Обзорная лекция

ML

Вспомним что было

- Линейная регрессия

  Функция потерь: MSE

- Логистическая регрессия

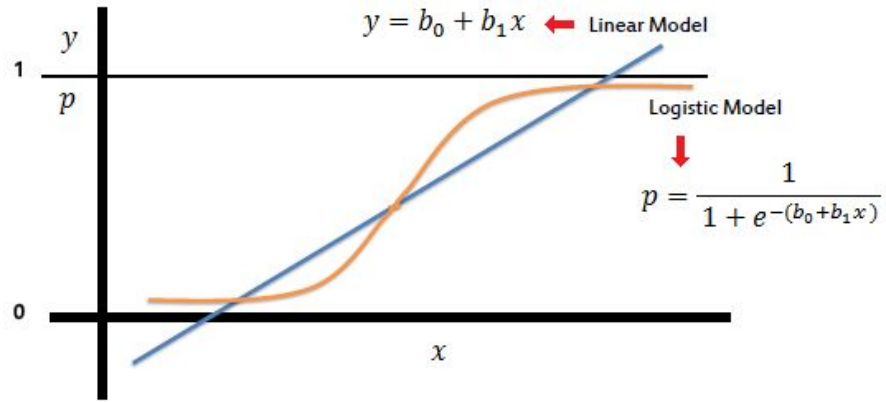
  Функция потерь: logloss

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$$

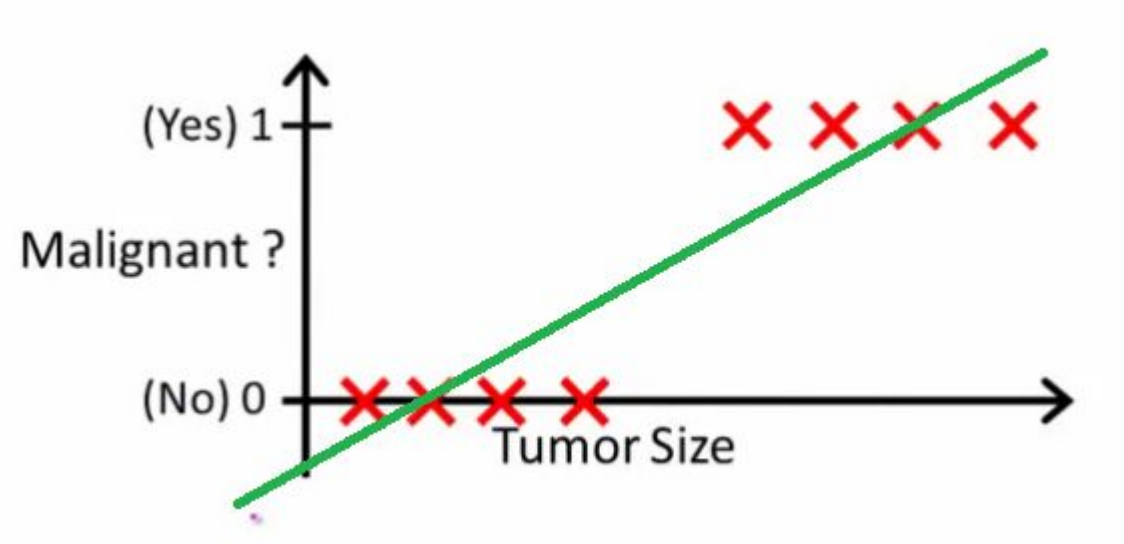
Вспомним регрессию

- почему бы не учить классификацию - регрессией на две метки?



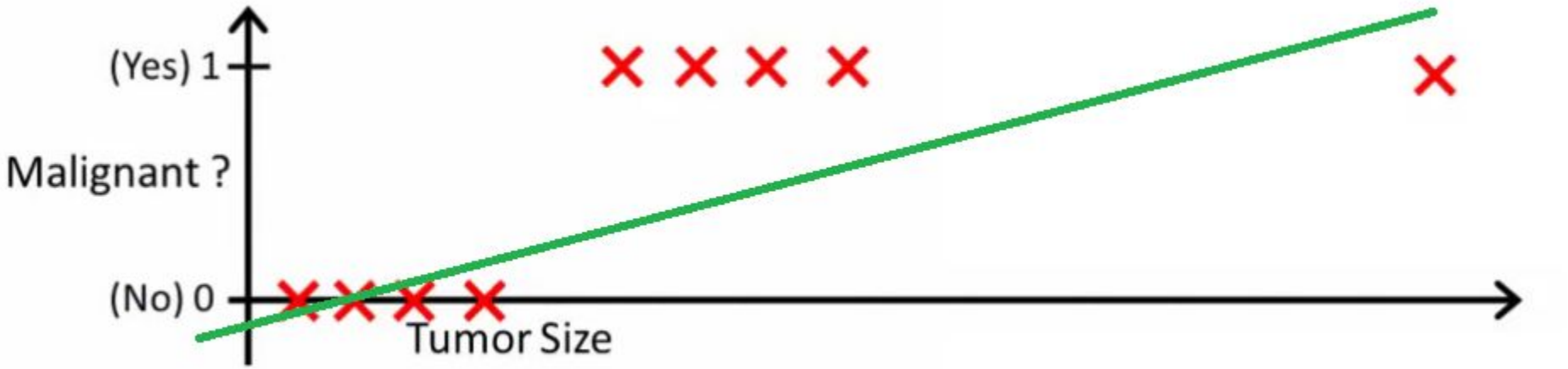
Вспомним регрессию

- почему бы не учить классификацию - регрессией на две метки?



