- ETL – операции не гибки
- Расширяемость сложна
- Полностью Open-Source
- Реализовано множество алгоритмов машинного обучения
- Средний порог входа для аналитика

Инструменты анализа Small Data: скриптовые языки - Python

Подготовка данных (ETL)

Первичный анализ (Data Mining)

Машинное обучение

- Есть подобие GUI – **Jupyter**
- Максимальная гибкость ETL – операций (**pandas**)
- Расширяемость за счет библиотек
- Полностью Open-Source
- Реализовано множество алгоритмов машинного обучения (**scikit-learn**)
- Высокий порог входа для аналитика

Инструменты анализа Small Data: скриптовые языки - R

Подготовка данных (ETL)

Первичный анализ (Data Mining)

Машинное обучение

- Хорошее GUI – **RStudio**
- Максимальная гибкость ETL – операций (**DataFrame**)
- Расширяемость за счет библиотек
- Полностью Open-Source (+ коммерческая реализация)
- Реализовано множество алгоритмов машинного обучения (**caret**)
- Высокий порог входа для аналитика

Инструменты анализа Small Data: сравнение

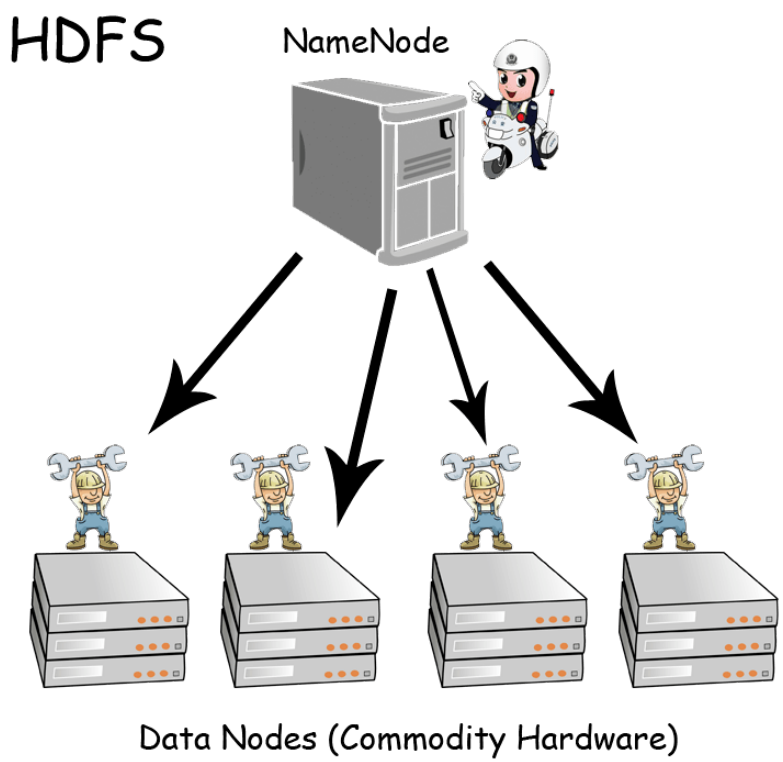
Продукт	GUI	ETL	Машинное обучение	Расширяемость	Порог входа	Цена
RapidMiner	10	10	8	7	3	Free
Weka	7	5	7	3	5	Free
Python	9	10	10	10	9	Free
R	10	10	9	9	10	Free

- На практике обычно **простой анализ** делается с помощью интерактивных инструментов **RapidMiner или Weka**, более **сложный** – с помощью **Python или R** с подключением дополнительных библиотек
- Ценность инструментов от **IBM**, **SAS** заключается в проприетарной поддержке

Введение в стек технологий Hadoop

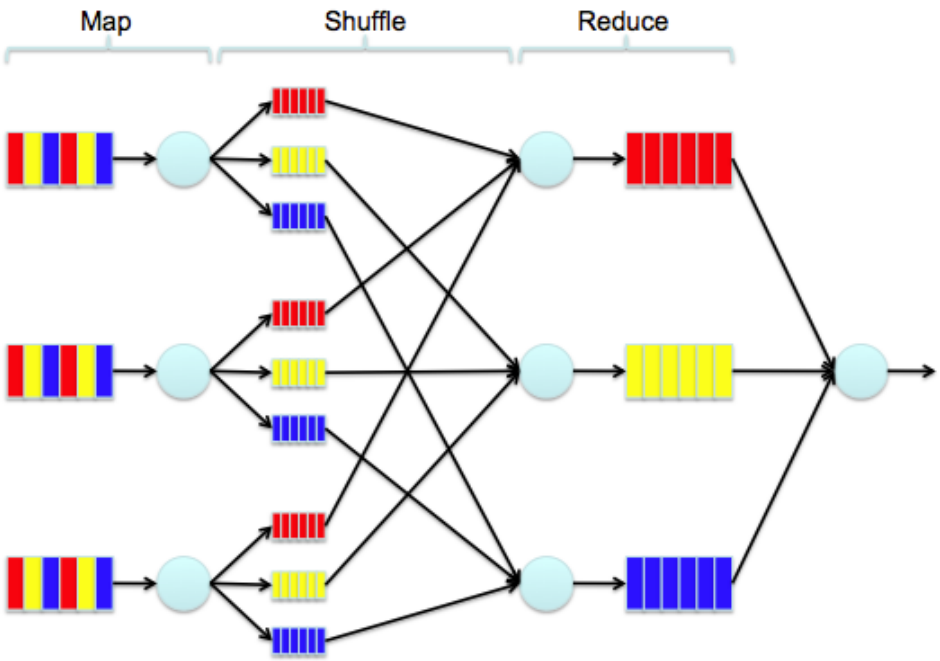
- Когда говорят о **Hadoop**, в первую очередь подразумевают:
  - Распределенную файловую систему **HDFS**
  - Модель вычислений **MapReduce**
- Также в стек технологий Hadoop входит большое кол-во дополнительных инструментов:
  - **ZooKeeper**, **Oozie**, **Azkaban** — координация процессов и планирование потоков задач
  - **Sqoop**, **Flume**, **Kafka** — импорт и экспорт данных
  - **Hbase**, **Cassandra** – noSQL базы данных
  - **Hive**, **Pig** – обработка данных, adhoc-запросы

Hadoop Distributed File System (HDFS)



- Данные хранятся распределенно на большом количестве машин
- Имеется **NameNode** – главная машина (может быть не одна), на которой хранится вся метаинформация
- Имеется большое количество **DataNode**, на которых хранятся непосредственно блоки данных
- Линейный рост мощности (добавление новых **DataNode**)
- Устойчивость и безопасность хранения достигается благодаря репликации

