КУРСЫ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ
(ФАКУЛЬТАТИВНЫЙ КУРС)
“МАШИНОЕ ОБУЧЕНИЕ.
ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ
И ГЕНЕТИЧЕСКИЕ АЛГОРИТМЫ”

Лекция № 2
Основы машинного обучения
Базовые алгоритмы машинного обучения

НИИЯФ МГУ

Лаборатория адаптивных методов обработки данных

Сайт курса: http://kpk-nnga.sinp.msu.ru

Авторы и преподаватели курса:
С.А.Доленко, А.О.Ефиторов, В.Р.Широкий
И.В.Исаев, И.М.Гаджиев, Р.Д.Владимиров
Ю.В.Орлов, И.Г.Персианцев, О.А.Агапкин, А.Г.Гужва

Использованы материалы, разработанные авторами при поддержке

Фонд интеллект

Лекцию читает зав. лаб. к.ф.-м.н. Сергей Анатольевич Доленко

СТАНДАРТНАЯ СТРУКТУРА ДАННЫХ ДЛЯ МО

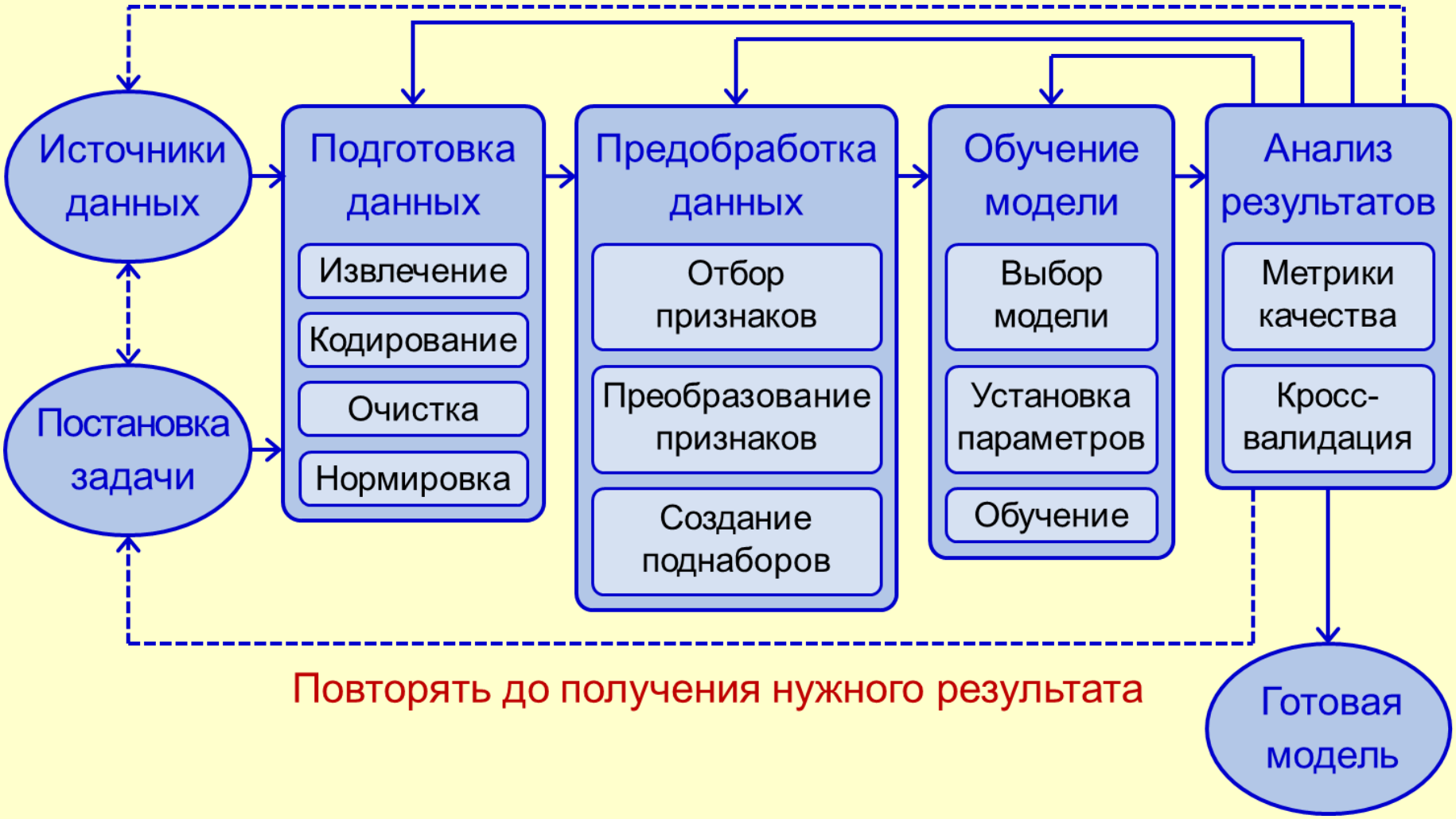
- **Прямоугольная матрица**

		Входные признаки					Выходные признаки			
	ID_прим	ВхПр1	ВхПр2	ВхПр3	…	ВхПрN	ВыхПр1	ВыхПр2	…	ВыхПрL
Примеры	Прим1	X11	X12	X13	…	X1N	Y11	Y12	…	Y1L
	Прим2	X21	X22	X23	…	X2N	Y21	Y22	…	Y2L
	Прим3	X31	X32	X33	…	X3N	Y31	Y32	…	Y3L
	Прим4	X41	X42	X43	…	X4N	Y41	Y42	…	Y4L
	…	…	…	…	…	…	…	…	…	…
	ПримP	XP1	XP2	XP3	…	XPN	YP1	YP2	…	YPL

- **Столбцы – признаки (features) (N входных и L выходных)**
- **Первая строка может содержать названия признаков**
- **Строки – примеры (patterns) (P примеров)**

  Термины «образцы», “examples”, “samples” имеют побочные смыслы.
- **Первый столбец может содержать идентификаторы примеров**
- **Матрица не должна содержать пропусков!**

ОБЩАЯ СХЕМА МАШИННОГО ОБУЧЕНИЯ



Могут использоваться не все элементы схемы.

Это зависит от специфики задачи и требований к решению.

МЕТОДЫ ОЦЕНКИ КАЧЕСТВА РЕШЕНИЯ ДЛЯ ЗАДАЧ РЕГРЕССИИ

Постановка задачи регрессии: минимизация отклонений между реальными значениями и значениями, предсказанными моделью.