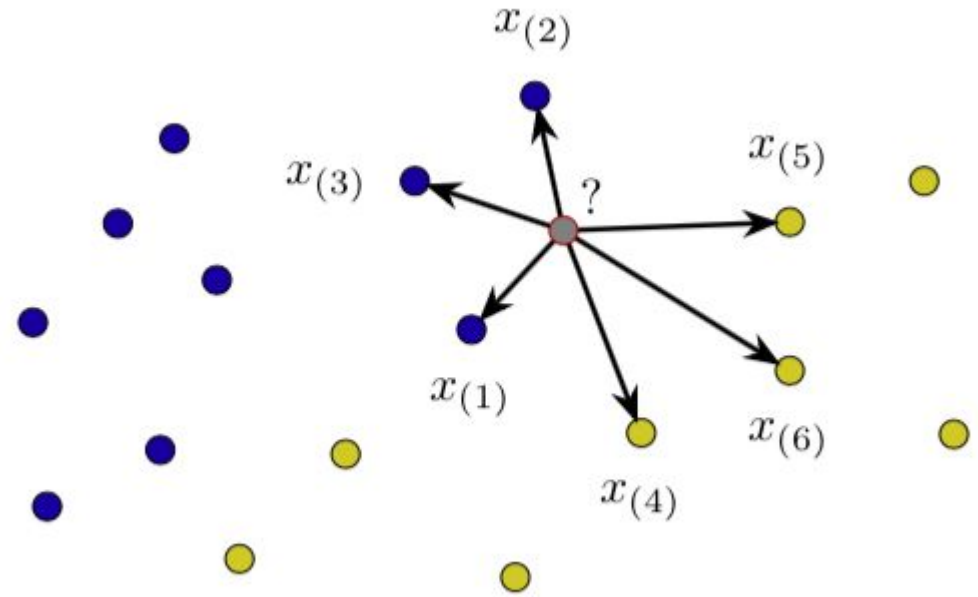
Вспомним регрессию

- почему бы не учить классификацию - регрессией на две метки?

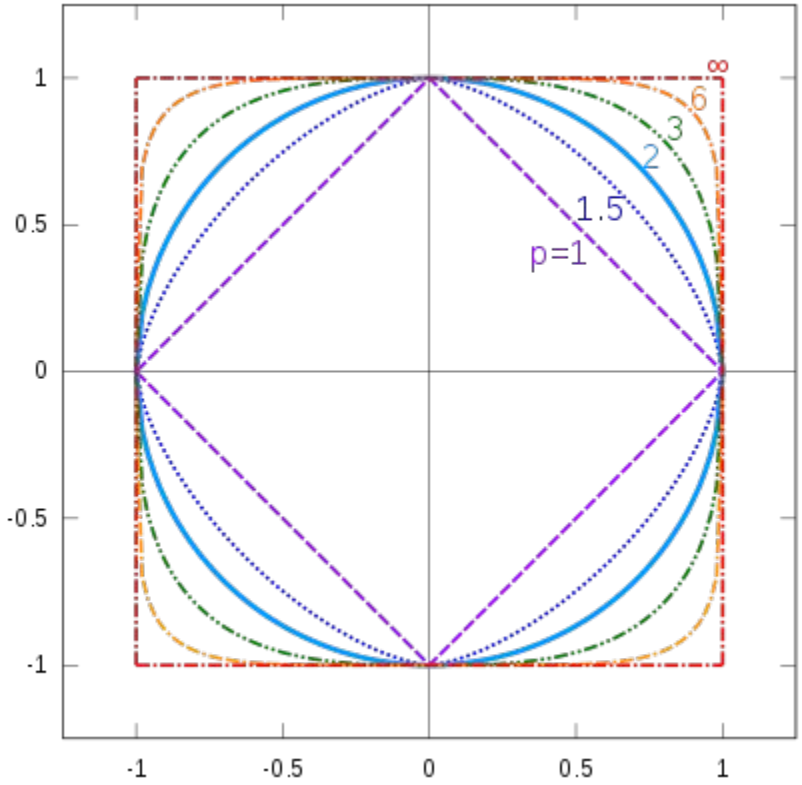


Метод K ближайших соседей

- как предсказывать
- как взвешивать (объекты, дистанция)
- как влияет метрика
-



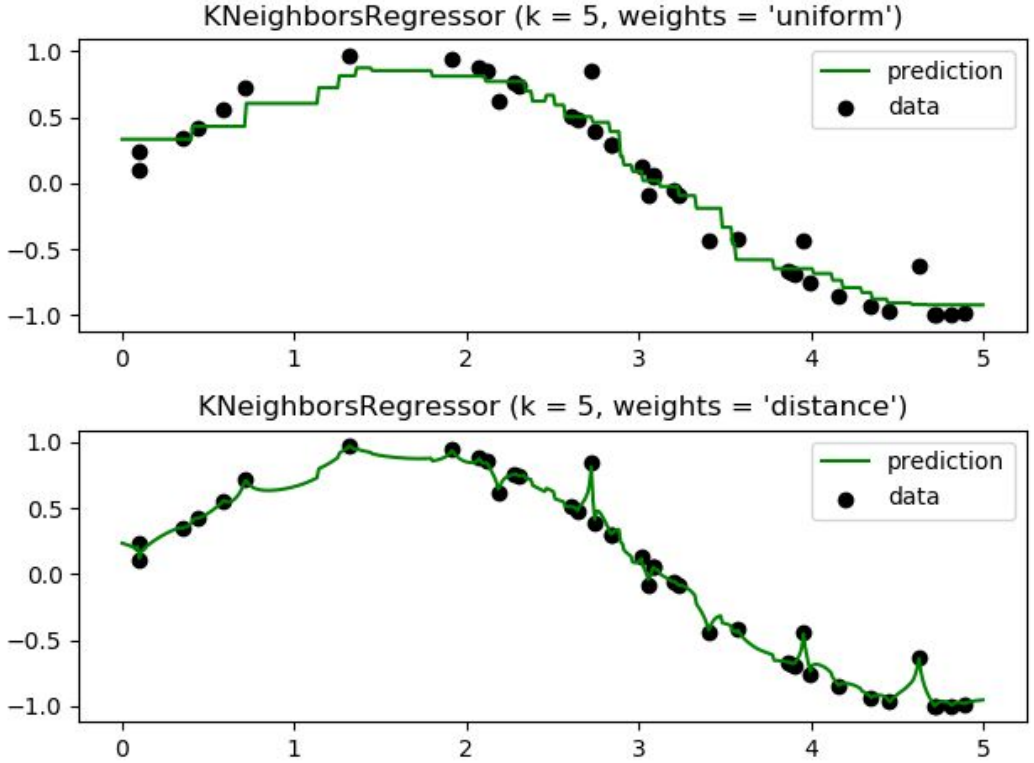
Lp нормы



https://en.wikipedia.org/wiki/Lp_space

Метод K ближайших соседей

$$\frac{\sum_{i=1}^{k} w(x_{(i)}) x_{(i)}}{\sum_{i=1}^{k} w(x_{(i)})}$$



Подбор гиперпараметров

- Какие гиперпараметры вы знаете?

