ИТМО

Basic ML
Не магия наука

Что такое Машинное обучение?

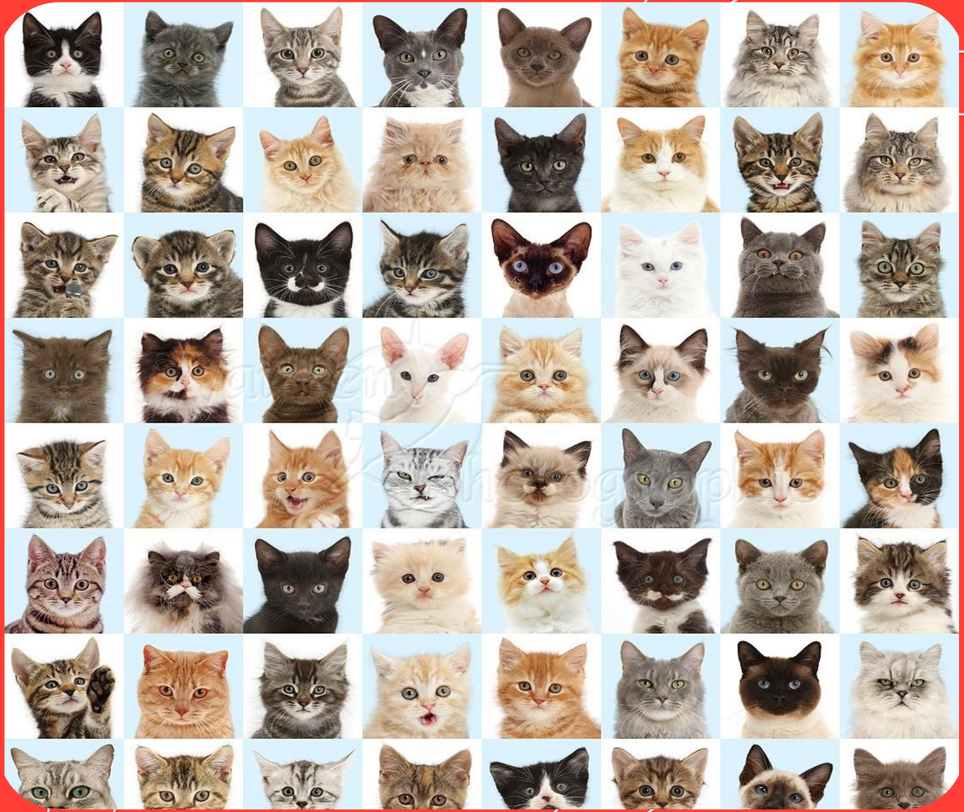
Машинное обучение – это наука, изучающая алгоритмы, автоматически улучшающиеся благодаря опыту.

Какие задачи решает? Те, которые “легки” для человека, но крайне трудно запрограммировать:

- Перевод текста
- Постановка диагноза
- Ранжирование документов по поиску
- Классификация картинок

Данные, предсказания модели.

ITMO



Данные бывают разные, но в общем и целом для моделей это:

- примеры
- предсказания

Набор примеров – датасет.

Часть датасета – выборка.

Примеры → предсказания – модель.

Какие задачи мы решаем?

Базовые задачи

$Y = R$ // $Y = R^M$ – Регрессия – пытаемся предсказать число.

$Y = 0,1$ // $Y = [1, K]$ – Классификация – предсказываем класс или классы

Задачи со звездочкой

$Y = [0,1]^K$ – Многоклассовая классификация с пересечением

Y - конечное упорядоченное множество – Ранжирование

Задачи сложнее и без учителя

Сегментация.
Перевод текста.

Создание новых объектов из ничего - дифузионки.

Кластеризация, “угадай следующее слово” (в сыром тексте) – без учителя.

А если все соединить?

Ответ:

Качество модели

При обучении модели в разработке используют разные метрики:

- бизнес метрики
- онлайн метрики
- асессоры
- оффлайн метрики

Мы не бизнес. Нам нужно понять, как мы хорошо обучили модель на текущих датасетах:

- Доли правильных ответов
- Разница между предсказанным и истинным значением
- Для ранжировки - доля пар документов, которые упорядочены неправильно

Все это можно назвать “функциями потерь”.