Оценка качества для задачи регрессии = анализ отклонений

Возможные негативные особенности отклонений:

- **Выбросы - высокие значения отклонений для некоторых примеров**
- **Смещение (bias) и высокий разброс (variance)**
- **Гетероскедастичность - неоднородность отклонений, зависимость величины отклонений от какого-либо фактора**

МЕТОДЫ ОЦЕНКИ КАЧЕСТВА РЕШЕНИЯ ДЛЯ ЗАДАЧ РЕГРЕССИИ

- **Визуальный**
  - **Диаграмма рассеяния**
  - **Гистограмма ошибок**
- **Метрики качества**
  - **Среднее абсолютное отклонение (САО) (MAE - Mean Absolute Error)**
  - **Среднеквадратичное отклонение (СКО) (RMSE – Root Mean Squared Error)**
  - **Коэффициент детерминации R^2**
  - **Коэффициент корреляции r**
  - **Взвешенная ошибка – если требуется разная цена ошибки в различных диапазонах**

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|a(x_i) - y_i|$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(a(x_i) - y_i)^2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(a(x_i) - y_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

$$r = \frac{\sum_{i=1}^{N}(a(x_i) - \bar{a})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(a(x_i) - \bar{a})^2} \cdot \sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

$$weighted\ MAE = \frac{1}{N}\sum_{i=1}^{N} w_i|a(x_i) - y_i|$$

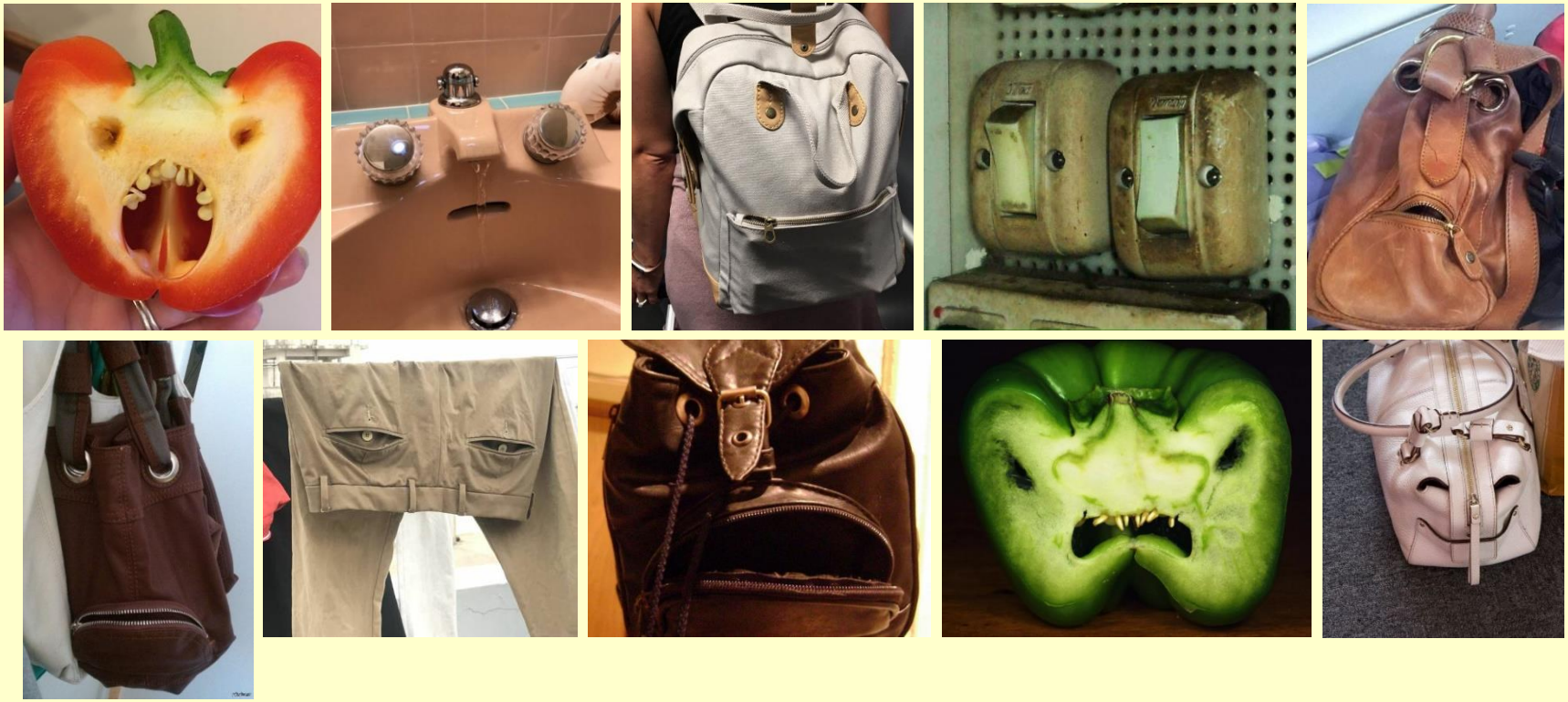
МЕТОДЫ ОЦЕНКИ КАЧЕСТВА РЕШЕНИЯ ДЛЯ ЗАДАЧ КЛАССИФИКАЦИИ

Особенности ошибок бинарной классификации:

- **Наличие двух типов ошибки:**
  - **Ошибка первого рода:**
    ложноположительная, “ложная тревога”, false positive
  - **Ошибка второго рода:**
    ложноотрицательная, “пропуск цели”, false negative
- **Ошибки взаимно симметричны**
  - **При оценке ошибок можно поменять местами положительный и отрицательный исход**
  - **За положительный исход обычно берется естественное положение вещей**
- **Ошибки связаны между собой:**
  - **Уменьшение ошибки одного типа может приводить к увеличению ошибки другого типа**
- **Для ряда задач цена ошибок 1-го и 2-го рода может быть разной**
  - **Диагностирование болезней – минимизация пропусков цели, за счет некоторого увеличения ложных тревог**

МИНИМИЗАЦИЯ ОШИБОК ВТОРОГО РОДА В ЖИВОЙ ПРИРОДЕ

Парейдолия - зрительная иллюзия, заключающаяся в формировании иллюзорных образов на основе реальных объектов



Тот, кто убежал от пустого куста, передал свои гены потомкам, а тот, кто не убежал от куста, где спрятался хищник – нет.

МЕТОДЫ ОЦЕНКИ КАЧЕСТВА БИНАРНОЙ КЛАССИФИКАЦИИ

- **Таблица сопряженности**
  - Количество примеров для каждого исхода

		Ответ модели	
		Positive	Negative
Истинное значение	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

- **Общая точность (Accuracy)**
  - Доля правильных ответов
  - Плохо работает при дисбалансе

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

- **Точность, полнота, специфичность (Precision, Recall, Specificity)**
  - Оценка качества на каждом классе по отдельности

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

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

$$Specificity = \frac{TN}{TN + FP}$$

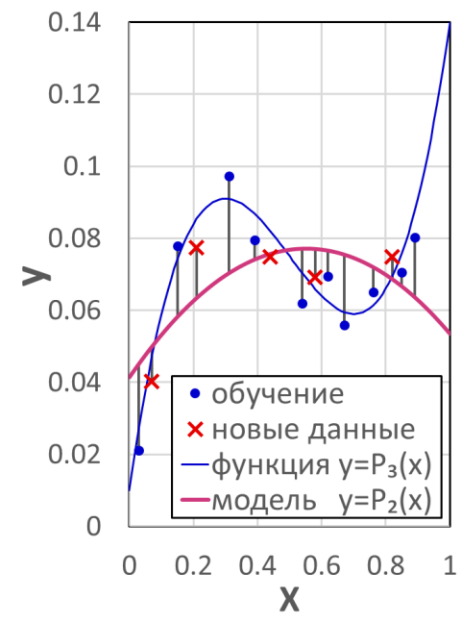
- **F1-мера (F1-score)**
  - Среднее гармоническое между точностью и полнотой