- Как их подбирать?

Оценка адекватности модели

- Как понять что модель хороша?

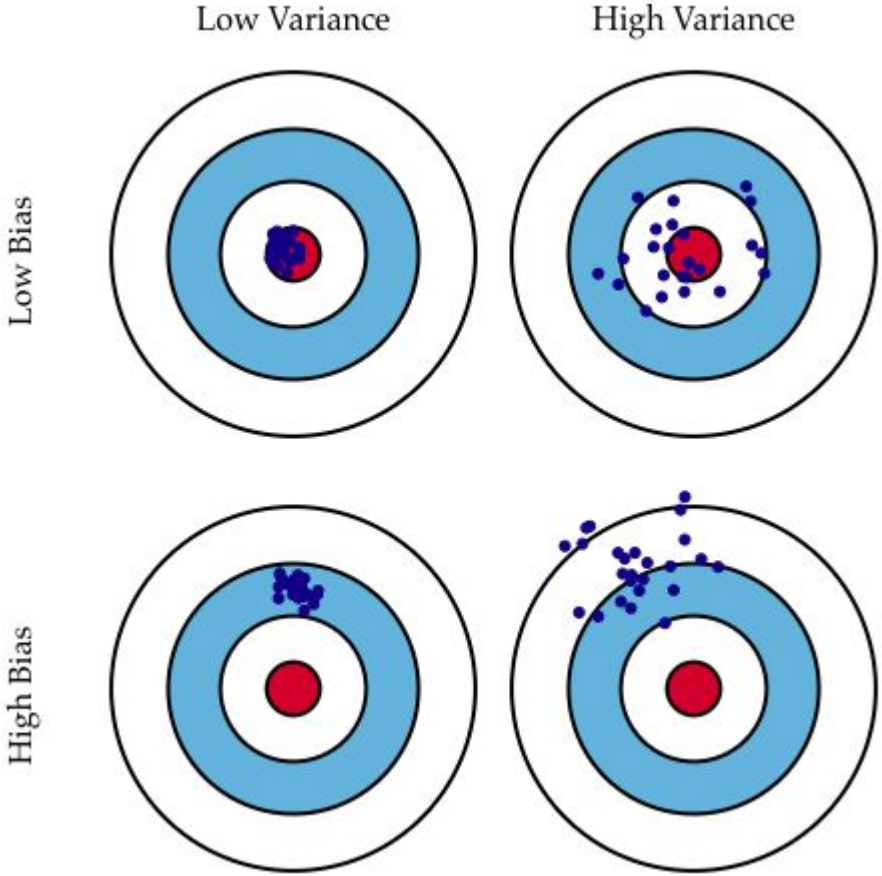
Bias Variance

$$Y = f(X) + \epsilon$$

$$Err(x) = E\left[(Y - \hat{f}(x))^2\right]$$

$$Err(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] + \sigma_e^2$$

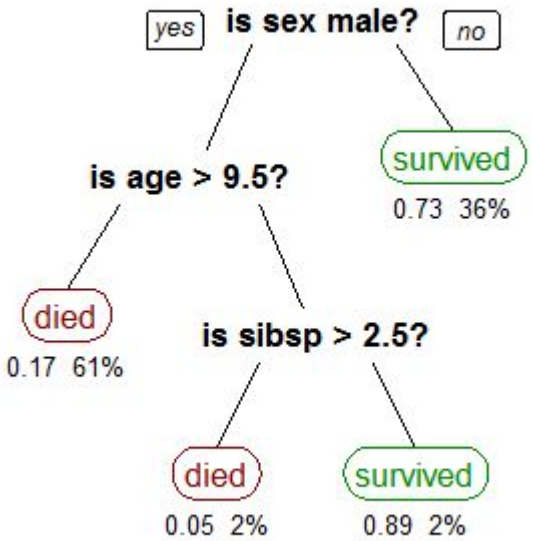
$$Err(x) = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$



http://scott.fortmann-roe.com/docs/BiasVariance.html

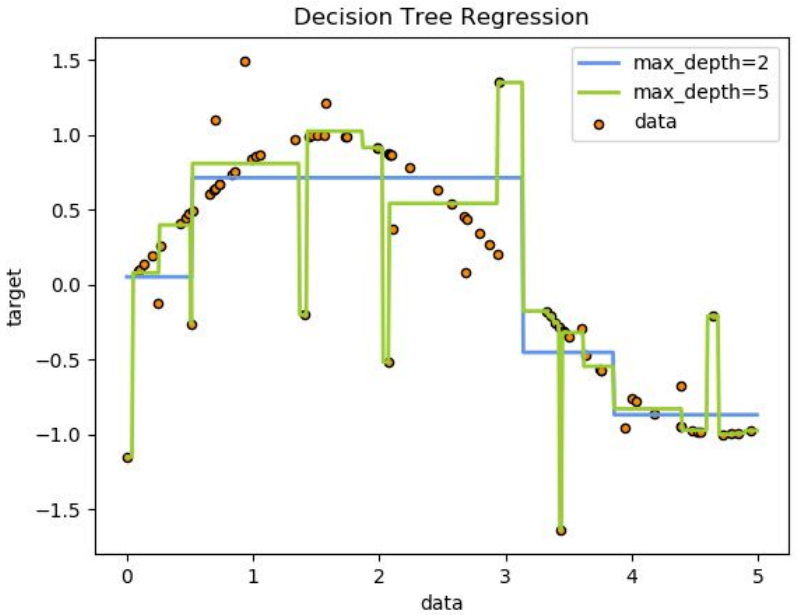
Дерево решений. Пример. Классификация.

sample = (age=8, sex=male, sibsp=4)



Деревья решений. Пример. Регрессия.