Модель вычислений MapReduce



- Данные хранятся в **HDFS** в виде файлов (построчно)
- Большинство запросов можно вычислить с помощью двух операций
- **Map**:
  - Input record => (key, value)
- **Reduce**:
  - (key, {v1,…,vn}) => output record

Инструменты анализа Big Data: Data Mining - Hive

```
-- import the file as lines
CREATE EXTERNAL TABLE lines(line string)
LOAD DATA INPATH 'books' OVERWRITE INTO TABLE lines;

-- create a virtual view that splits the lines
SELECT word, count(*) FROM lines
      LATERAL VIEW explode(split(text, ' ' )) lTable as word
      GROUP BY word;
```

Adhoc - запросы

Подготовка данных

- **SQL**-подобный синтаксис
- Слабая гибкость
- Расширяемость за счет пользовательских функций
- Полностью Open-Source (входит в стек Hadoop)
- Низкий порог входа для аналитика

Инструменты анализа Big Data: Data Mining - Pig

- Sample Pig script:

```
stock = LOAD '/user/fred/stock' AS (id, item);
orders= LOAD '/user/fred/orders' AS (id, cost);
grpd = GROUP orders BY id;
totals = FOREACH grpd GENERATE group,
SUM(orders.cost) AS t;
result = JOIN stock BY id, totals BY group;
DUMP result;
```

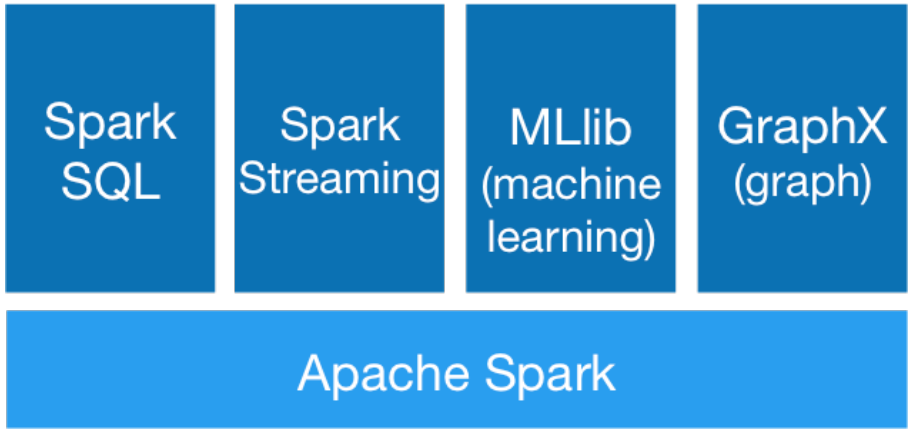
- We will investigate Pig in more detail later in the course

Подготовка данных (ETL)

Adhoc - запросы

- Собственный процедурный язык (Pig Latin)
- Достаточно гибкие запросы
- Расширяемость за счет пользовательских функций
- Полностью Open-Source (входит в стек Hadoop)
- Средний порог входа для аналитика

Инструменты анализа Big Data: машинное обучение – Apache Spark



Adhoc – запросы

Первичный анализ (Data Mining)

Машинное обучение

- Имеет streaming для Python, Scala, Java
- Вычисления im-memory
- Высокая гибкость
- Расширяемость практически не требуется
- Полностью Open-Source
- Высокий порог входа для аналитика

Инструменты анализа Big Data: машинное обучение – Vowpal Wabbit

Машинное обучение

- Предназначен для машинного обучения на больших данных **без кластера**
- Для работы используется **консоль**
- Высокая гибкость
- Базового функционала хватает
- Полностью Open-Source, коммерческого происхождения
- Средний порог входа для аналитика

Разработка продукта: бизнес-процесс

Разработка продукта: бизнес-процесс

- Идея
- Бизнес-требования
- Формальные требования
- Определение необходимых данных
- Выгрузка данных
- Предобработка данных
- Построение модели, тестирование
- «ВАУ» - эффект на пилоте
- Разработка продуктивного решения

Разработка продукта: бизнес-процесс

Идея	Бизнес-требования	Формальные требования
Определение необходимых данных	Выгрузка данных	Предобработка данных
Построение модели	«ВАУ» - эффект на пилоте	Разработка продуктивного решения

Идея

Идея

- Идеи обычно генерирует бизнес
- Генерирует, не думая
- Идеи фантастические, часто не реализуемые
- Задачи, которые можно решать хорошо – известны давно
- Кроме телекома – там почти все задачи новые и в книгах их нет=)

- Что такое отток клиентов?
- Хочется находить абонентов со вторыми устройствами
- Неплохо было бы знать пол абонента
- Ну и возраст заодно
- А еще хорошо сделать рекомендательную систему для IPTV

Бизнес-требования

Бизнес-требования

- Рассчитывается кейс
- Как правило, рассчитывается на основе кампании
- Определяется то, что хочется на выходе
- **Бизнес решает «делать или нет»!**

- Что такое отток клиентов?
- Как будет планироваться кампания по удержанию?
- Сколько времени занимает компания?
- Насколько критична «ложная тревога»?
- Как должно выглядеть продуктивное решение?
- Каков бизнес – эффект от кейса?

Формальные требования

Формальные требования

- Четко определено целевое действие
- Понятно, к какому качеству стремиться
- Понятно, какие данные придется выгружать, какие признаки генерировать
- **Можно назвать «сроки»!**

- Отток – «не звонил X раз, не писал смс Y раз за последние Z дней»
- После прогнозирования оттока «в течении X дне будет проводиться коммуникация с клиентов посредством канала Y»
- Компания занимает X дней с момента выгрузки отчета
- Точность должна быть не менее X, полнота должна быть не менее Y
- В продуктивном решении отчет выгружается раз в X дней

Определение необходимых данных

Определение необходимых данных

- Понятно, сколько придется писать код
- Понятно, сколько придется по времени выгружать данные
- **Пока еще не понятны признаки, которые нужны в задаче**

- Для оттока скорее всего нужны агрегированные показатели по активности – звонки, смс
- Скорее всего также потребуются агрегированные данные по трафику
- Наверное, также потребуется обработка обращений
- Звонки на номера конкурентов тоже потребуются
- И что-то еще, но пока возьмем то, что есть
- Где брать «обучающую» выборку?