Немного математики

Начальные метрики

TP - сделали верное +предсказание
FP - сделали НЕверное +предсказание

TN - сделали верное -предсказание
FN - сделали НЕверное -предсказание

Predicted class \ Actual class	+	-
+	TP True Positives	FP Flase Positivies
-	FN Flase Negatives	TN True Negatives

Серьезные метрики

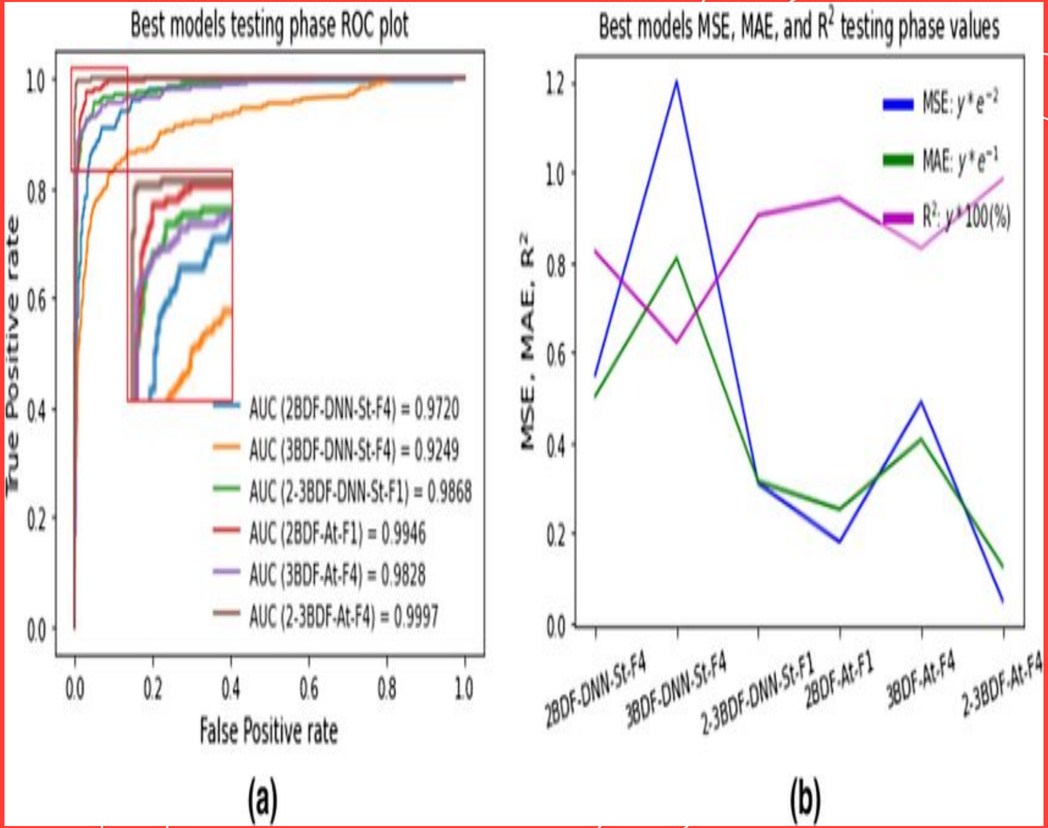
Accuracy = TP+TN / (TP+TN+FP+FN)

Precision = TP / (TP+FP)

Recall = TP / (TP+FN)

F1 = 2 * (Precision * Recall) / (Precision + Recall)

“Крутые” метрики



MAE = (1/n) * ∑ | y[i] - y’[i] |

MSE = (1/n) * ∑ (y[i] - y’[i])^2

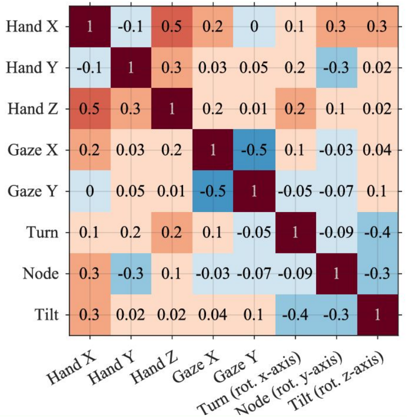
RMSE = √MSE

MAPE = (100% / n) *
* ∑ | (y[i] - y’[i]) / y[i] |

LOG LOSS = - (1/n) * ∑ [y[i] *
log(p[i]) + (1 - y[i] * log(1-p[i])]

Преобразование данных

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S



1. Машина не любит тип string.
2. Машина не любит разброс в датасете.
3. Машина не любит СИЛЬНО коррелирующие данные
4. Машина не любит слабо коррелирующие данные
5. Машина не любит выбросы
6. Машина не любит пропуски
7. Машина любит, когда данные находятся в интервале [-1;1]
8. Машина любит, когда данных МНОГО (очень. НУ ОЧЕНЬ МНОГО)
9. Машина любит, когда вы сделали датасет информативным, каждый признак несет в себе суть для таргета.