что такое регрессия на деревьях?
(сколькими константами мы
приближаем наши данные)



Построение

Какие бы деревья для нас были хороши?

- Глубокие - каждый объект в одном листе (оверфит)
- Не очень глубокие, но при этом не допускающие ошибок на обучении

Построение

- Для всей обучающей выборки X найдем наилучшее разбиение данных на две части по признаку feature и порогу threshold
  с точки зрения функционала Q(X, feature, threshold)

- Найдя наилучшие значения feature и threshold создадим корневую вершину и положим туда предикат X[feature] < threshold
  Объекты разобьются на две части X_left и X_right

- Рекурсивно повторим такую же процедуру для обеих частей
- Повторяем так пока не выполнится критерий останова

Критерий остановки

- максимальная глубина
- минимальное число объектов в листе
- максимальное количество листьев в дереве
- если все объекты в листе - одного класса
- если функционал качества не увеличивается хотя бы на k %
- итд

Построенное дерево

Каждому листу ставится в соответствие ответ
- классификация : (вектор вероятностей-долей или самый часто встречаемый класс)
- регрессия : (среднее, медиана значений элементов в листе)

Выбор разбиения и порога

$$Q(X, feature, threshold) = H(X) - \left(\frac{N_{left}}{N} H(X_{left}) + \frac{N_{right}}{N} H(X_{right}) \right)$$

H(X) - критерий информативности (impurity criterion)

Q() - показывает насколько “информативнее” (лучше) стало наше разбиение на две части (некоторый прирост информации)

будем стараться максимизировать Q

Энтропия

Классификация на К классов.

Энтропия показывает меру неупорядоченности системы (log (1) = 0).

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in X_m} I(y_i = k)$$

$$H(X_m) = -\sum_{k=1}^{K} p_{mk} \log_2 p_{mk}$$

$$Q(X, feature, threshold) = H(X) - \left(\frac{N_{left}}{N} H(X_{left}) + \frac{N_{right}}{N} H(X_{right}) \right)$$

Gini impurity

Классификация на К классов.
Максимизируем число объектов одного класса в одном поддереве.

$$p_{mk} = \frac{1}{N_m} \sum_{x_i \in X_m} I(y_i = k)$$

$$H(X_m) = \sum_{k=1}^{K} p_{mk} \, (1 - p_{mk}) = 1 - \sum_{k=1}^{K} (p_{mk})^2$$

$$Q(X, feature, threshold) = H(X) - \left(\frac{N_{left}}{N} H(X_{left}) + \frac{N_{right}}{N} H(X_{right}) \right)$$

MSE

Регрессия. Оптимизируем MSE.
Уменьшаем разброс целевой переменной.

$$\bar{y}_m = \frac{1}{N_m} \sum_{i \in N_m} y_i$$

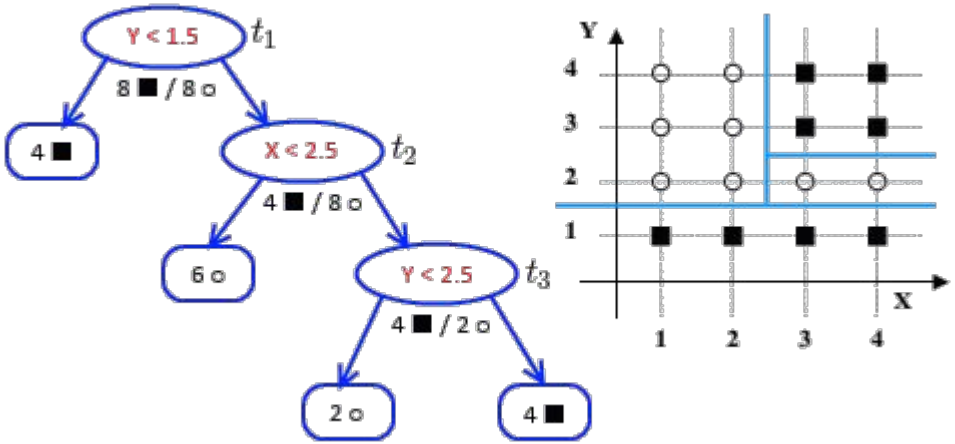
$$H(X_m) = \frac{1}{N_m} \sum_{i \in N_m} (y_i - \bar{y}_m)^2$$

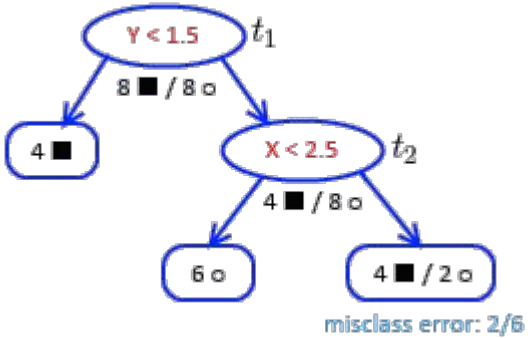
$$Q(X, feature, threshold) = H(X) - \left(\frac{N_{left}}{N} H(X_{left}) + \frac{N_{right}}{N} H(X_{right}) \right)$$

Стрижка дерева

Строим дерево максимальной глубины.