Выгрузка данных

Выгрузка данных

- Источники данных - разные
- Пишем код
- Тестируем
- Не все сразу работает, на больших данных часто что-то «падает»
- Запускаем и ждем

- На выходе разнородные, не очищенные, часто не правильные данные
- **Over X файлов на выходе, которые нужно предобработать**

Предобработка данных

Предобработка данных

- Из сырых данных генерируем признаки (часто берем лишние с запасом)
- Удаляем выбросы
- Пропущенные значения либо удаляем, либо думаем, как заполнять
- Подготавливаем матрицу «объект-признак»

- На выходе имеет данные в формате, пригодном для разработки алгоритма
- **Фактически, теперь мыслим в терминах «объект-признак», ведь матрица уже есть**

Построение модели

Построение модели

- Проверяем простые гипотезы – смотрим корреляцию, зависимости между признаками
- Отбираем признаки, генерируем новые, сокращаем размерность
- Строим несколько моделей
- Сравниваем между собой
- Выбираем лучший алгоритм

- На выходе имеем законченную модель, протестированную на обучающей выборке
- **Ограничения продукта на данном этапе видны еще не окончательно!**

«ВАУ» - эффект на пилоте

«ВАУ» - эффект на пилоте

- Возвращаемся к заказчику
- Просим несколько абонентов
- Угадываем
- Подписываем чек
- Начинаем внедрять в продуктив

- Фактически, прошла продажа, результат утвержден
- **Можно работать над продуктивным решением**

Разработка продуктивного решения

Разработка продуктивного решения

- Продуктивное решение важно сделать максимально простым с точки зрения поддержки
- Продуктивное решение тоже нужно тестировать

- Определяем самый простой путь от сырых данных до нужных признаков
- Создаем новые «витрины» данных при необходимости
- Пишем максимально качественный код
- Готовим презентации, отдаем результат

Подготовка специалистов в областях Data Science и Big Data

- Математическая подготовка
- Навыки программирования
- Машинное обучение
- Практика построения алгоритмов
- Знакомство с методами хранения и обработки больших данных
- Знакомство с вопросами архитектуры и Real-Time методами анализа данных

Математическая подготовка

- Важно развивать в себе математическое мышление:
  - Работа с дискретными объектами
  - Алгоритмическое мышление
  - Понимание понятия «вероятность»

Навыки программирования

- Навыки программирования в **Data Science**:
  - Минимальные (**для аналитиков** и тех, кто строит модели)
    - **Python, R, SQL** – на этапе тестирования гипотез и разработки алгоритмов машинного обучения
  - Требовательные (**для разработчиков** продуктивных решений)
    - **Java, Scala, C++** - в зависимости от архитектуры конечного продуктивного решения

Машинное обучение

- Важно понимать, какие задачи есть, какие решаются хорошо, какие – пока плохо
- Важно научиться правильно и эффективно извлекать признаки из сложных объектов
- Важно понимать, как выбирать модели, тестировать алгоритмы, какие метрики используются
- Важно научиться объяснять результаты простыми словами

Практика построения алгоритмов

- Важно наступить на все грабли, которые не описаны в книгах:
  - Где и как искать обучающую выборку?
  - Какие признаки использовать для конкретной задачи?
  - Как лучше разбивать выборку на тестовую и обучающую?
  - Какие алгоритмы больше подходят для классификации текстов?
  - Что делать, если объектов не достаточно для построения алгоритма?
  - И множество других вопросов, ответы на которые можно найти только решив большое кол-во задач

Знакомство с большими данными

- Важно понимать, как хранятся и обрабатываются большие данные
- Надо научиться мыслить в парадигме **MapReduce**
- Необходимо понимать, как существующие алгоритмы машинного обучения реализуются в модели вычислений **MapReduce**
- Важно научиться импортировать модели на кластер и запускать на большой тестовой выборке
- Необходимо научиться работать с инструментами **Hive**, **Pig**, **Apache Spark**

Знакомство с вопросами архитектуры и Real-Time методами анализа данных

- Важно отличать в продуктивном решении **batch**-обработку данных (анализ истории) от real-time (**mini-batch** либо **транзакции**)
- Необходимо понимать, как алгоритмы могут «**дообучаться**» в режиме реального времени
- Важно правильно строить архитектуру обработки данных
- Нужно быть знакомым с инструментами обработки данных в реальном времени (**Apache Spark**, **Apache Storm**)
- Очень важно оценивать нагрузку на сервис

Примеры решения прикладных задач

Примеры решения прикладных задач

- Предсказание судьбы пассажиров Титаника
- Кластеризация городов России по социально-экономическим признакам
- Классификация отзывов к фильмам
- Поиск необычных моделей Playboy

Предсказание судьбы пассажиров Титаника

Оригинал:
habrahabr.ru/post/202090/

Предсказание судьбы пассажиров Титаника

- Имеются **пассажиры (объекты)**, описываемые **признаками**:
  - **PClass** – социально-экономический статус (высокий, средний, низкий)
  - **Name** – имя пассажира
  - **Sex** – пол пассажира
  - **Age** – возраст
  - **SibSp** – кол-во родственников 2го порядка (муж, жена, братья) на борту
  - **Parch** – кол-во родственников 1го порядка (мать, отец, дети) на борту
  - **Ticket** – номер билета
  - **Fare** – цена билета
  - **Cabin** – номер каюты
  - **Embarked** – порт посадки (C – Cherbourg, Q – Queenstown, Sa - Southampton)

Для каждого пассажира необходимо узнать – утонет он или нет?

Смотрим на данные

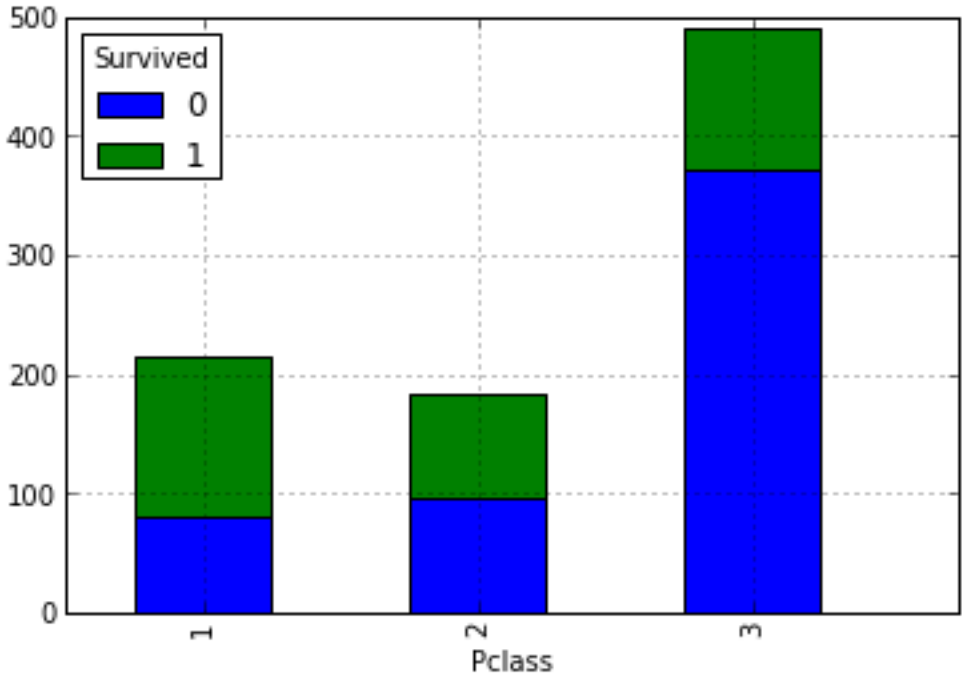
ID	PClass	Name	Sex	Age	SubSp	Parch	Ticket	Fare	Cabin	Embarked	Survived
1	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.2500	**NaN**	S	0
2	1	Cumings, Mrs. John Bradley	female	38	1	0	PC 17599	71.2833	C85	C	1
3	3	Heikkinen, Miss. Laina	female	26	0	0	STON/ O2. 3101282	7.9250	**NaN**	S	1
4	1	Futrelle, Mrs. Jacques Heath	female	35	1	0	113803	53.1000	C123	S	**?**
5	3	Allen	male	35	0	0	373450	8.0500	**NaN**	S	**?**