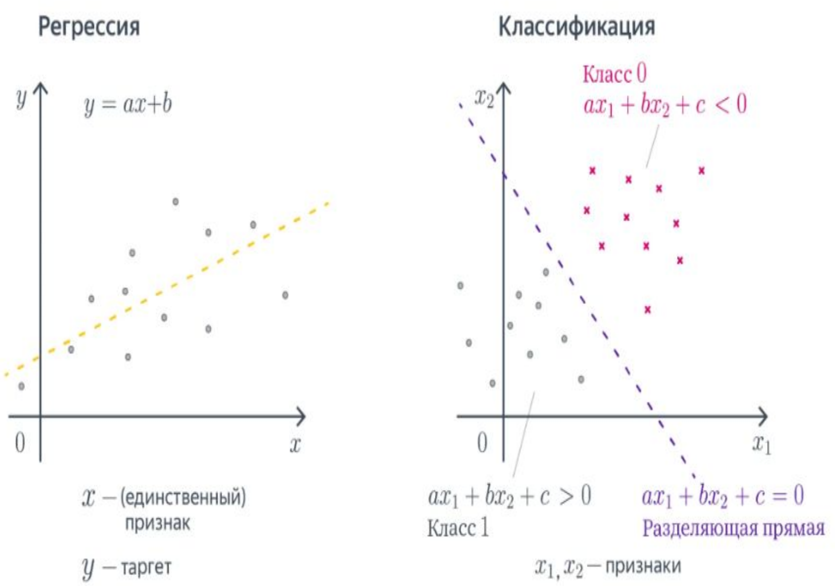
Так я модель сделаю уже?

У линейных моделей достаточно понятная и “прямая” формула:

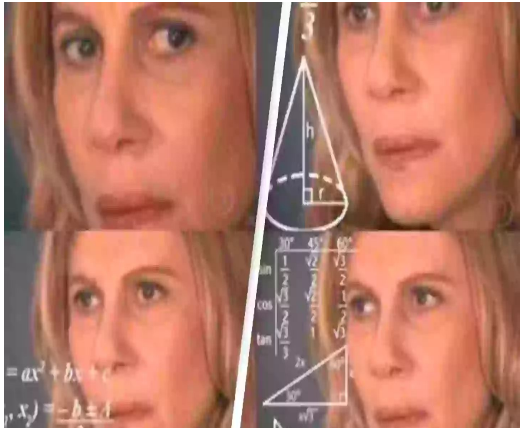
y = w1*x1 + … wi*xi + w0 == <x, w> + w0

В задачах регрессии нам важно расположить “прямую” настолько близко к точкам, насколько возможно.

В задачах классификации нам надо разделить два класса, допуская выборосы.



Oh, no, it’s a math!



Как нам подкрутить веса, чтобы все завелось?

Xw = w1*x1 + ... wi*xi, w - веса, x - признаки
Пусть y = y_pred + y_ort, т.е.
y_ort = y - Xw $\perp$ x1,.xi → в матрички → $X^T * (y - Xw) = 0$ →

→ $w = (X^T * X)^{-1} X^T * y$

Проблемы:

1. Обращение матрицы 1e20 x 1e20
2. $X^T X$ в 99.9% обратима, но чем больше признаков, тем чаще мы встречаем линейные зависимости. Малые “возмущения” таргета вызовут большие изменения w.

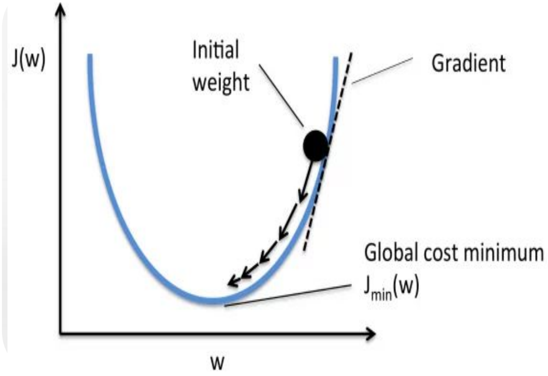
Немного схитрим

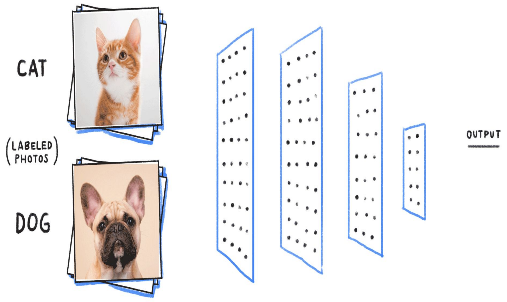
Посчитаем, как сильно мы ошибаемся и попробуем это уменьшать?

```
y_pred = Xw
err = y_pred - y_true
changes = (2 * XT * err) / N
w -= alpha * changes
```

Поздравляю! А теперь повторим раз 500 на одном датасете!

