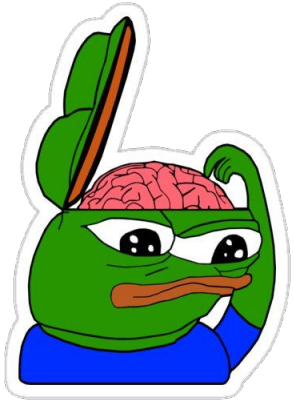
Explainable artificial intelligence

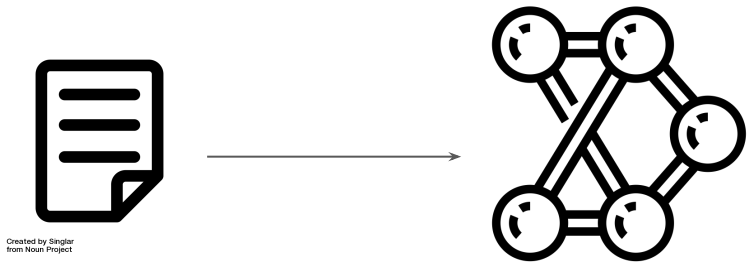


НИС
Алексей Космачев
2021

Что это и зачем?

Created by Singlar
from Noun Project

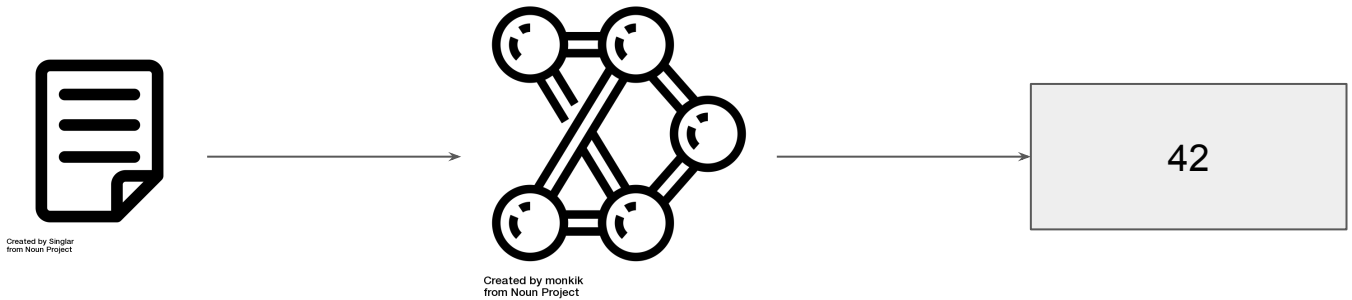
Что это и зачем?



Created by Singlar
from Noun Project

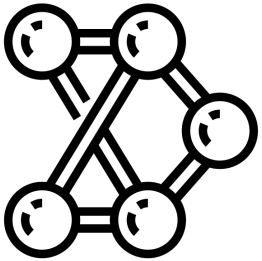
Created by monkik
from Noun Project

Что это и зачем?



Что это и зачем?

Created by Singlar
from Noun Project



Created by monkik
from Noun Project

42

Почему?

Когда это важно?

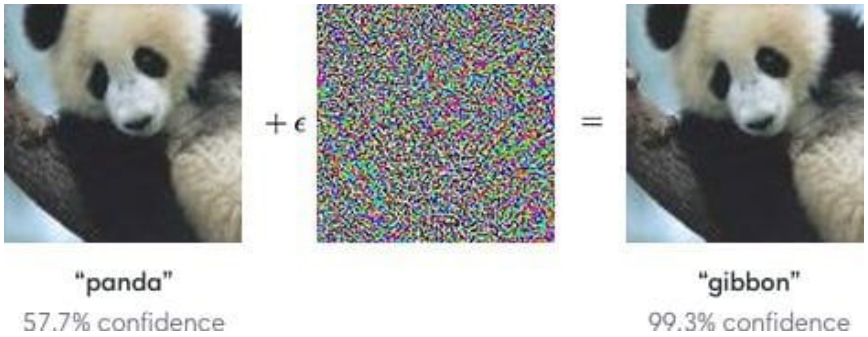
Когда речь идет о человеческой жизни

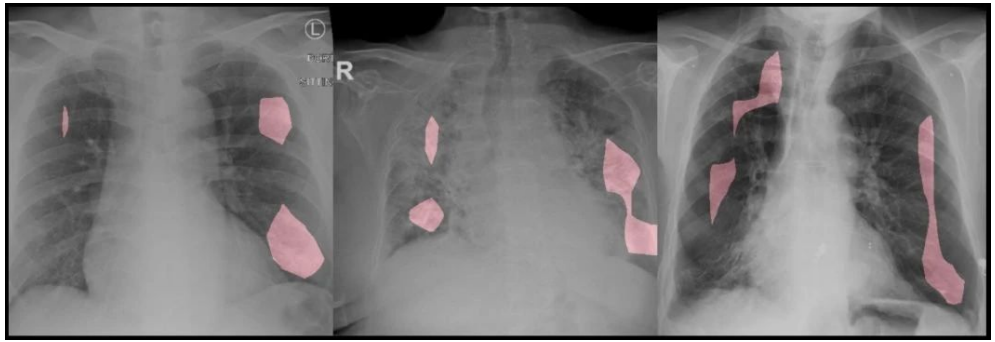
Машина всегда лишь отображает информацию, которую в нее загрузили люди

Диагноз должен быть поставлен специалистом из мяса

Когда это важно?