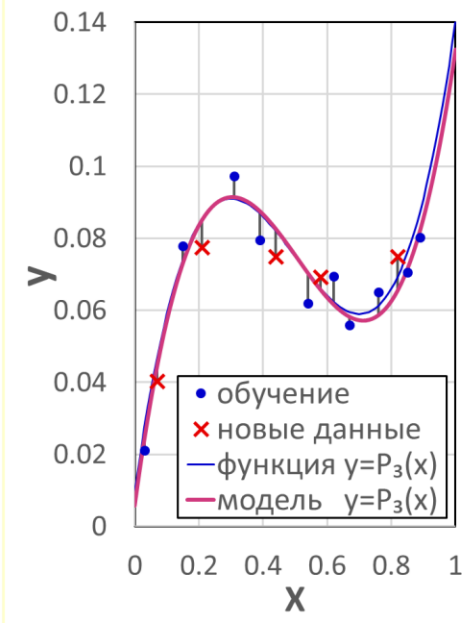
$$F_1\ score = 2\frac{Precision \cdot Recall}{Precision + Recall}$$

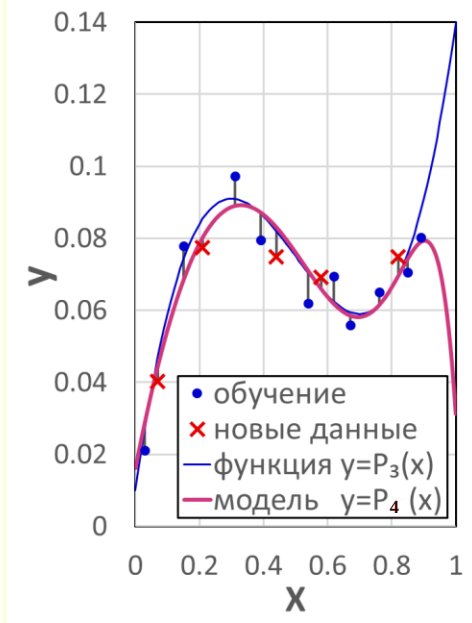
- **ROC – параметрическая кривая TP(FP), AUC – площадь под ней**

НЕДООБУЧЕНИЕ И ПЕРЕОБУЧЕНИЕ

Качество работы моделей на новых данных:







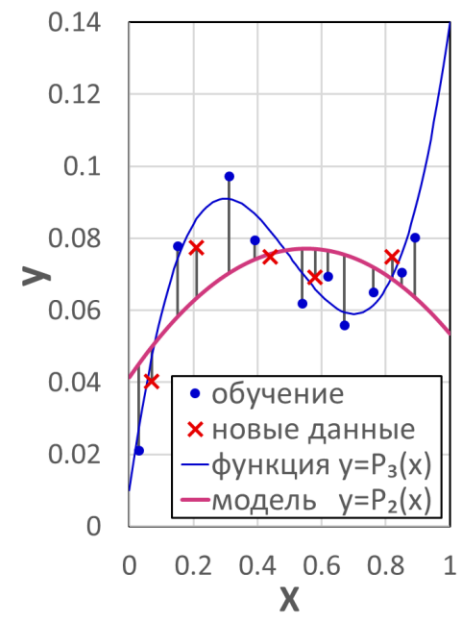
Недообучение **Хорошая модель** **Переобучение**

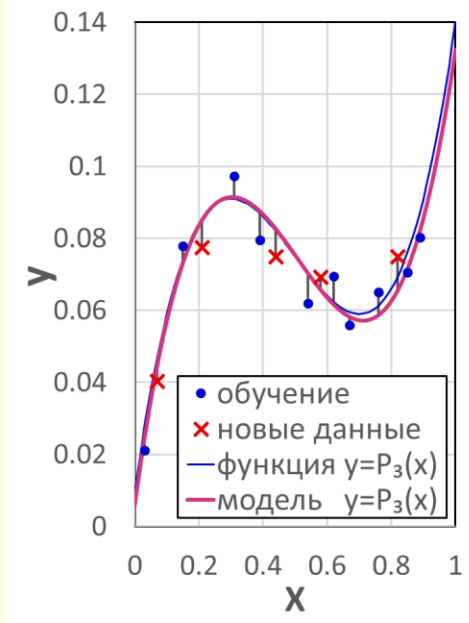
Недообучение – большая ошибка и на данных обучения, и на новых данных

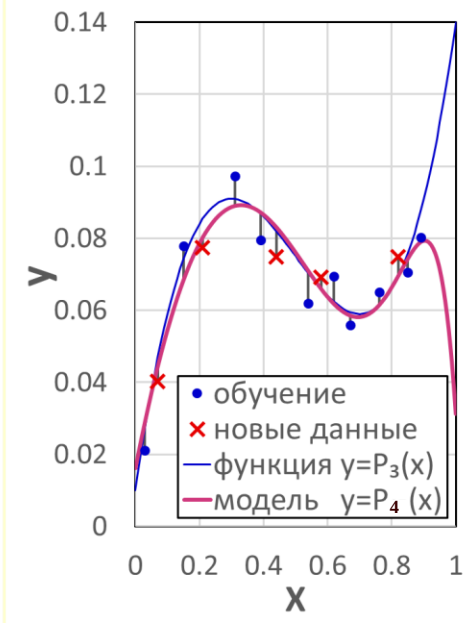
Переобучение – малая ошибка на данных обучения, большая ошибка на новых данных

НЕДООБУЧЕНИЕ И ПЕРЕОБУЧЕНИЕ

Качество работы моделей на новых данных:







Недообучение **Хорошая модель** **Переобучение**

Для выявления случаев переобучения моделей

оценка качества их работы

должна производиться на независимых данных,

которые в обучении не участвовали.

ЛИНЕЙНАЯ РЕГРЕССИЯ

Один из самых простых алгоритмов машинного обучения.

Моделируем зависимость целевой переменной

от *p* признаков в виде:

$$\hat{y} = \sum_{j=1}^{p} w_i x_i + w_0$$

Пусть *X* – тренировочный набор (в строках находятся примеры x_i).

$$X = \begin{pmatrix} 1 & X_0 \\ \dots & \dots \\ 1 & X_N \end{pmatrix}$$

Формулу можно записать в матричном виде:

$$\hat{Y} = Xw$$

ЛИНЕЙНАЯ РЕГРЕССИЯ: ПОЛУЧЕНИЕ ВЕСОВ

Обозначим за *Y* ответы примеров тренировочного набора

Для получения весов можно воспользоваться

методом наименьших квадратов:

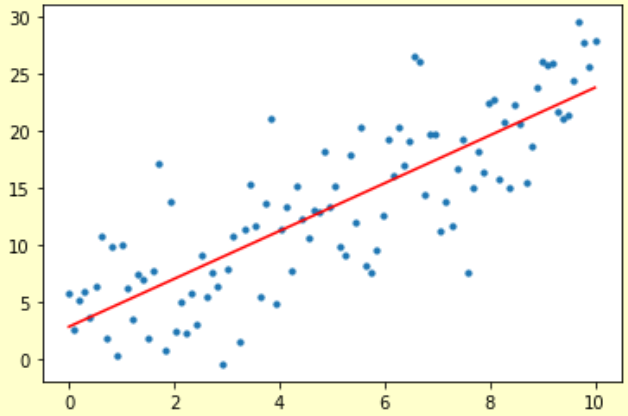
$$L(w) = (Y - Xw)^r (Y - Xw) \rightarrow min$$

Условие минимума *L(w)* – равенство нулю производной,

откуда можно получить выражение для весов

в задаче линейной регрессии:

$$w = (X^T X)^{-1} X^T Y$$



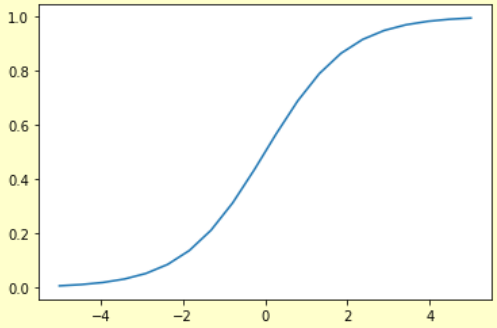
ЛИНЕЙНАЯ РЕГРЕССИЯ: ЗАМЕЧАНИЯ

- Метод наименьших квадратов дает оптимальную оценку параметров линейной регрессии (ЛР) согласно критерию максимального правдоподобия
- Перед использованием линейной регрессии лучше провести масштабирование данных, т.к. признаки могут иметь разную размерность
- Для подбора параметров можно использовать и САО (MAE) – для этого нужно применить численные методы
- ЛР легко превращается в ЛР в нелинейном базисе (полиномиальную, гармоническую и др.) – достаточно добавить столбцы с соответствующими функциями признаков

ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ

Линейная регрессия плохо подходит для решения задачи классификации, т.к. получаемые значения не ограничены. Для получения значений в диапазоне от [0,1] и моделирования вероятности принадлежности к классу 0 или 1 используется логистическая регрессия:

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}} \qquad z = \sum_{j=1}^{p} w_i x_i + w_0$$



Для выбора оптимальных весов w обычно максимизируют (например, методом Ньютона) логарифм правдоподобия

$$L = \sum_{i=1}^{N} y_i \log(\hat{y}_i) \rightarrow max$$

ЛОГИСТИЧЕСКАЯ РЕГРЕССИЯ: ЗАМЕЧАНИЯ

- Перед использованием логистической регрессии лучше провести масштабирование данных, т.к. признаки могут иметь разную размерность;
- Линейная логистическая регрессия легко превращается в логистическую регрессию в нелинейном базисе – достаточно добавить столбцы с соответствующими функциями признаков
- Многомерный вариант логистической регрессии – использование функции *softmax* (часто используется в качестве передаточной функции выходного слоя нейронной сети, решающей задачу многоклассовой классификации)

$$softmax(x) = \frac{e^{x_i}}{\sum e^{x_i}}$$

РЕГУЛЯРИЗАЦИЯ Lp

Проблема переобучения – возникает, когда

модель имеет слишком большое число параметров

Решение: использование регуляризации – подавления весов.

Заставляем модель выбирать веса

максимально «экономным» способом,

штрафуя за большие величины весов.

Для этого модифицируем функцию потерь *L*

Общий вид *Lp* регуляризации:

$$L_p = L(w) + \lambda \sum_{j=1}^{s} |w_j|^p$$

РЕГУЛЯРИЗАЦИЯ L2 (ГРЕБНЕВАЯ)

Пусть *P=2*, т.е. в качестве штрафной добавки к функции потерь используем сумму квадратов весов:

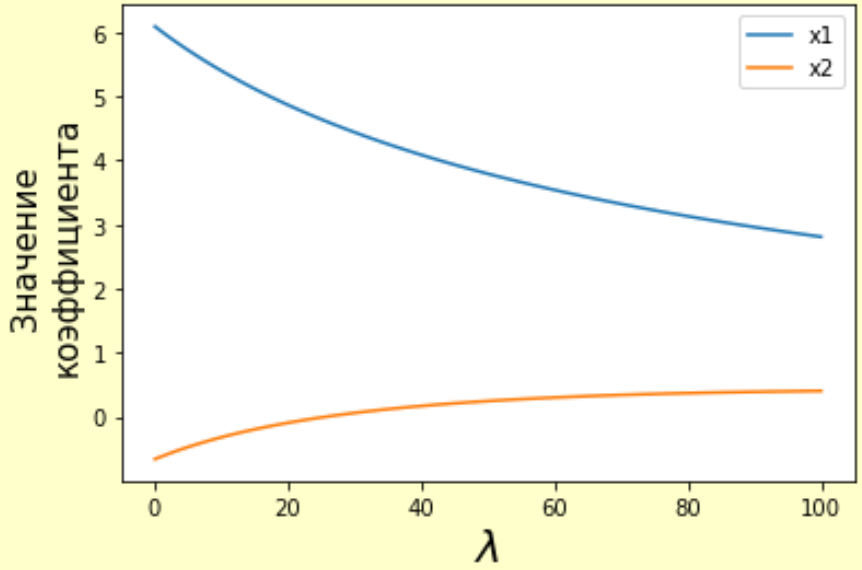
$$L_2 = L(w) + \lambda \sum_{j=0}^{s} |w_j|^2$$

В случае, если в качестве исходной модели используем линейную регрессию (в таком случае модель обычно называют гребневой (ridge) регрессией), а в качестве функционала ошибки – сумму квадратов остатков, выражение для весов принимает вид:

$$w = (X^T X + \lambda Y)^{-1} X^T Y$$

РЕГУЛЯРИЗАЦИЯ L2: ПОВЕДЕНИЕ КОЭФФИЦИЕНТОВ

- Рассмотрим простую задачу линейной регрессии с двумя скоррелированными признаками
- Смотрим на поведение значений коэффициентов при увеличении параметра регуляризации λ
- Характерное поведение – коэффициенты либо плавно убывают, либо плавно возрастают
- Чем больше параметр – тем больший вес имеет их сумма квадратов



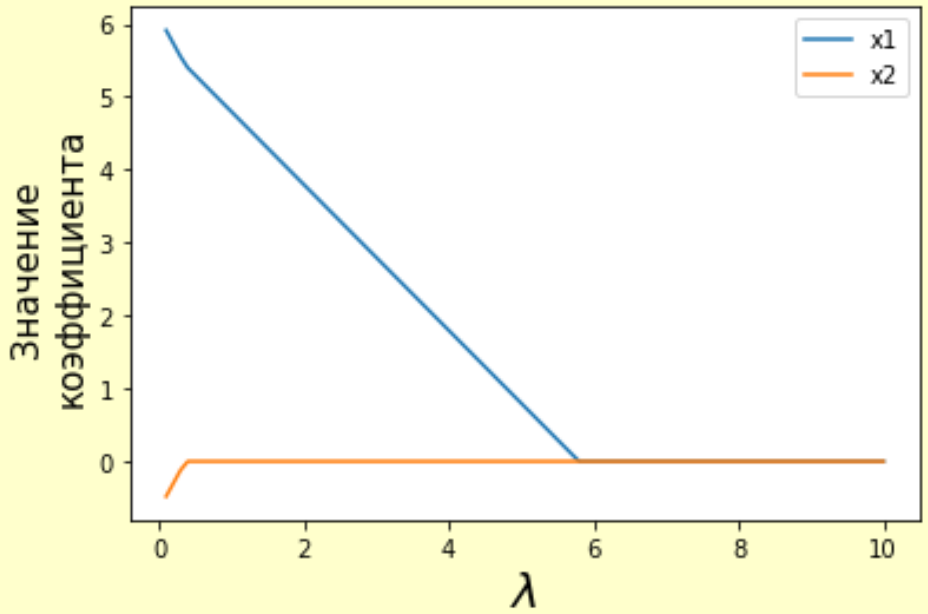
РЕГУЛЯРИЗАЦИЯ L1 (ЛАССО)