Что за колдунство?





$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Binary Cross-Entropy / Log Loss

На самом деле, мы нашли минимум

Да, псевдокод из прошлого слайда, по сути – решение всех проблем линейной регрессии *базово*.

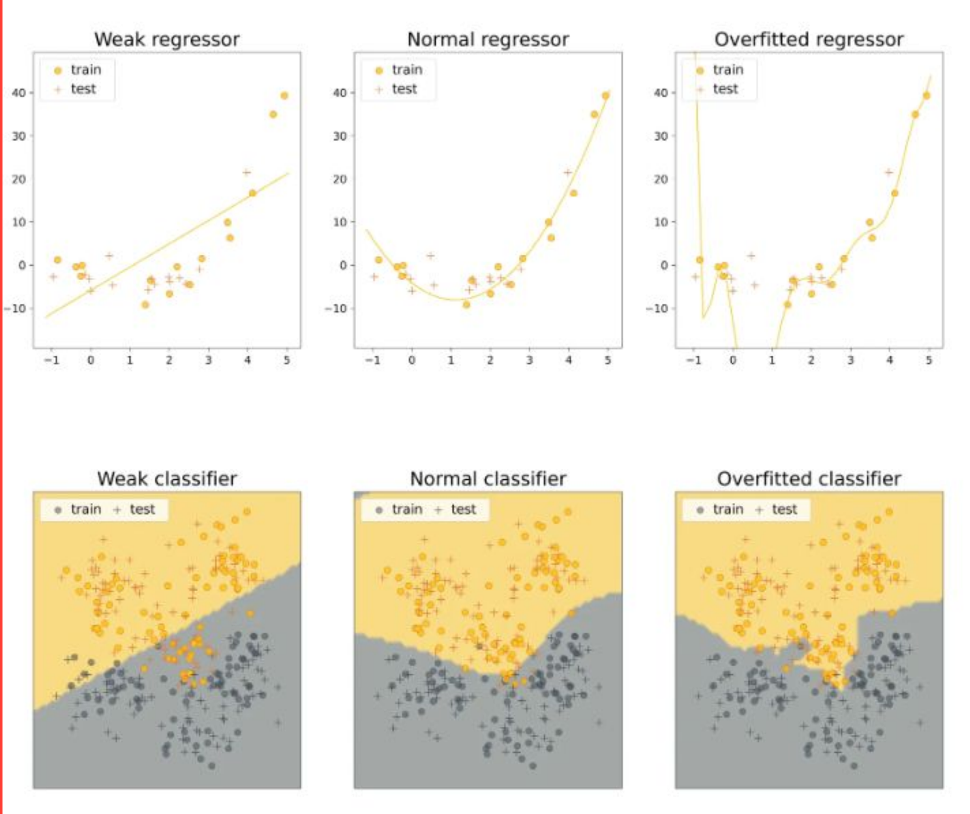
Ладно, а с классами что?

Тут немного другой псевдокод... Нам надо ошибку завернуть в сигмоиду.

А “изменения” той страшной штукой?

Если взять производную от LogLoss’а, то получится, на деле, тот же самый mse :)

Первые проблемы

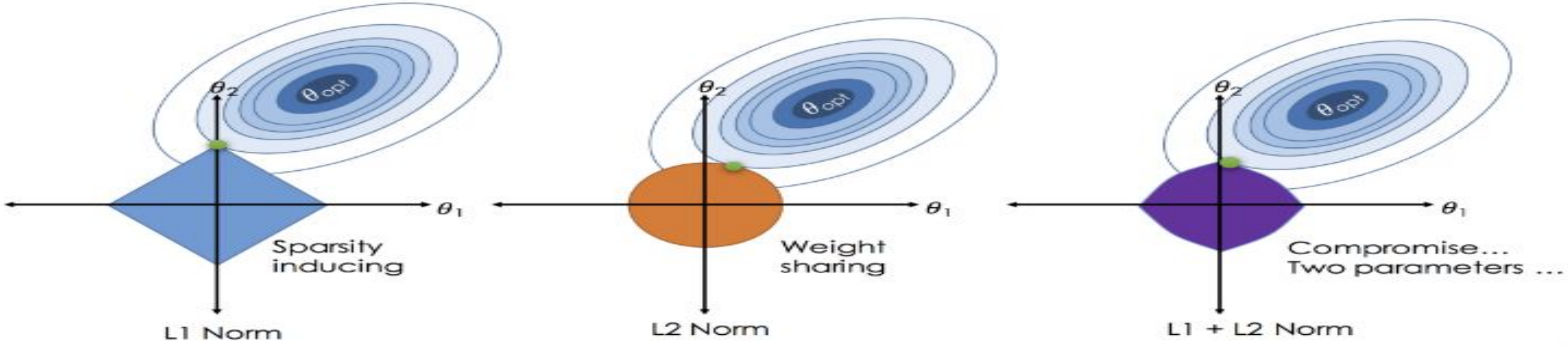


Недообучение – модель не нашла закономерности и решает задачу на “авось”.