Вводим функционал который штрафует за глубину дерева $R_\alpha(T) = R(T) + \alpha|T|.$





Дерево

- Само по себе дерево относительно слабый алгоритм для предсказания

- Но оно умеет:
- - предсказывать (уже хорошо)
- - оценивать важности признаков
- - быть одним из алгоритмов в композиции решающих алгоритмов

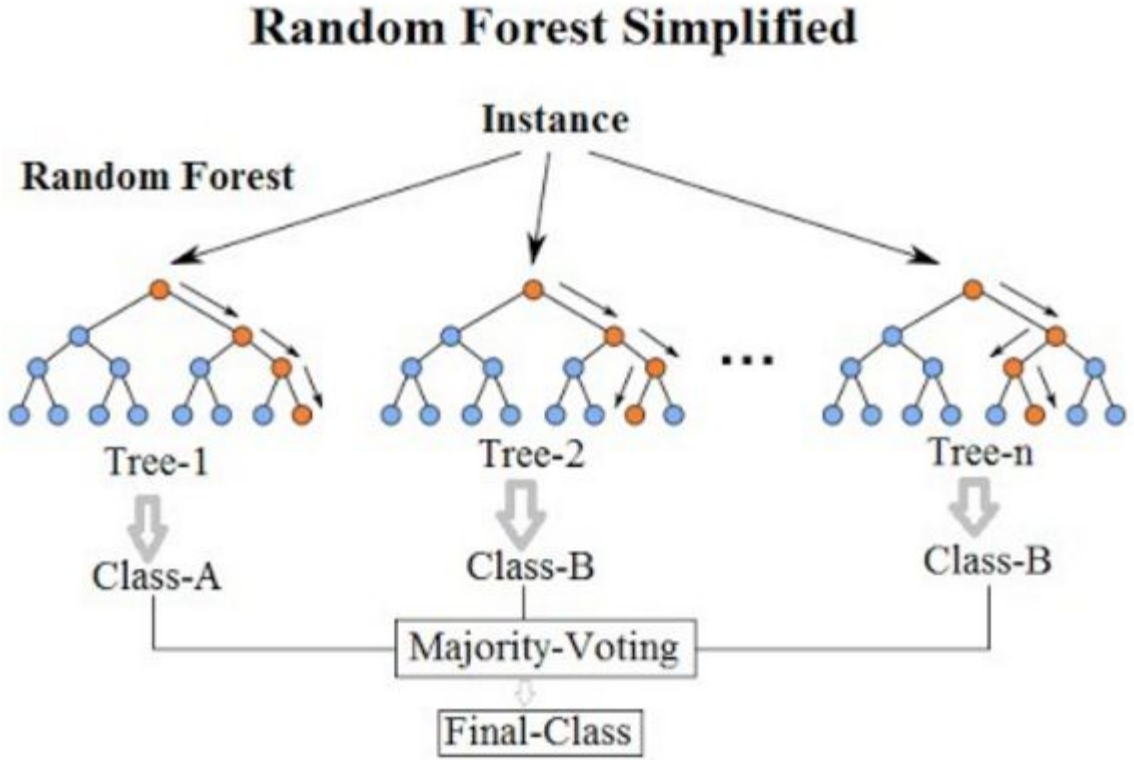
Ансамбль

- Есть несколько алгоритмов
- Можно взять их ответы и устроив голосование - выдать новый (можно взвешивать ответы)

  эффект “мудрость толпы”