- Обучающая выборка содержит из **890** объектов
- Часть данных пропущена

Бинарная классификация

Тестируем простые гипотезы - 1

- **Гипотеза 1**: чем выше социальный статус (**PClass**), тем прочнее каюта, тем больше вероятность спасения человека

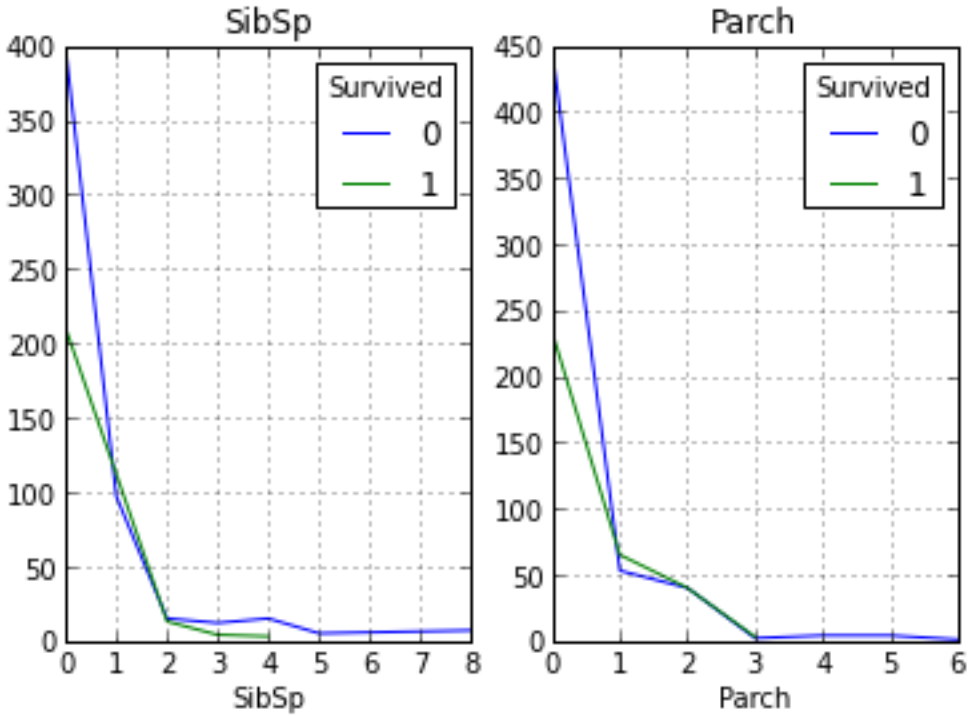


- В целом, почти верно
- Можно добавить новый признак - «Pclass = 3», характеризующий тех, кто с большой вероятностью спасется

До появления машинного обучения все примерно так и работало

Тестируем простые гипотезы - 2

- **Гипотеза 2**: человек будет спасать своих родственников при крушении. Посмотрим, как влияет количество родственников на вероятность спасения



- Действительно, из тех, кто имеют родственников, спаслись не многие
- Можно добавить новые признаки – «**кол-во родственников больше X**»

Заполняем пропуски, отбираем признаки

- Данные о каютах (**Cabin**) заполнены на 204 / 890 **< 23% - можно отбросить**
- Данные о возрасте (**Age**) заполнены на 713 / 890 **> 80% - надо научиться заполнять!**
  - Можно заполнить средним или медианным значением
- Есть 2 человека, у которых не известен порт посадки (**Embarked**)
  - С большой вероятностью они сели там же, где и большинство людей
- Наверное, номер билета и имя пассажира не влияют на вероятность спасения (в нумерологию не верим, хотя, интересно проверить)

Преобразование категориальных признаков

- Признаки **Sex** (пол) и **Embarked** (порт посадки) – категориальные
- Большинство алгоритмов машинного обучения работают с числовыми признаками
- Сделаем следующее преобразование:
  - Заменим пол (**Sex**) на **1** (мужчина) и **0** (женщина)
  - Порт посадки заменим на **0** (Cherbourg), **1** (Queenstown), **2** (Southampton) – **почему это не совсем верно?**

Все описанные выше операции применяем и к тестовой выборке

Пред обработанные данные

- После всех преобразований, наша **обучающая** выборка готова для тестирования алгоритмов машинного обучения