Переобучение - модель просто запомнила весь датасет.

Для решения таких проблем используется деление на подвыборки (train, test, validate) и регуляризация

Регуляризация



L1 или lasso

Штрафует за сумму абсолютных весов.

Функция потерь:

changes = our_err +λ*∑ |y[i]|

L2 или Ridge

Штрафует за сумму абсолютных весов.

Функция потерь:

changes = our_err +λ*∑ (y[i])^2

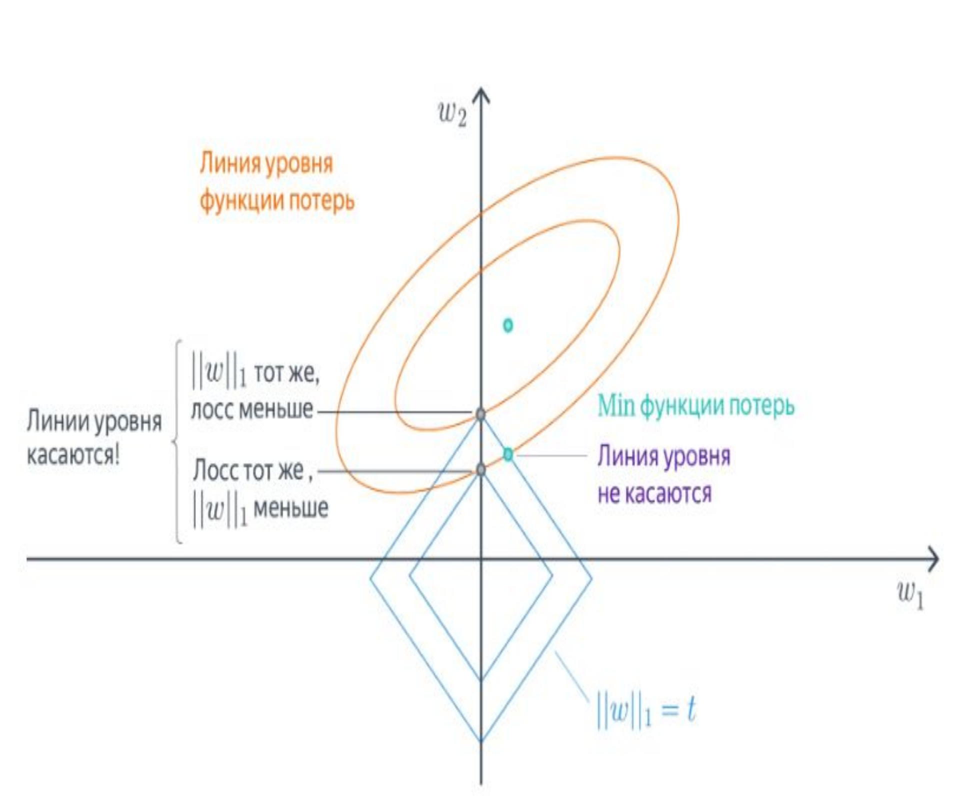
Combine them! or ElasticNet

Пытаемся найти компромисс между L1 и L2

Функция потерь:

changes = our_err + λ1*∑ |y[i]| + λ2*∑ (y[i])^2

Немного подробнее про L1



При применении l1, веса по некоторым признакам зануляются → они почти не влиют на наш таргет → мы автоматом “лечимся” от мультиколлинеарности.

Если твой сосед - я, то ты == я. KNN

У нас есть выборка из N записей. Запомнили ее - X

Прилетает какой-то объект u. Ищем K соседей (по сути, наименьшие K расстояний от u до каких-то объектов из X)

Какие они в *большинстве своем*? Значит и он такой же.

Глубже в соседей.

ИТМО

Как математически найти метку класса?