Случайный лес

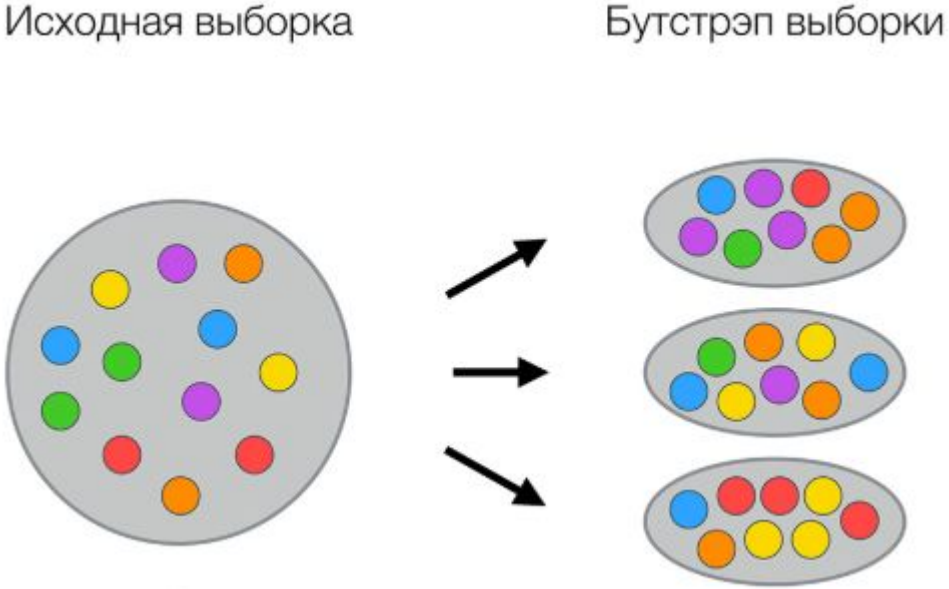
Leo Breiman



https://habr.com/company/ods/blog/324402/

Как сделать деревья не коррелирующими

- обучать на разных данных
- обучать на разных признаках

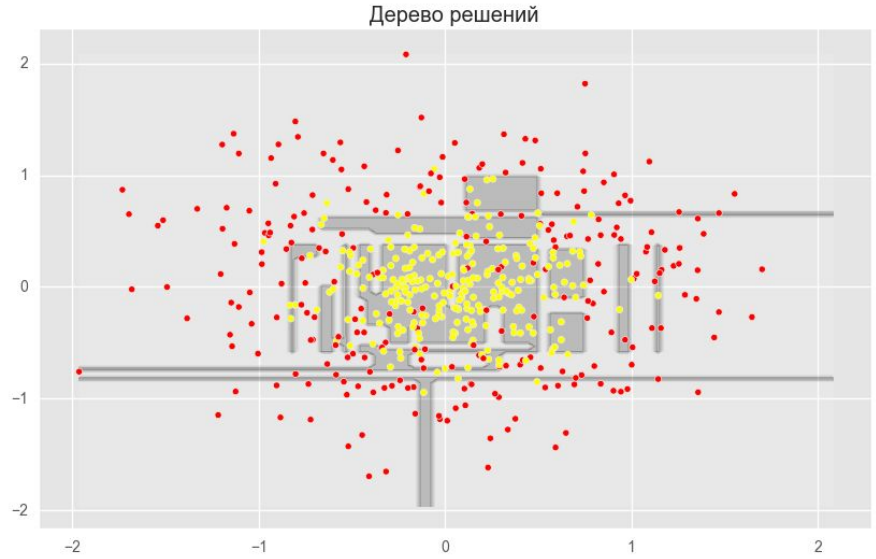


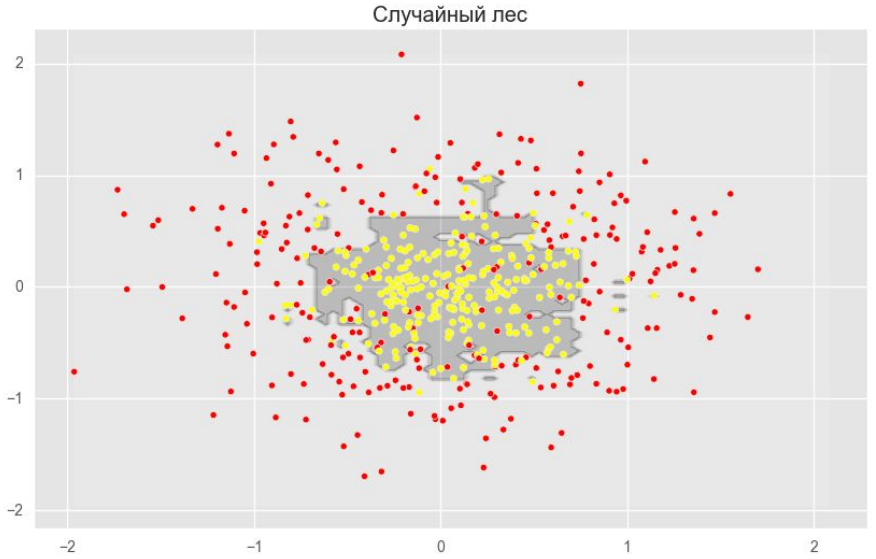
Случайный лес

- Генерируем N случайных подвыборок с помощью бутстрапа
- Обучаем N деревьев (каждое дерево - на своей подвыборке)
- - чтобы добавить еще больше рандомизации, при построении деревьев выбираем лучшее разбиение среди случайного подмножества признаков

  (долю подмножества признаков задаем как гиперпараметр - классификация корень из числа признаков, регрессия - треть признаков, но лучше подбирать)
-

Случайный лес

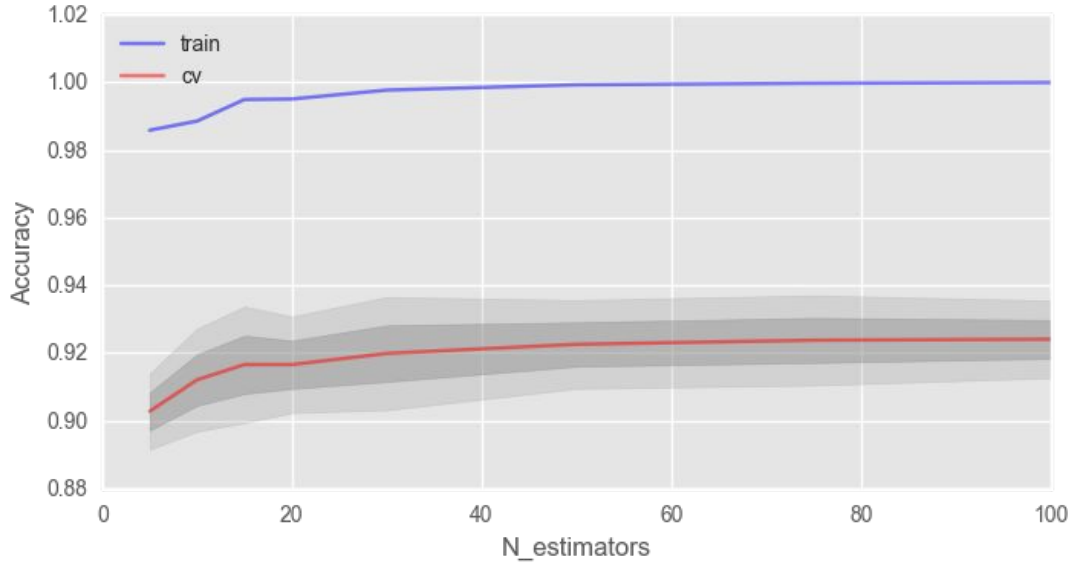




https://habr.com/company/ods/blog/324402/

Случайный лес

- смещение такое же как у одного алгоритма (или чуть хуже)
- разброс меньше (тем меньше, чем сильнее некоррелированы деревья)
-



Случайный лес

параметры

https://dyakonov.org/2016/11/14/%D1%81%D0%BB%D1%83%D1%87%D0%B0%D0%B9%D0%BD%D1%8B%D0%B9-%D0%BB%D0%B5%D1%81-random-forest/

Почитать

- https://habr.com/company/ods/blog/322534
  https://habr.com/company/ods/blog/324402/

- https://github.com/esokolov/ml-course-hse/blob/master/2018-fall/lecture-notes/lecture07-trees.pdf

- https://github.com/esokolov/ml-course-hse/blob/master/2018-fall/lecture-notes/lecture08-ensembles.pdf