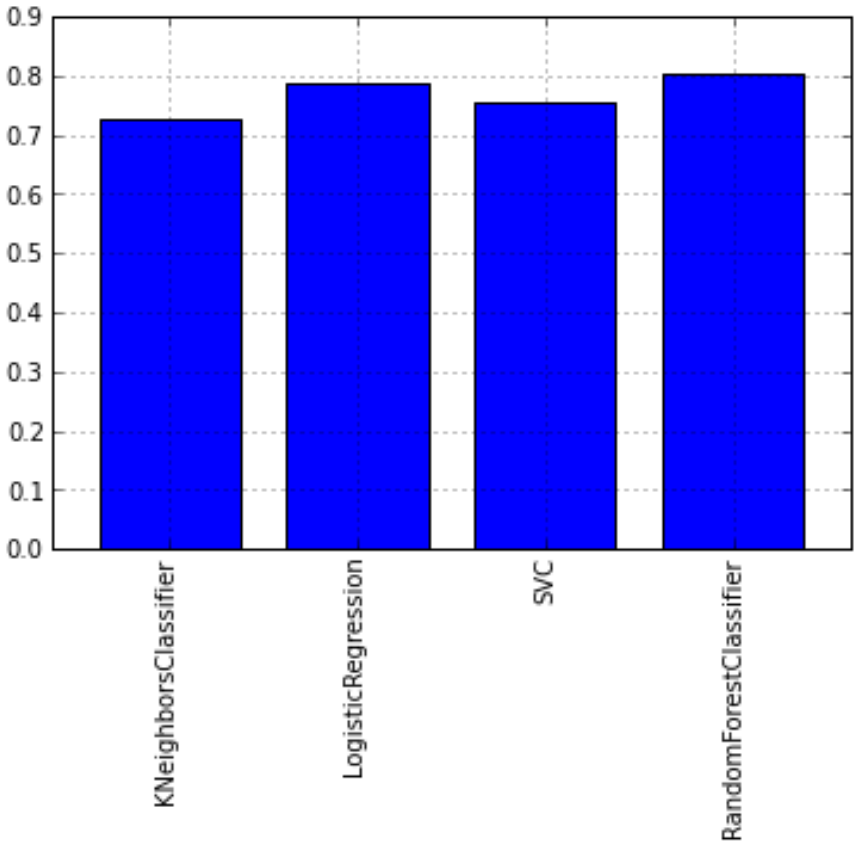
ID	PClass	Sex	Age	SubSp	Parch	Fare	Embarked	Survived
1	3	1	34.5	0	0	7.8292	1	0
2	3	0	47.0	1	0	7.0000	2	1
3	2	1	62.0	0	0	9.6875	1	1
4	3	1	27.0	0	0	8.6625	2	**?**
5	3	0	22.0	1	1	12.2875	2	**?**

Обучающая и тестовая выборки отличаются одним столбцом

Планирование методик тестирования алгоритма

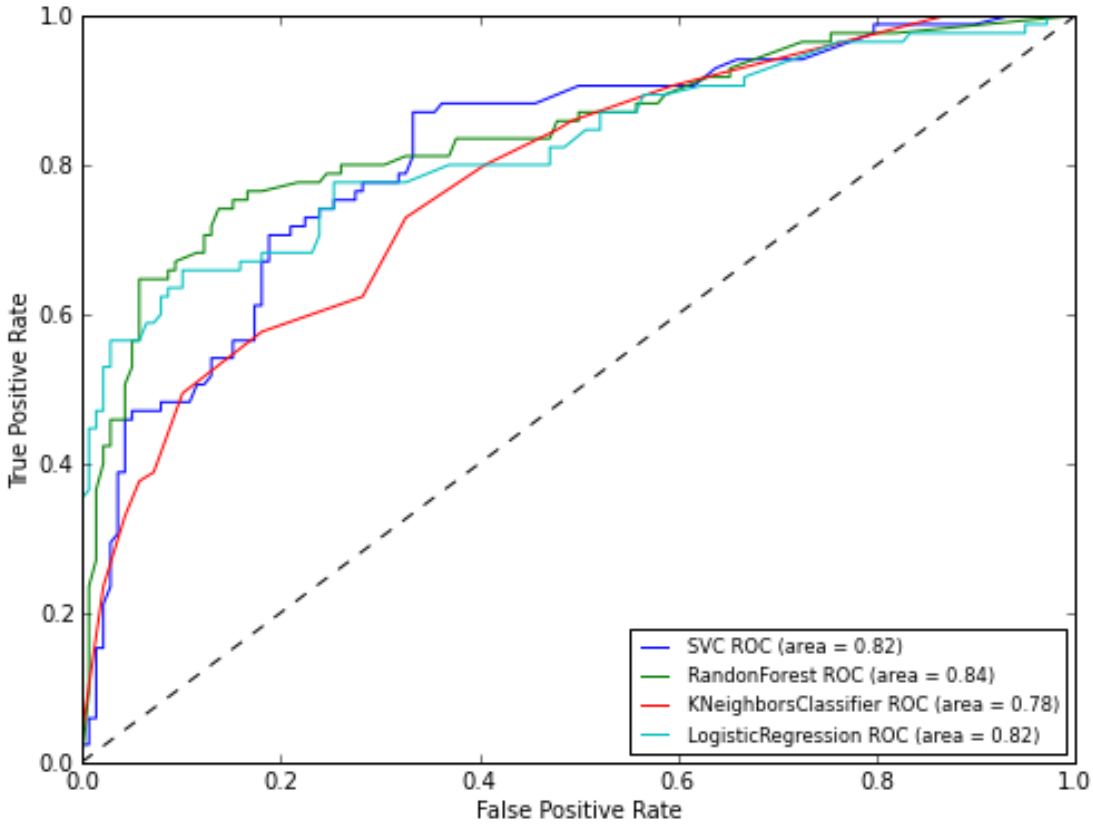
- Целевые значения на тестовой выборке нам не известны. Как локально проверять качество алгоритма?
- Можно **обучающую** выборку разделить на **2 части:**
  - **Train** – например, первые **80% объектов** от обучающей выборки
  - **Validation** – проверочная - оставшиеся **20% объектов**
- После чего «**обучить**» алгоритм на выборке **Train**, «**проверить**» - на выборке **Validation**
- Сделаем так много раз и посчитаем среднее значение **Precision – данный подход носит название 5-fold**
- Также можем построить **ROC-кривые**

Тестирование алгоритмов машинного обучения, выбор модели



- Выбираем среди KNN, SVM, RandomForest и логистической регрессии
- Для каждого алгоритма также необходимо подбирать параметры, например, в KNN мы использовали **18** соседей, а в RandomForest **70** деревьев
- Видим, что наилучшее значение метрики качества показал алгоритм **RandomForest**

Тестирование алгоритмов машинного обучения, выбор модели



- Строим **ROC – кривые** для KNN, SVM, RandomForest и логистической регрессии
- Чем выше и выпуклее ROC-кривая, тем качественнее алгоритм
- Снова наилучшее значение метрики качества показал алгоритм **RandomForest**

Анализ результатов, презентация внутри команды

- Итого, в итоговом варианте мы получили алгоритм, **который на обучающей выборке показывает значение метрики 5-fold равное 0.77**
- На входе алгоритм принимает значения признаков:
  - **PClass** (социально - экономический статус), **Sex** (пол), **Age** (возраст), **SubSp** (кол-во родственников 1го порядка), **Parch** (кол-во родственников 2го порядка), **Fare** (цена билета), **Embarked** (порт посадки)
- На выходе алгоритм прогнозирует бинарное значение (**утонет/не утонет**)

Осталось правильно это «продать» бизнесу, оперируя понятиями: качество, точность, охват, лифт, value и другими, понятными конкретному заказчику понятными метриками

Как продать?