1. Пройдись по всем классам (y), которые лежат в выборке.
2. Посчитай сколько среди k ближайших соседей объекта u имеют метку этого класса.
3. Присвой объекту u тот класс, каких соседей большинство

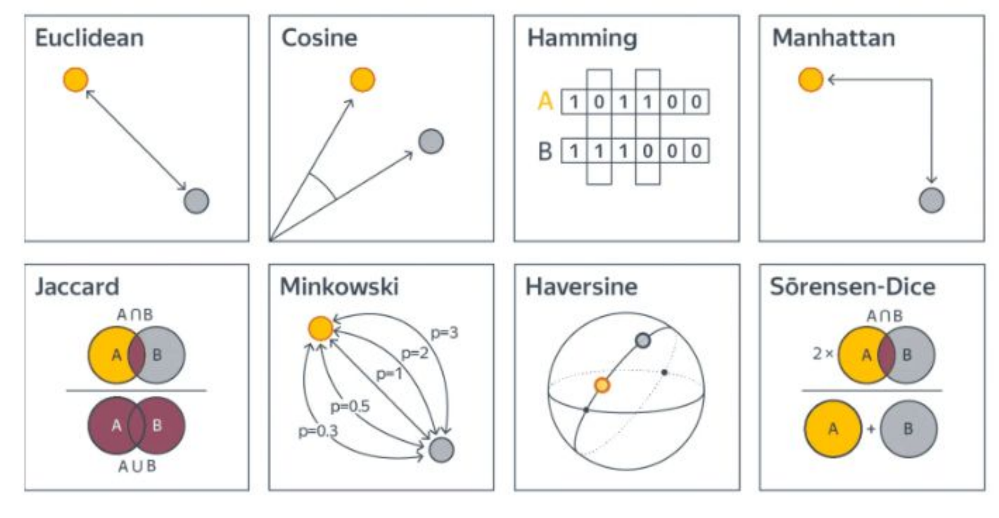
$$a(u) = argmax_{y \in Y} \sum_{i=1}^{k} ||(y_u^{(i)} = y)$$

$$P(u \sim y) = \frac{\sum_{i=1}^{k} ||(y_u^{(i)} = y)}{k}$$

А оценить вероятность?

1. Выполняем шаги 1-2 из прошлого пункта
2. Каждый полученный результат делим на наше k.

Метрики в KNN



База это Евклид. Он будет работать отлично

ρ(x,y) = √(∑i(xi-yi)^2)

Манхеттенское:
ρ(x,y) = ∑i |xi-yi|

Минковский
ρ(x,y) = (∑i |xi-yi| ^ p) ^(1/p)

Косинусное
ρ(x,y) = 1- cos(θ) = 1 - (xy)/||x||*||y||

Делаем KNN круче!

$$a(u) = argmax_{y \in Y} \sum_{i=1}^{k} w_i || (y_u^{(i)} = y)$$

$$w_i = \frac{k+1-i}{k} \qquad w_i = q^i, 0 < q < 1$$

$$a(u) = argmax_{y \in Y} \sum_{i=1}^{k} K\left(\frac{p(u,x_u^i)}{h}\right) || (y_u^{(i)} = y)$$

1. Добавляем веса. Вес тем больше, чем ближе объект к цели.
2. Чаще всего у нас затухающие веса берутся линейно или экспоненциально.

А можем еще круче? Да. На ядрах.

Какие бывают ядра?

- $K(x) = \frac{1}{2}\mathbb{I}\left[|x| \leqslant 1\right]$ — прямоугольное ядро;
- $K(x) = (1 - |x|)\,\mathbb{I}\left[|x| \leqslant 1\right]$ — треугольное ядро (непрерывное);
- $K(x) = \frac{3}{4}\left(1 - x^2\right)\mathbb{I}\left[|x| \leqslant 1\right]$ — ядро Епанечникова (гладкое везде, кроме −1 и 1);
- $K(x) = \frac{15}{16}\left(1 - x^2\right)^2\mathbb{I}\left[|x| \leqslant 1\right]$ — биквадратное ядро (гладкое везде);
- $K(x) = \frac{1}{\sqrt{2\pi}}e^{-2x^2}$ — гауссовское ядро (бесконечно гладкое везде).

От ядра зависит гладкость аппроксимации.
Чем ниже, тем более гладкое.

На практике, нужно либо прямоугольное (быстро и просто), либо гауссовское (супер гладко)

h, ширина окна, как и learning rate подбирается от задачи к задаче

Не хочу делить котов. Хочу квартиру!