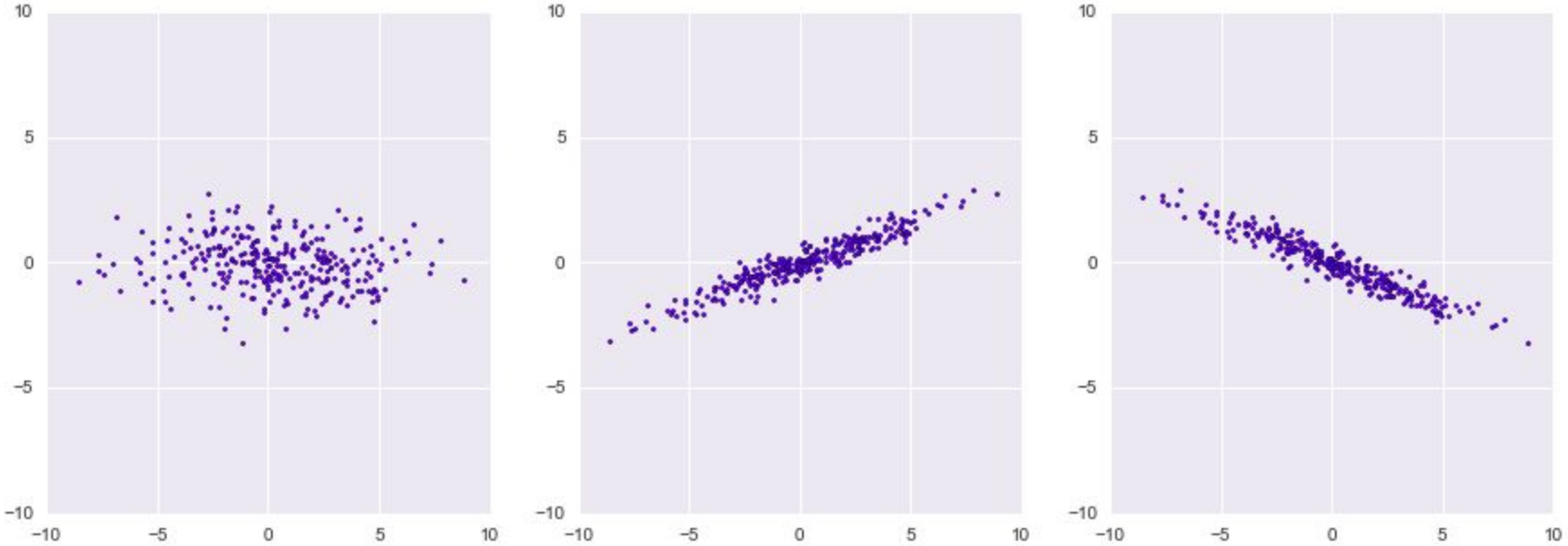
Вспышка холеры на Брод-стрит

[https://ru.wikipedia.org/wik
BA%D0%B0_%D1%85%D
D%D0%B0_%D0%91%D1
D0%B8%D1%82](https://ru.wikipedia.org/wik)

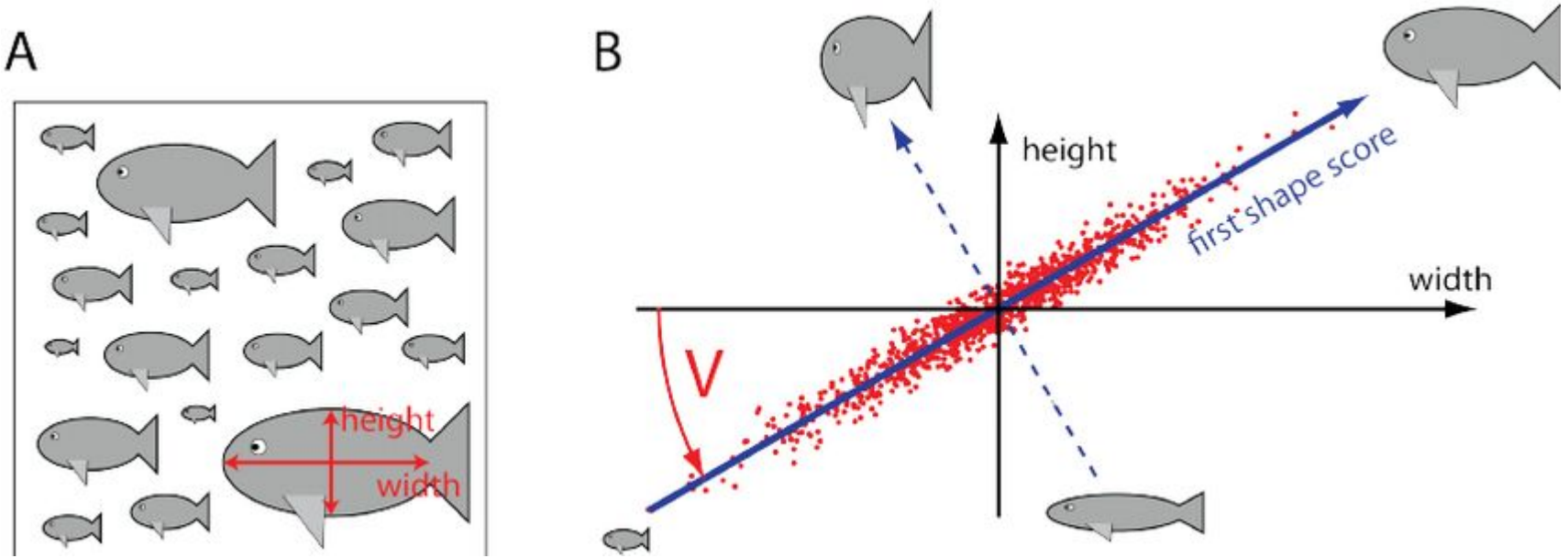
Понижение размерности

- устранение избыточных признаков
- возможность визуализации многомерных данных

Понижение размерности



Метод Главных Компонент



МГК

- будем искать такие “новые оси” - дисперсия проекции данных на которые будет максимальной

  тем самым будем стараться сохранять информацию

  Оптимизационные постановки:
  - минимизация отклонения точек от поверхности
  - сингулярное разложение
  - поиск собственных векторов и с.з.

sne (t-sne)

Хочется сохранить структуру данных, но при этом уменьшить размерность
- близкие точки должны лежать близко
- далекие - далеко