Пусть *P=1*, т.е. в качестве штрафной добавки к функции потерь используем сумму модулей весов:

$$L_1 = L(w) + \lambda \sum_{j=0}^{s} |w_j|$$

Такую модель обычно называют лассо (lasso) регрессией

Поведение коэффициентов из предыдущего примера:



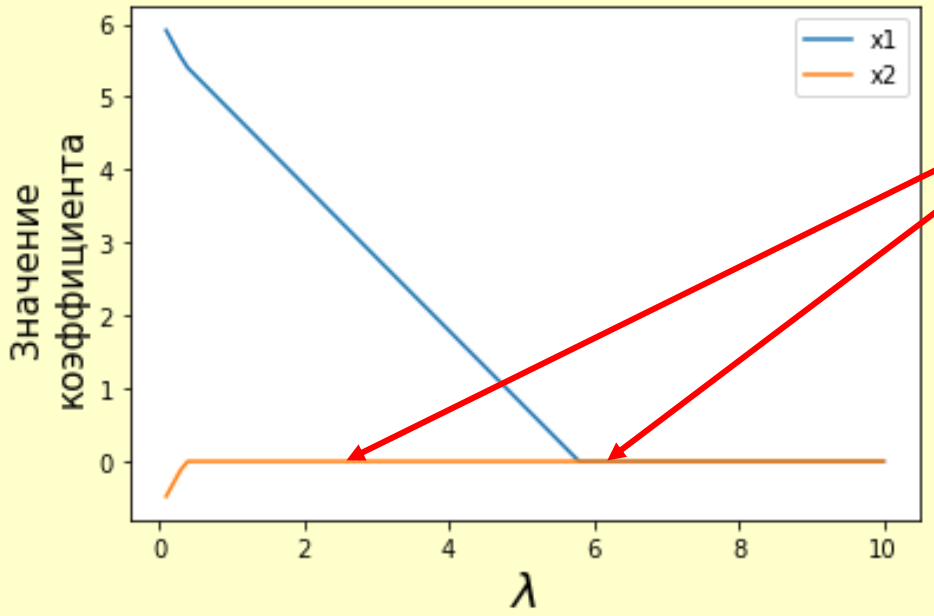
РЕГУЛЯРИЗАЦИЯ L1 (ЛАССО)

Пусть *P=1*, т.е. в качестве штрафной добавки к функции потерь используем сумму модулей весов:

$$L_1 = L(w) + \lambda \sum_{j=0}^{s} |w_j|$$

Такую модель обычно называют лассо (lasso) регрессией

Поведение коэффициентов из предыдущего примера:



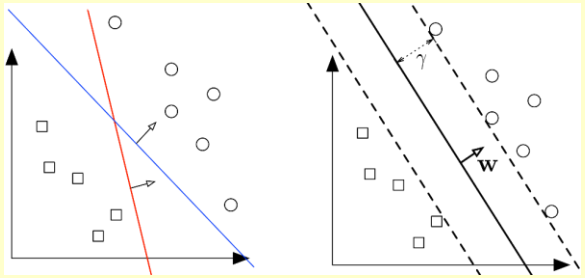
Начиная с некоторого значения параметра, коэффициенты обращаются в 0

РЕГУЛЯРИЗАЦИЯ: ЗАМЕЧАНИЯ

- L2 регуляризация позволяет уменьшить переобучение модели
- L2 регуляризация позволяет уменьшить веса «плохих» признаков или даже полностью обнулить их.
  Таким образом осуществляется «мягкий» отбор признаков
- Иногда используют комбинацию L1 и L2 регуляризации – эластичную сеть (Elastic Net).
- При этом используется сумма линейной и квадратичной штрафных добавок к функции потерь
- Эластичная сеть позволяет сочетать плавное уменьшение и полное обнуление «плохих» признаков
- Параметры регуляризации следует подбирать индивидуально для каждой задачи!

МЕТОД (МАШИНА) ОПОРНЫХ ВЕКТОРОВ

- Support Vector Machine (SVM)
- Решаем задачу бинарной классификации
- Главная идея – провести такую разделяющую гиперповерхность в пространстве признаков, которая максимально удалена от соседних примеров из противоположных классов (имеет максимальный отступ)
- Формализация: поиск гиперплоскости, расстояние от которой до ближайшего примера должно быть максимально
- При этом она должна правильно отделять классы друг от друга
- Выражение для отступа, который должен быть максимизирован:



$$M = \min_{i=1..n} \frac{y_i(wx+w_0)}{|w|}$$

https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote09.html

МЕТОД ОПОРНЫХ ВЕКТОРОВ: ФОКУС С ЯДРОМ

- Чаще всего классы **линейно неразделимы**
- Тогда можно **нелинейно отобразить**

  исходные вектора примеров в некоторое пространство,

  в котором будет строиться линейная граница
- Выберем в качестве такого отображения h(x)

  Максимизируем отступ:

$$\max_{\alpha}\left(\frac{1}{2}\sum_{i=1}^{n}\alpha_i-\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j h(x_j)^T h(x_j)\right)$$

Тогда оптимальные веса можно записать в виде: $w=\sum_{i=1}^{n}\alpha_i{}^* h(x_i)y_i$

Общее решение: $f(x)=h(x)^T w+w_0=\sum_{i=1}^{n}\alpha_i{}^* h(x)^T h(x_i)y_i+w_0$

МЕТОД ОПОРНЫХ ВЕКТОРОВ: ФОКУС С ЯДРОМ

- Чаще всего классы линейно неразделимы

Kernel Trick

- Тогда можно нелинейно отобразить исходные вектора примеров в некоторое пространство, в котором будет строиться **линейная** граница
- Выберем в качестве такого отображения h(x)

Максимизируем отступ:

$$\max_{\alpha}\left(\frac{1}{2}\sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j h(x_j)^T h(x_j)\right)$$

Достаточно определить $K(x,x') = h(x)^T h(x')$

Тогда оптимальные веса можно записать в виде: $w = \sum_{i=1}^{n}\alpha_i^* h(x_i) y_i$

Общее решение: $f(x) = h(x)^T w + w_0 = \sum_{i=1}^{n}\alpha_i^* h(x)^T h(x_i) y_i + w_0$

МЕТОД ОПОРНЫХ ВЕКТОРОВ: ФОКУС С ЯДРОМ

- Чаще всего классы линейно неразделимы

Kernel Trick

- Тогда можно нелинейно отобразить исходные вектора примеров в некоторое пространство, в котором будет строиться **линейная** граница
- Выберем в качестве такого отображения h(x).