ИТМО

Логичные варианты

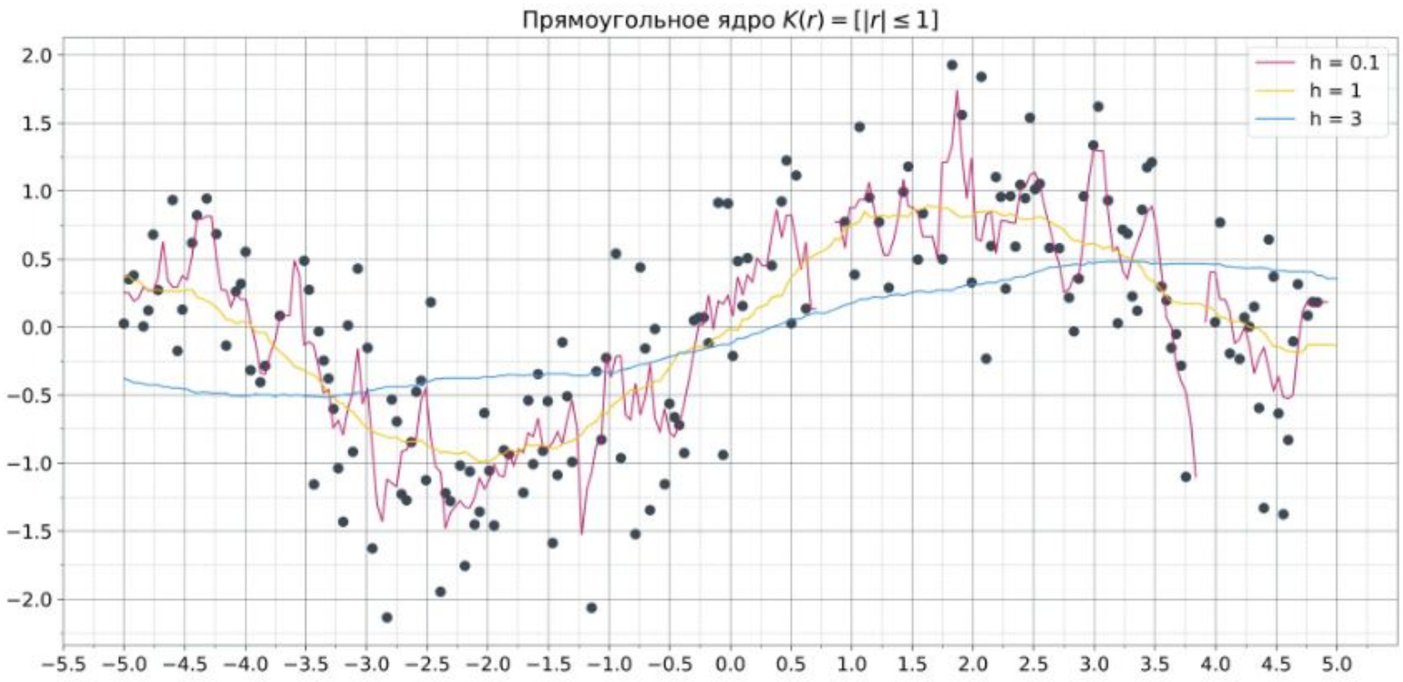
$$a(u) = 1/k * \sum_{i=1}^{k} y_u^{(i)}$$

$$a(u) = (\sum_{i=1}^{k} K(\frac{p(u,x_u^i)}{h}) y_u^{(i)}) / (\sum_{i=1}^{k} K(\frac{p(u,x_u^i)}{h}))$$