Идея: Попробуем уйти от дистанции и сохраним распределение точек

$$p_{j|i} = \frac{exp(-\parallel x_i - x_j \parallel^2 / 2\sigma_i^2)}{\sum_{k \neq i} exp(-\parallel x_i - x_k \parallel^2 / 2\sigma_i^2)}$$

(насколько j близка к i при гауссовом распределении вокруг i)

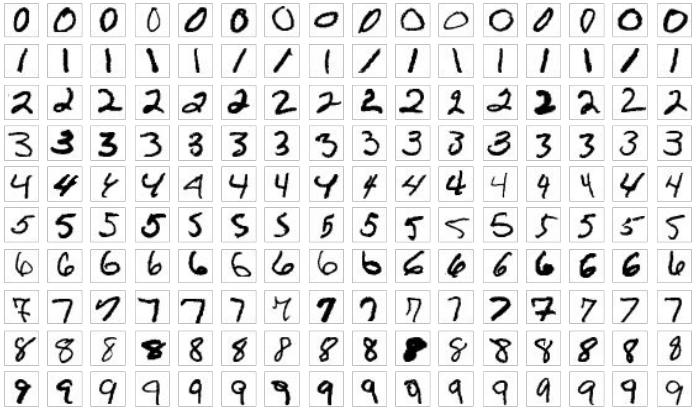
Минимизируем дивергенцию Кульбака-Лейблера

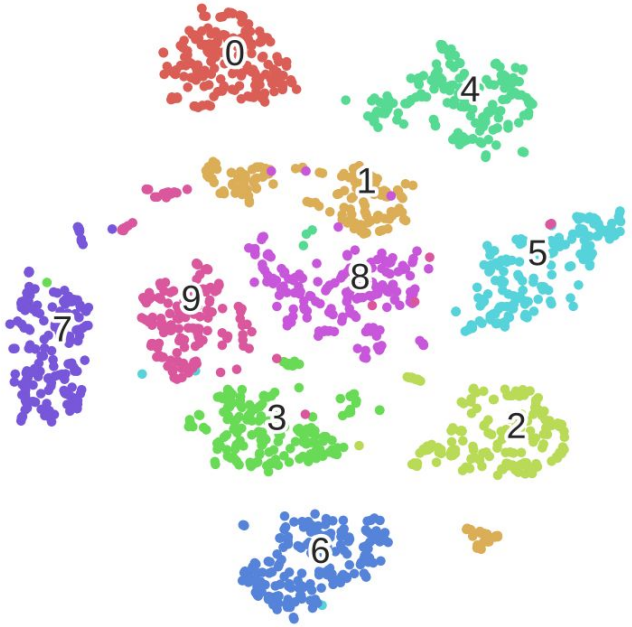
https://habr.com/post/267041/
https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf

t-sne

Пример

датасет рукописных цифр MNIST





Предобработка признаков

Для линейных моделей

- шкалирование (лучше для SGD), если есть регуляризация.
- переход в спрямляющее пространство в случае нелинейной зависимости
  (добавление полиномиальных признаков)
- преобразование таргета (например логарифм для тяжелохвостых распределений - цена на недвижимость - будет лучше если распределение будет нормальным)
- преобразование не_”нормальных” признаков к “нормальным”
- бинаризация категориальных признаков (one_hot)
- бинаризовать (дискретизация) непрерывный признак (как гистограмма, в какой бин попали)

Предобработка признаков

Категориальные признаки для деревьев:

- label encoding (можно несколько разных и выбрать лучшее)
- counter encoding (не всегда хорошо, хотя лучше случайного label_encoding)
- бинаризация категориальных признаков (one_hot)
- Кодирование средним по таргету

Редкие категории -> одну категорию

пропущенное значение -> Новая категория

На тесте новый категориальный признак?

Label encoding. One hot encoding. Counter encod.

	city	class	degree	income	city_le
0	Moscow	A	1	10.2	2
1	London	B	1	11.6	1
2	London	A	2	8.8	1
3	Kiev	A	2	9.0	0
4	Moscow	B	3	6.6	2
5	Moscow	B	3	10.0	2
6	Kiev	A	1	9.0	0
7	Moscow	A	1	7.2	2

	city	class	degree	income	city=0	city=1	city=2
0	Moscow	A	1	10.2	0	0	1
1	London	B	1	11.6	0	1	0
2	London	A	2	8.8	0	1	0
3	Kiev	A	2	9.0	1	0	0
4	Moscow	B	3	6.6	0	0	1
5	Moscow	B	3	10.0	0	0	1
6	Kiev	A	1	9.0	1	0	0
7	Moscow	A	1	7.2	0	0	1

	city	class	degree	income	city_c
0	Moscow	A	1	10.2	4
1	London	B	1	11.6	2
2	London	A	2	8.8	2
3	Kiev	A	2	9.0	2
4	Moscow	B	3	6.6	4
5	Moscow	B	3	10.0	4
6	Kiev	A	1	9.0	2
7	Moscow	A	1	7.2	4

Target encoding

	city	class	degree	income	city_mean_income
0	Moscow	A	1	10.2	8.5
1	London	B	1	11.6	10.2
2	London	A	2	8.8	10.2
3	Kiev	A	2	9.0	9.0
4	Moscow	B	3	6.6	8.5
5	Moscow	B	3	10.0	8.5
6	Kiev	A	1	9.0	9.0
7	Moscow	A	1	7.2	8.5

Отбор признаков

Зачем?

- Легче переобучиться на большом числе признаков
- Проще перебирать меньшее пространство признаков
- Убрать шумовые признаки

Как?

- жадно-итеративно, умно (направленный поиск подмножеств признаков)
- с помощью моделей (линейные с регуляризацией, деревянные) смотреть важность признаков

https://habr.com/company/ods/blog/325422/
https://www.youtube.com/watch?v=n4qKbFd25Sk

Генерация

Что почитать про генерацию признаков

http://www.machinelearning.ru/wiki/images/4/46/PZAD2017_08_featureengenearing.pdf

https://habr.com/company/mailru/blog/346942/