Максимизируем отступ:

$$\max_{\alpha}\left(\frac{1}{2}\sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_j, x_j)\right)$$

Достаточно определить $K(x, x') = h(x)^T h(x')$

Тогда оптимальные веса можно записать в виде: $w = \sum_{i=1}^{n} {\alpha_i}^* h(x_i) y_i$

Общее решение: $f(x) = h(x)^T w + w_0 = \sum_{i=1}^{n} {\alpha_i}^* K(x_j, x_j) + w_0$

МЕТОД ОПОРНЫХ ВЕКТОРОВ: ФОКУС С ЯДРОМ

Для построения нелинейной разделяющей границы **Kernel Trick**

достаточно выбрать ядро $K(x_j, x_j)$, имеющее смысл выражения для скалярного произведения преобразованных векторов.

Само преобразование выбирать не нужно!!!

$$\max_{\alpha}\left(\frac{1}{2}\sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_j, x_j)\right)$$

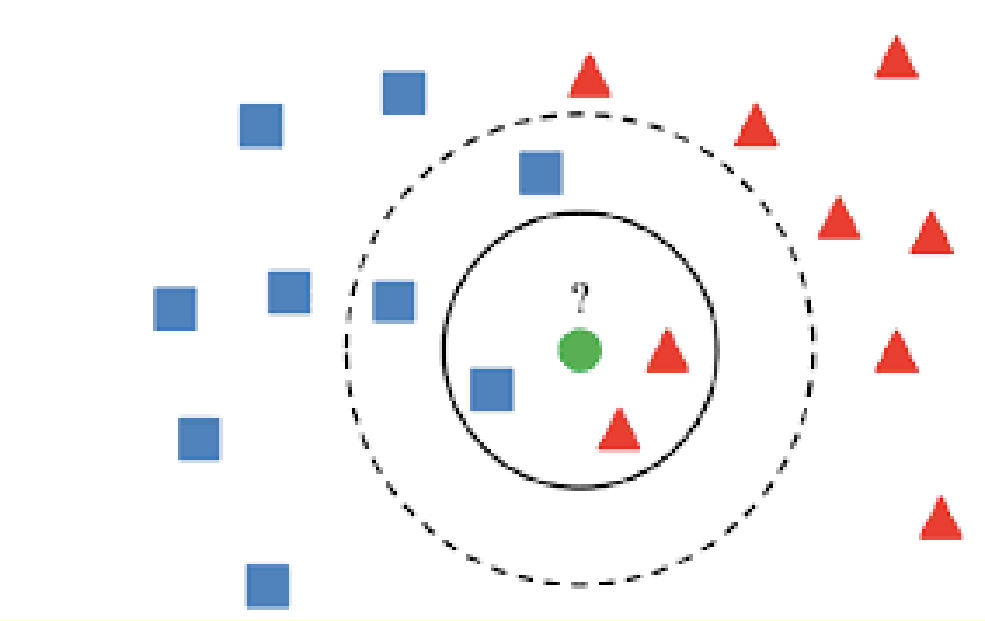
$$f(x) = \sum_{i=1}^{n}\alpha_i{}^{*} K(x_j, x_j) + w_0$$

- **Линейное ядро** $K(x, x') = x^T x'$
- **Полиномиальное ядро** $K(x, x') = (1 + x^T x')^{\,d}$
- **Ядро на радиальных базисных функциях**

$$K(x, x') = \exp(-\gamma|x - x'|^2)$$

МЕТОД K БЛИЖАЙШИХ СОСЕДЕЙ (K СРЕДНИХ)

- Один из самых простых алгоритмов регрессии
- Пусть дан тренировочный набор *X*, *y* и новый пример *x'*
- Тогда ответ для этого примера можно вычислить, взяв *K* ближайших примеров к *x'* по некоторой метрике (например, евклидово расстояние) и усреднив их ответы
- Если решается задача классификации – соседи «голосуют»



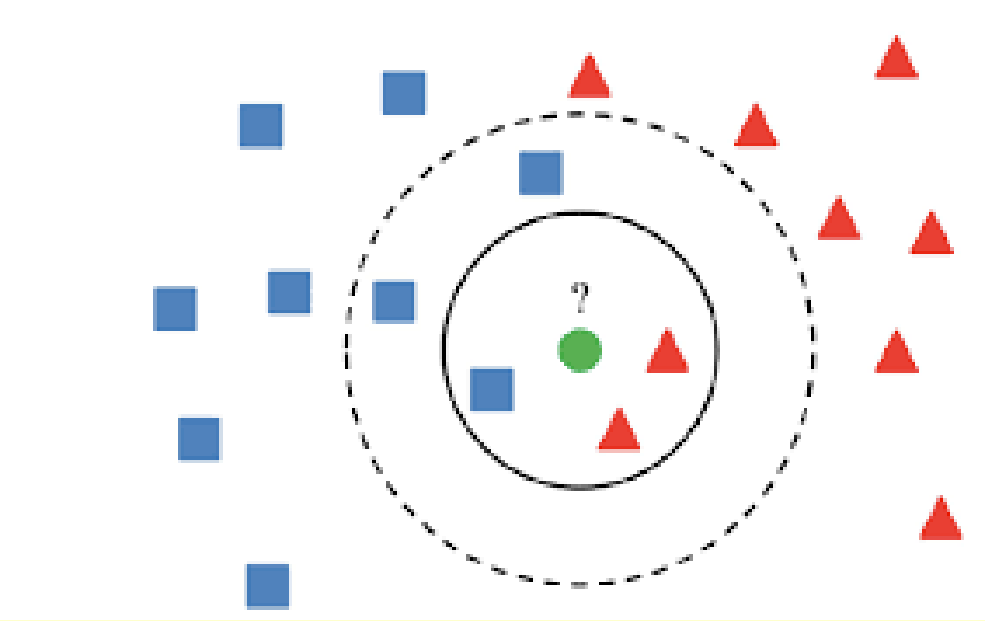
https://vitalflux.com/k-nearest-neighbors-explained-with-python-examples/

МЕТОД K БЛИЖАЙШИХ СОСЕДЕЙ (K СРЕДНИХ)

Метод имеет следующие гиперараметры:

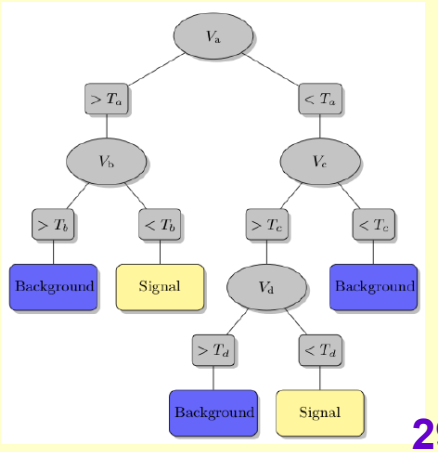
- **Количество соседей *K***
- **Метрика, по которой считается расстояние**
- **Вес каждого примера в окрестности**