- Можно пойти **4мя** путями:
  - Мы получили алгоритм, **который на обучающей выборке показывает значение метрики 5-fold равное 0.77**
  - Мы сделали алгоритм **с точностью 0.77**, который для каждого пассажира предсказывает, утонет он или нет
  - Мы взяли обучающую выборку, разбили ее на **2 части**, на одной обучились, на второй протестировали. Так сделали несколько раз. Усреднили результаты. Получили тем самым значение метрики **5-fold** равным **0.77**. Готовы брать признаки и предсказывать факт спасения человека
  - **Дайте нам несколько пассажиров с последнего Титаника, а мы расскажем, кто из них утонул**

Не поспоришь!

Работает!!!

Кластеризация городов России по социально-экономическим признакам

Оригинал:
habrahabr.ru/post/140837/

Кластеризация городов России по социально-экономическим признакам

- Имеется **17** крупнейших (по численности населения) **городов** России
- Каждый город описывается **10ю** социально-экономическими **признаками**:
  - Население, плотность населения, прирост населения
  - Средняя зарплата, средняя численность трудоустроенных
  - Общая площадь жилых помещений
  - Кол-во преступлений на одного жителя, число обучающихся на одного жителя
  - …
- Необходимо найти «**похожие**» между собой города, например, чтобы прогнозировать бюджет или сравнивать города между собой

Разделение на 3 кластера

Санкт-Петербург, Екатеринбург, Новосибирск Нижний Новгород, Уфа, Пермь, Красноярск, Краснодар	В основном схожи по численности населения
Москва	Отдельный кластер
Самара, Казань, Омск, Челябинск, Волгоград, Ростов-на-Дону, Воронеж, Саратов	Опять же, разделение в основном по демографии

Разделение на 5 кластеров

Самара, Ростов-на-Дону, Челябинск, Нижний Новгород

Омск, Волгоград, Саратов, Казань, Воронеж

Новосибирск, Екатеринбург, Пермь, Красноярск, Краснодар, Уфа

Санкт-Петербург

Москва

- Появляется новый нетипичный объект – Санкт-Петербург
- Оба города похожи между собой по демографическим признакам, но разнятся, например, в средней заработной плате
- Далее необходимо все больше изучать разницу между городами и «всматриваться» в отдельный кластер, чтобы попять результат кластеризации

Дальнейшее объяснение кластеризации индивидуально

Классификация отзывов к фильмам

Классификация отзывов в фильмам

- Имеются наборы отзывов к фильмам:
  - **25 тыс** размеченных отзывов к фильмам (**обучающая выборка**)
  - **25 тыс** неразмеченных отзывов к фильмам (**тестовая выборка**)
  - **50 тыс** дополнительных неразмеченных отзывов (**зачем?**)
- Необходимо научить классифицировать отзывы к фильмам на 5 классов, например, чтобы прогнозировать рейтинг фильма

Задача классификации на 5 классов

Сложные объекты - тексты

Посмотрим на данные

ID	Review	Sentiment
5814_8	1	With all this stuff going down at the moment ...
2381_9	0	The Classic War of the Worlds by Timothy ...
7759_3	4	The film starts with a manager (Nicholas Bell...
3630_4	?	It must be assumed that those who praised thi...
9495_8	?	Superbly trashy and wondrously unpretentious ...

Осталось научиться извлекать признаки

Извлечение признаков из текстовых данных

Bag of Words

- Документ – это набор слов
- Признаки интерпретируются
- Не учитываются связи между словами
- Признаки для документа – наличие очередного слова
- Требуется предобработка данных

Word2Vec

- Слово описывается набором числовых параметров
- Учитываются связи между словами
- Признаки для слова – не интерпретируются
- Не требуется предобработка данных

Извлечение признаков: Bag of Words

- **Очищаем** каждый текст:
  - Удаляем мусор (html-теги, заголовки, служебные отметки)
  - Выбрасываем числа и знаки препинания
  - Разделяем на слова
  - Приводим к нижнему регистру
  - Делаем **лемматизацию**
  - Делаем **стемминг**
  - Удаляем стоп-слова (в каждом языке - свои)
  - Удаляем наиболее редко встречающиеся слова
- **Отзыв** – это набор слов (**bag of words**)

Строим Document Term Matrix

- Строим **словарь**
  - Берем все слова, используемые во всех отзывах, отсекаем **N** самых часто встречающихся
- Для каждого отзыва, набор признаков – это факт наличия в нем очередного слова **из словаря**
- Итого, каждый **отзыв** имеет **N** бинарных признаков

Словарь может строиться на всех объектах

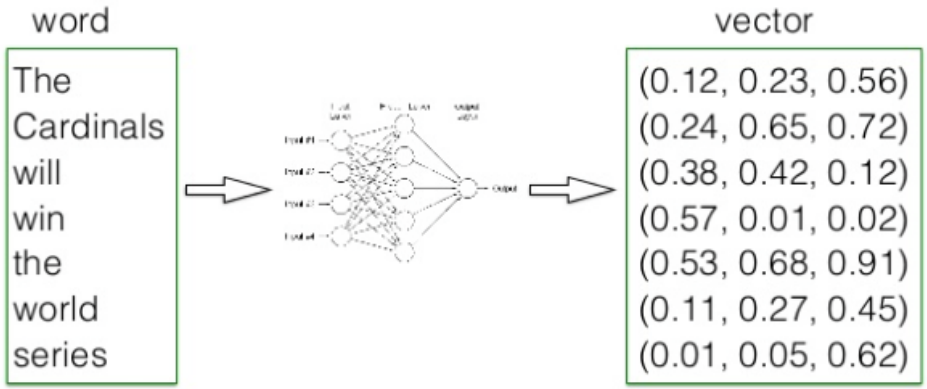
Можно для каждого слова из словаря считать кол-во его вхождений в очередной отзыв

Document Term Matrix

ID	Review	“film”	“actor”	“love”
5814_8	1	1	0	1
2381_9	0	0	1	1
7759_3	4	1	1	0
3630_4	**?**	1	0	0
9495_8	**?**	0	1	1

Далее, также как и в предыдущей задаче, используя алгоритмы машинного обучения строим модель

Извлечение признаков: Word2Vec



После преобразования слова можно складывать, вычитать и т. д.