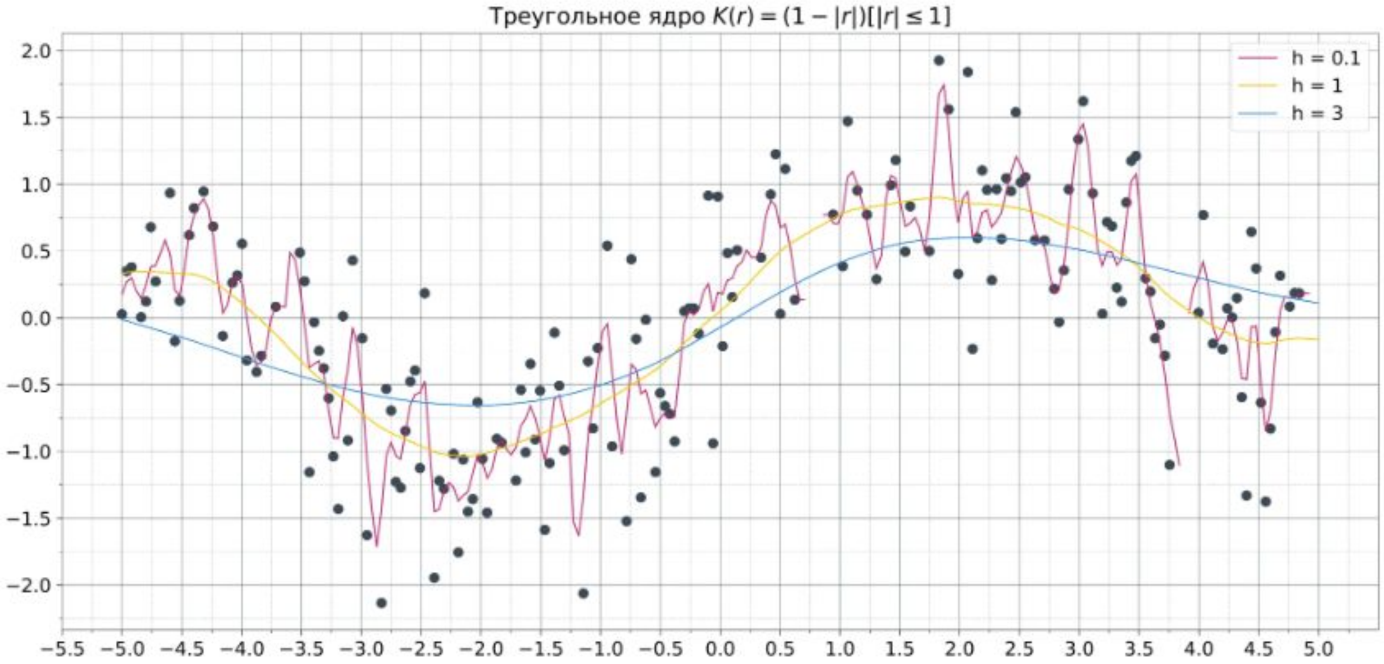
Вариант спустя пару логичных замечаний

$$a(u) = argmin_{y \in R} \sum_{i=1}^{k} K(\frac{p(u,x_u^i)}{h}) * (y - y_u^{(i)})^2$$

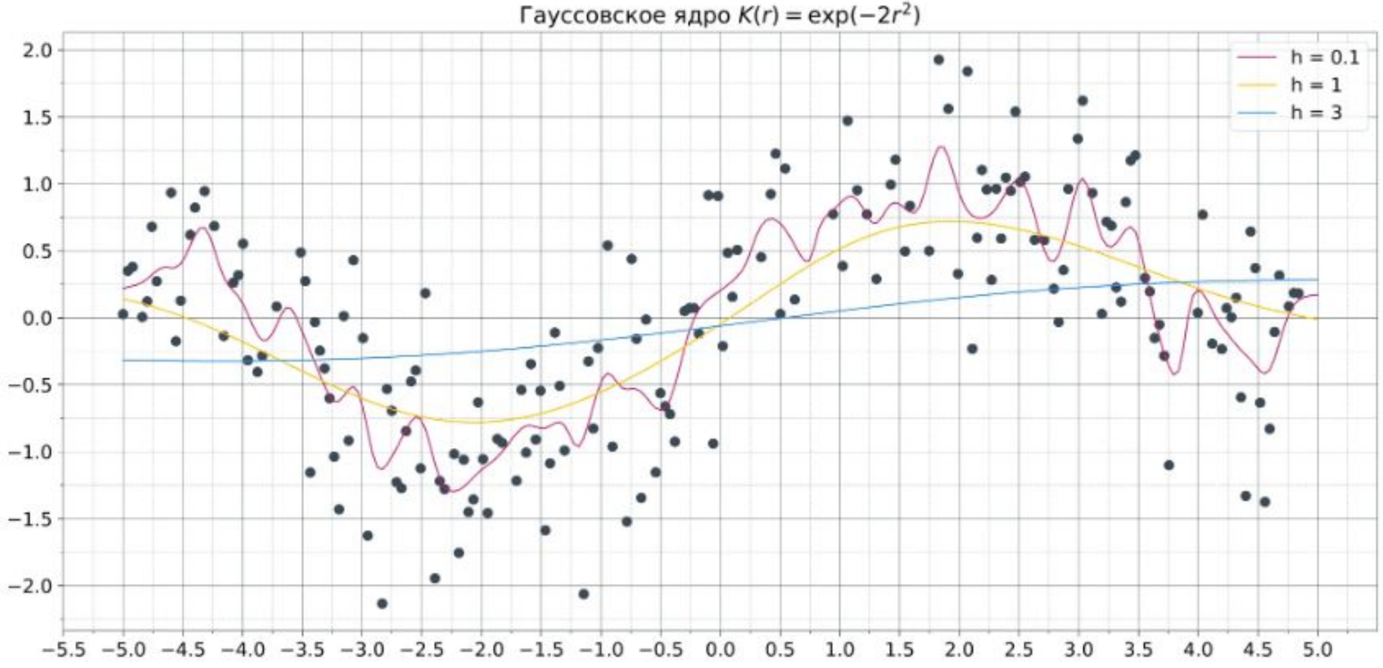
Разная ширина окна



Разная ширина окна



Разная ширина окна



Круче KNN (с оговоркой)

Деревья

Не ML:

- K-d деревья
- Random projection trees

ML:

- “Леса”

Хэши

- LSH
- HNSW

Спасибо за внимание!

ITsMOre than a UNIVERSITY