  (например, можно брать веса, обратно пропорциональные расстоянию до примера)



ДЕРЕВО РЕШЕНИЙ

- **Алгоритм классификации имитирует «человеческую» логику принятия решений**
- **Для получения ответа на примере *x* необходимо рекурсивно пройти по дереву, каждый узел которого представляет собой развилку на основе критерия $I[x_j>C]$ (где x_j – *j*-й признак примера)**
- **Финальный ответ определяется в конечном узле дерева путем усреднения (голосования) по ответам в этом узле**
- **Пример дерева для отделения фона от сигнала:**



https://www.researchgate.net/figure/Schematic-of-a-decision-tree-Decision-nodes-for-each-event-variable-V-a-V-b-V-c_fig2_236935633

ПОСТРОЕНИЕ ДЕРЕВА РЕШЕНИЙ

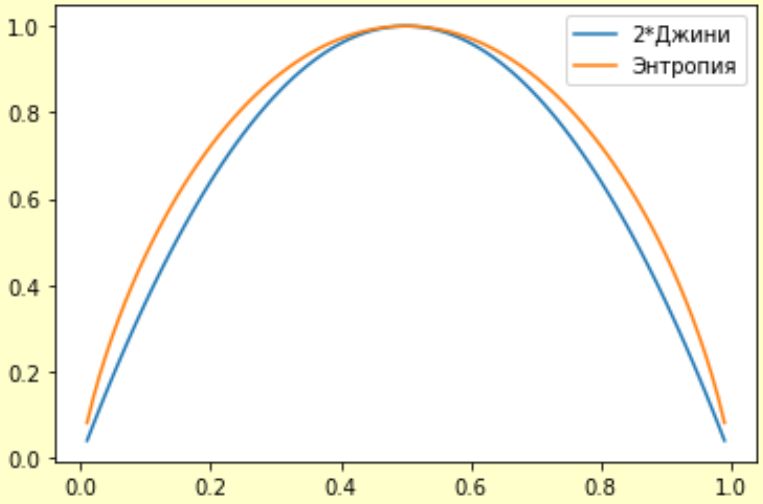
- Построение дерева решений – рекурсивный процесс
- Необходимо выбрать **признак и разбиение по нему**
  для исходного тренировочного набора, затем
  проделать то же самое для двух получившихся частей и т. д.
- Для проведения этой процедуры
  необходимо определить:
  - Метрику **неоднородности** ответов выборки
  - Сравнение однородности **подвыборок**,
    получившихся путем разбиения
  - Критерий **остановки** построения дерева

ЭНТРОПИЯ И ИНДЕКС ДЖИНИ

Определимся с метриками неоднородности

Пусть p_k- доля примеров выборки, принадлежащих к классу k (полное число классов K)

- **Индекс Джини** $\sum_{k=1}^{K} p_k(1-p_k)$
- **Энтропия** $-\sum_{k=1}^{K} p_k \log(p_k)$



КРИТЕРИЙ ВЫБОРА РАЗБИЕНИЯ

Для осуществления разбиения выборки по признаку необходима пара *(j,C)* – где *j* – номер признака,
C – значение, по которому производится разбиение.
Тогда разбиение можно записать в виде $I[x_j > C]$.

S(X) – критерий неоднородности выборки (энтропия, Джини, …).
Изменение неоднородности в результате разбиения (или **прирост информации**):

$$IG = S(X_0) - \frac{N_1}{N_0} S(X_1) - \frac{N_2}{N_0} S(X_2)$$

Здесь X_0 - исходная выборка, N_0 - размер исходной выборки,
$X_{1,2}$ - выборки, полученные в результате разбиения

АЛГОРИТМ ПОСТРОЕНИЯ ДЕРЕВА РЕШЕНИЙ

Упрощенная рекурсивная версия:

Для узла дерева:

1. Для каждого возможного разбиения каждого признака

посчитать прирост информации

2. Выбрать разбиение, соответствующее наибольшему приросту

3. Если прирост меньше порога – остановиться

4. Создать два дочерних узла, соответствующих разбиению

5. Для каждого узла повторить процедуру

АЛГОРИТМ ПОСТРОЕНИЯ ДЕРЕВА РЕШЕНИЙ

На деле есть гораздо больше нюансов.

Известные алгоритмы построения деревьев решений:

CART

Breiman L., et al., “Classification and regression trees”, 1984

Leo Breiman
1928 – 2005

C4.5

Quinlan, J. R. “C4.5: Programs for Machine Learning”, Morgan Kaufmann Publishers, 1993.

РЕГУЛЯРИЗАЦИЯ ДЕРЕВЬЕВ РЕШЕНИЙ

- Ограничение на максимальную глубину дерева
- Ограничение на количество рассматриваемых признаков в каждом узле (случайно выбираем m признаков из p и рассматриваем их разбиения)
- Ограничение на количество примеров в конечном узле
- Ограничение на количество примеров в разделяемом узле
- Ограничение на максимальное количество конечных узлов дерева

СВОЙСТВА ДЕРЕВЬЕВ РЕШЕНИЙ

- Склонны к переобучению
- Велика доля случайности – при повторных построениях могут получиться разные деревья
- Легко интерпретируются
- Можно приспособить для задачи регрессии, если использовать в качестве метрики СКО (MSE)

БУТСТРЭП И БЭГГИНГ

Бутстрэп (bootstrap)

Процедура получения множества выборок $(X_1, X_2, X_3, \ldots)$ из исходной выборки X.

Каждая выборка получается путем многократного выбора с повторением из исходной выборки.

Бэггинг (bagging)

Способ ансамблирования моделей. Обучаем исходную модель на N подвыборках, полученных с помощью бутстрэпа.

Усредняем предсказания полученных моделей.

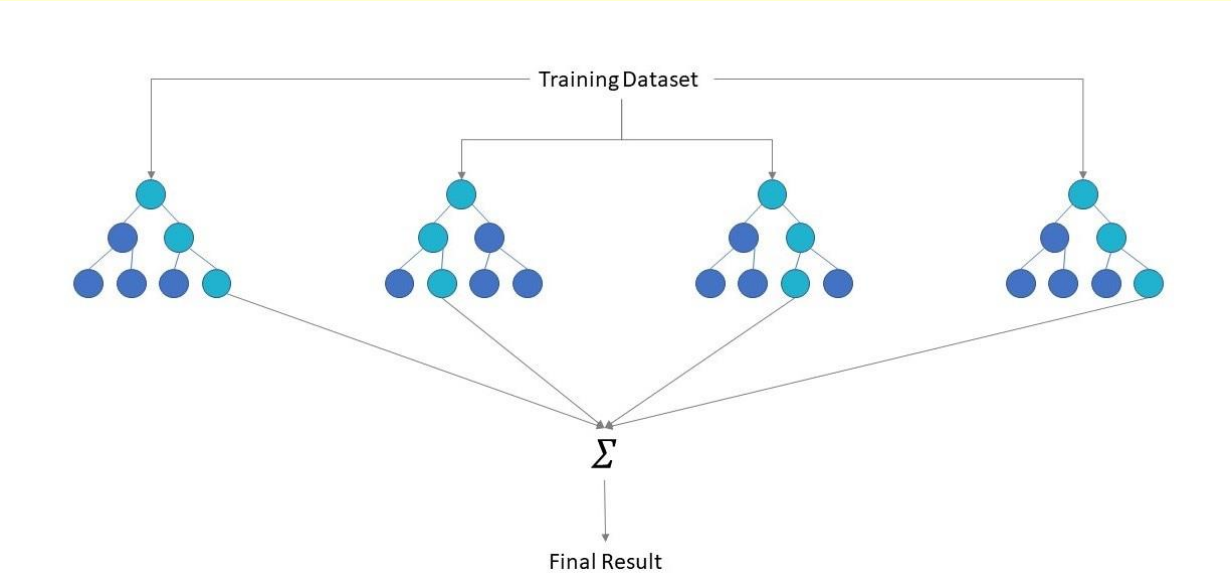
Теория подсказывает, что усреднение большого количества некоррелированных моделей уменьшает разброс предсказаний.