- **Word2Vec** – структура данных, первоначально разработанная и используемая компанией Google для анализа поисковых запросов
- На **вход** подается большое количество необработанных текстов, а также другие дополнительные параметры
- На **выходе** для каждого слова имеем вектора одинаковой длины, состоящие из не интерпретируемых числовых признаков

Word2Vec: примеры

- Подадим на вход **word2vec** большое количество поисковых запросов на русском языке

- Посмотрим на возможности **word2vec** на примерах:
  - Опечатки
  - Транслитерация
  - Слова-синонимы
  - Семантически близкие слова
  - Кластеризация запроса
  - Семантическая разница между словами
  - Оценка важности слов в поисковом запросе

Оригинал:
habrahabr.ru/post/249215/

Word2Vec: опечатки

Word: преключение
Word Cosine distance
— приключение 0.748698
преключения 0.726111
риключения 0.692828
приключеия 0.670168
прключение 0.666706
приключеня 0.663286
приключени 0.659609

- Вычисляя угол между векторами (скалярное произведение) – мы можем узнать «расстояние» между словами

Таким образом, наиболее близкое по значению слово есть правильное, если имеется опечатка

Word2Vec: транслитерация

Word: avito	
Word	**Cosine distance**
— awito	**0.693721**
авито	0.675299
fvito	0.661414
авита	0.659454
irr	0.642429
овито	0.606189
avito	0.598056

- Вычисляя угол между векторами (скалярное произведение) – мы можем узнать «расстояние» между словами

Можем находить не только опечатки, но и сайты-аналоги

Word2Vec: слова-синонимы

Word: mail	
Word	**Cosine distance**
— rambler	**0.777771**
meil	0.765292
inbox	0.745602
maill	0.741604
yandex	0.696301
maii	0.675455
myrambler	0.674704

- Вычисляя угол между векторами (скалярное произведение) – мы можем узнать «расстояние» между словами

Можем определять также слова-синонимы

Word2Vec: слова-синонимы

Word: кофе	
Word	**Cosine distance**
— коффе	**0.734483**
чая	0.690234
чай	0.688656
капучино	0.666638
кофн	0.636362
какао	0.619801
эспрессо	0.599390

- Вычисляя угол между векторами (скалярное произведение) – мы можем узнать «расстояние» между словами

В зависимости от настроек word2vec мы можем выдавать слова, синонимичные запросу

Word2Vec: описание запроса

Word: кофе
Word Cosine distance
— зернах 0.757635
растворимый 0.709936
чая 0.709579
коффе 0.704036
сублемированный 0.694553
молотый 0.690066
кофейные 0.680409

- Вычисляя угол между векторами (скалярное произведение) – мы можем узнать «расстояние» между словами

В зависимости от настроек word2vec мы можем выдавать слова, характеризующие запрос

Word2Vec: кластеризация запроса

Word: мобильный телефон
Word Cosine distance
— сотовый 0.811114
телефона 0.776416
смартфон 0.730191
телфон 0.719766
мобильного 0.717972
мобильник 0.706131
телефо 0.698894

- «Расстояние» можно считать не только между словом, но и между предложениями и документами, предварительно вычисляя вектор признаков для предложения/документа

Сложный запрос может быть сведен к более простому

Word2Vec: семантическая разница между словами

Word: франция париж германия	
Word	**Cosine distance**
— мюнхен	**0.716158**
берлин	0.671514
дюссельдорф	0.665014
гамбург	0.661027
кельн	0.646897
амстердам	0.641764
франкфурт	0.638686

- «Расстояние» можно считать не только между словом, но и между предложениями и документами, предварительно вычисляя вектор признаков для предложения/ документа

Нашли слово, которое к Германии относится также как Париж к Франции

Word2Vec: оценка важности слов в поисковом запросе

Word: купить пиццу в москве
Importance
купить = 0.159387
пиццу = 1
в = 0.403579
москве = 0.455351

- На наших векторах можно также делать кластеризацию
- После, можно определить, к какому кластеру тяготеет запрос в целом
- А дальше упорядочить слова по расстоянию до центра кластера

Word: владимир путин
Importance
владимир = 0.28982
путин = 1

Можно определить объект, который описывается в предложении

Возвращаемся к задаче

- Подадим на вход **word2vec** большое количество неразмеченных предложений
- Каждый **отзыв** описывается **словами**
- Каждое **слово** – вектором **признаков**
- Посчитаем для каждого слова **среднее значение векторов слов, входящих в него** – это и будет результирующий **вектор признаков**

Далее, также как и в предыдущей задаче, используя алгоритмы машинного обучения строим модель