Надеемся, что бэггинг обеспечивает достаточную некоррелированность.

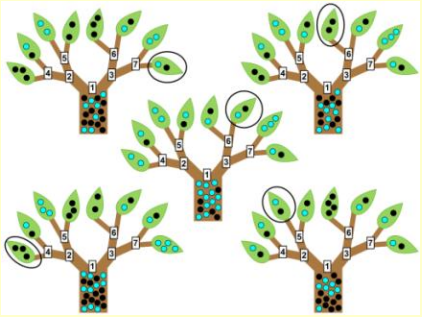
СЛУЧАЙНЫЙ ЛЕС

Случайный лес = Бэггинг над деревьями решений

- Алгоритм легко распараллеливается
- Качество обычно улучшается с ростом количества деревьев



https://www.ibm.com/cloud/learn/random-forest



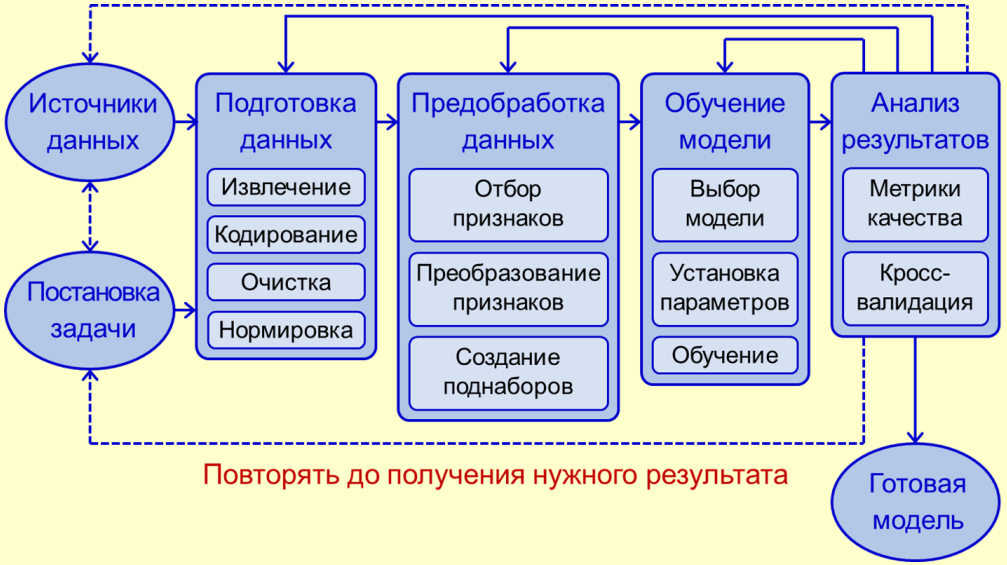
ГРАДИЕНТНЫЙ БУСТИНГ

- Бустинг – способ ансамблирования, при котором модели итеративно обучаются на остатках предыдущей модели
- Один из самых популярных алгоритмов машинного обучения – градиентный бустинг
- В качестве базовой модели в этом алгоритме используется дерево решений

Реализации градиентного бустинга:

- XGBoost https://xgboost.readthedocs.io/en/stable/
- CatBoost https://catboost.ai/
- LightGBM https://lightgbm.readthedocs.io/en/v3.3.2/

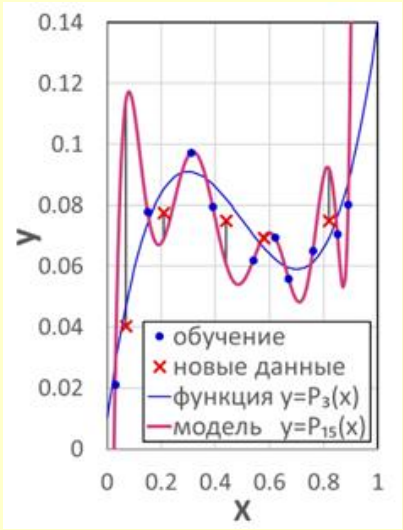
		Входные признаки					Выходные признаки			
	ID_прим	ВхПр1	ВхПр2	ВхПр3	...	ВхПрN	ВыхПр1	ВыхПр2	...	ВыхПрL
Примеры	Прим1	X11	X12	X13	...	X1N	Y11	Y12	...	Y1L
	Прим2	X21	X22	X23	...	X2N	Y21	Y22	...	Y2L
	Прим3	X31	X32	X33	...	X3N	Y31	Y32	...	Y3L
	Прим4	X41	X42	X43	...	X4N	Y41	Y42	...	Y4L
	...	...	...	...	...	...	...	...	...	...
	ПримP	XP1	XP2	XP3	...	XPN	YP1	YP2	...	YPL

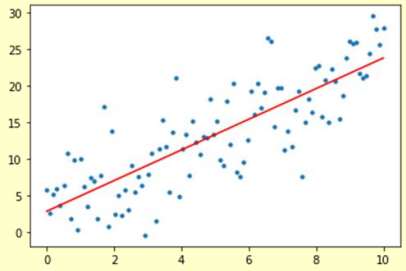


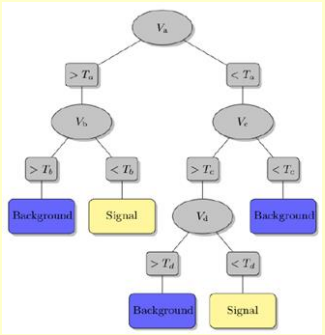
$$R^2 = 1 - \frac{\sum_{i=1}^{N}(a(x_i) - y_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

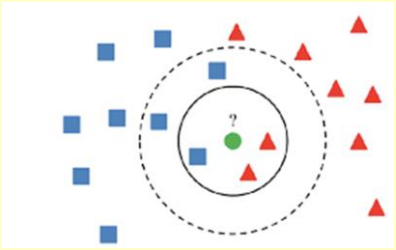
		Ответ модели	
		Positive	Negative
Истинное значение	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

Спасибо за внимание !









$$L_p = L(w) + \lambda \sum_{j=1}^{s} |w_j|^p$$