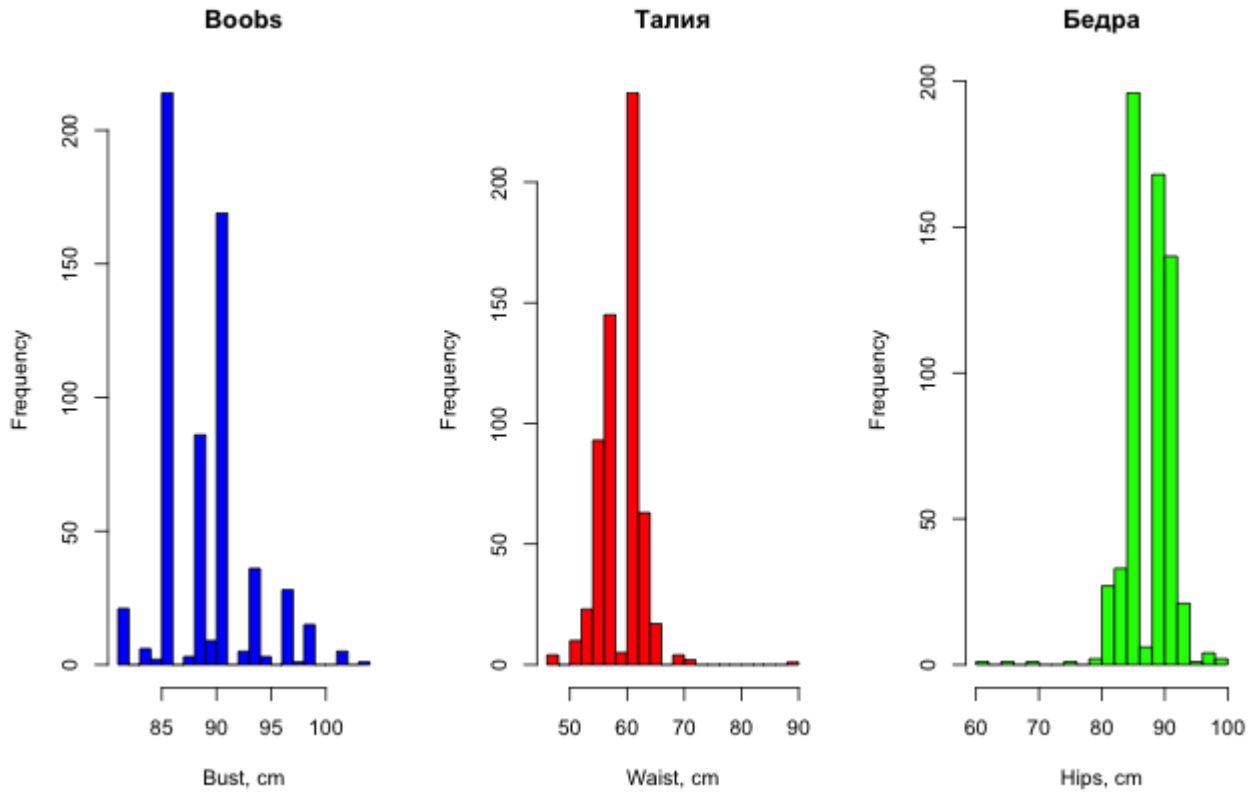
Поиск необычных моделей Playboy

Оригинал:
habrahabr.ru/post/251225/

Поиск необычных моделей Playboy

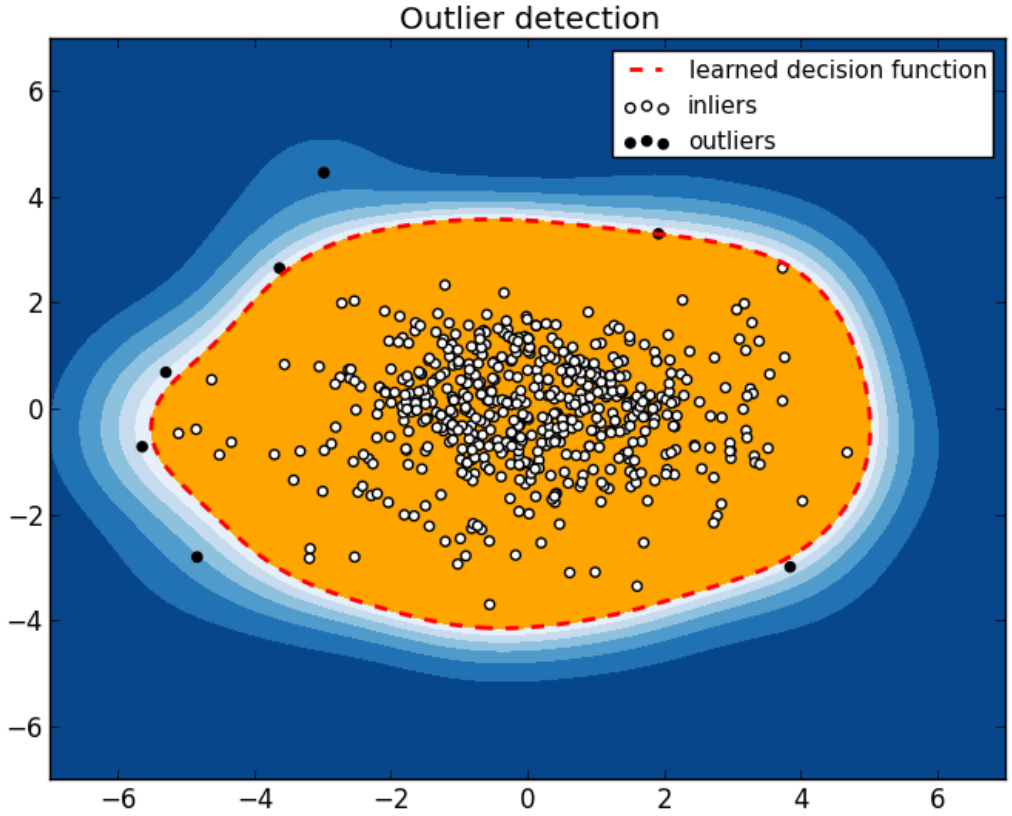
- Имеется 604 «девушек месяца» Playboy с декабря 1953 по январь 2009
- Для каждой из них известны:
  - Обхват груди (Bust, в см)
  - Обхват талии (Waist, в см)
  - Обхват бедер (Hips, в см)
  - Рост (Height, в см)
  - Вес (Weight, в кг)
- Хотелось бы найти самых «необычных» моделей за весь период

Первичный анализ данных



- По данным видно, что средние формы модели Playboy – 89-60-89
- Средний рост – 168 см
- Вес – 52 кг
- Есть одна модель с талией 89 см – позже выяснится, что это выброс
- Есть одна модель с бедрами 61 см – это тоже выброс

One-Class SVM



- Существует множество алгоритмов для выделения нетипичных (**outliers**) объектов в выборках. Один из них – модификация **SVM**
- Задается процент выбросов
- Строится разделяющая поверхность, отделяющая выбросы от обычных объектов

Задав **1%** выбросов, получаем **7** моделей

One-Class SVM

Мики Уинтерс. Сентябрь, 1962. 91-46-86, рост 152, вес 45

Талия 46 — это, конечно, круто! Как у них при этом грудь 91?

One-Class SVM

Кристин Уильямс. Октябрь, 1963. 94-66-94, рост 183, вес 68

Достаточно большая девушка, учитывая средние значения тех лет

One-Class SVM

Розмари Хилкрест. Октябрь, 1964. 104-64-97, рост 168, вес 66

Еще более внушительная женщина

One-Class SVM

Сюзан Миллер. Сентябрь, 1972. 98-64-99, рост 185, вес 64

Необычное сочетание рост/вес

One-Class SVM

Тройняшки Дам. 86-89 (реально 64)-86, рост 173, вес 52

Это как раз те самые выбросы, которые обнаружились вначале

One-Class SVM

Кара Мишель. Декабрь, 2000. 86-66-91, рост 188, вес 61

Рост 188 см - многовато для того времени

One-Class SVM

Кармелла де Сезаре.
Апрель, 2003. 86-61-69,
рост 173, вес 54

«Пожалуй, из-за бедер»

Ссылки на использованные ресурсы

- Часть материала основана на статьях:
  - http://habrahabr.ru/users/akrot/
  - http://habrahabr.ru/post/202090/
  - http://habrahabr.ru/post/140837/
  - http://habrahabr.ru/post/249215/
  - http://habrahabr.ru/post/251225/
- А также форумах:
  - https://www.kaggle.com
- И ресурсов по машинному обучению:
  - http://www.machinelearning.ru, http://mlclass.ru

Ресурсы для дальнейшего изучения

- Пишите по любым вопросам/проектам:
  - al.krot.kav@gmail.com
  - MLClass.ru
- Статьи на хабре
  - http://habrahabr.ru/users/akrot/
- Курс Andrew Ng на Coursera.org: Machine Learning
- Решение практических задач: https://www.kaggle.com

Спасибо за внимание