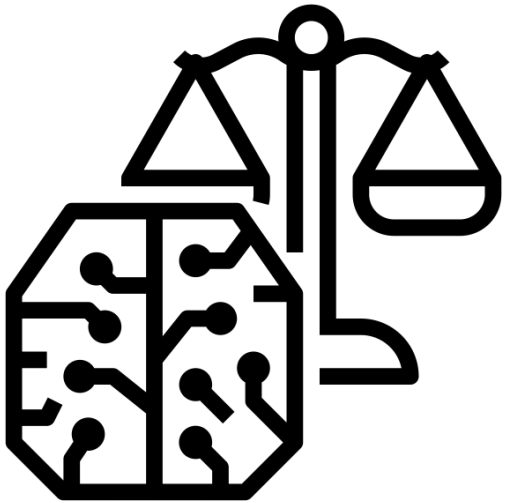
В медицине хочется получить основные признаки болезни, а не сам диагноз





Когда это важно?

Когда мы находимся в правовом поле



Created by monkik
from Noun Project

Когда это важно?

Итоговое решение по правовому делу должен принимать человек

Когда это важно?

В Европейском регламенте по защите данных GDPR даже зафиксирован такой пункт как “**Right to explanation**”.

Область применения захватывает банковские системы, здравоохранение и многие другие

Когда это важно?

Когда наша основная задача - это понять что-то про данные

Например мы можем хотеть разобраться, почему возникает та или иная ошибка

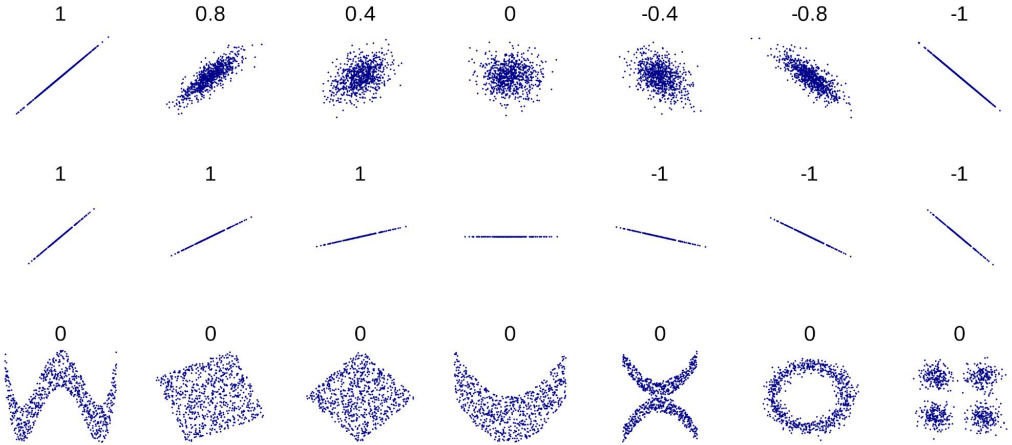
Mastercard	...	DO NOT HONOR
Visa	...	DO NOT HONOR
Visa	...	SUCCESS
Maestro	...	DO NOT HONOR
Mastercard	...	DO NOT HONOR
Visa	...	NO MONEY
UnionPay	...	DO NOT HONOR
Visa	….	SUCCESS
Mir	...	DO NOT HONOR

Как же нам делать интерпретируемые алгоритмы?

Как же нам делать интерпретируемые алгоритмы?

1. Не обучать машины вовсе!

Старая добрая математическая статистика



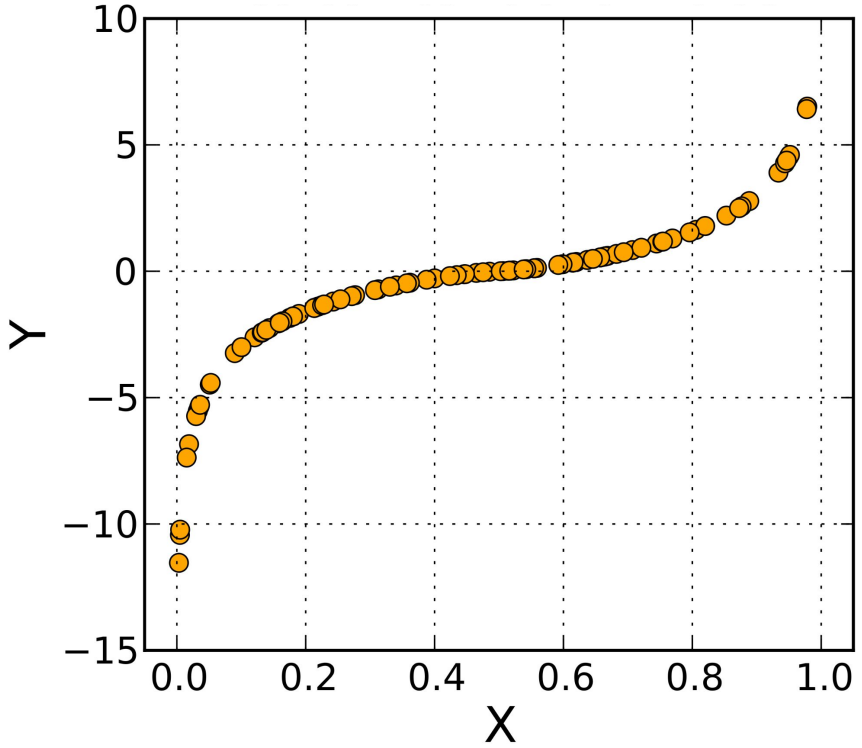
$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Старая добрая математическая статистика

Корреляция пирсона ловит только линейные зависимости

Что делать с нелинейными?

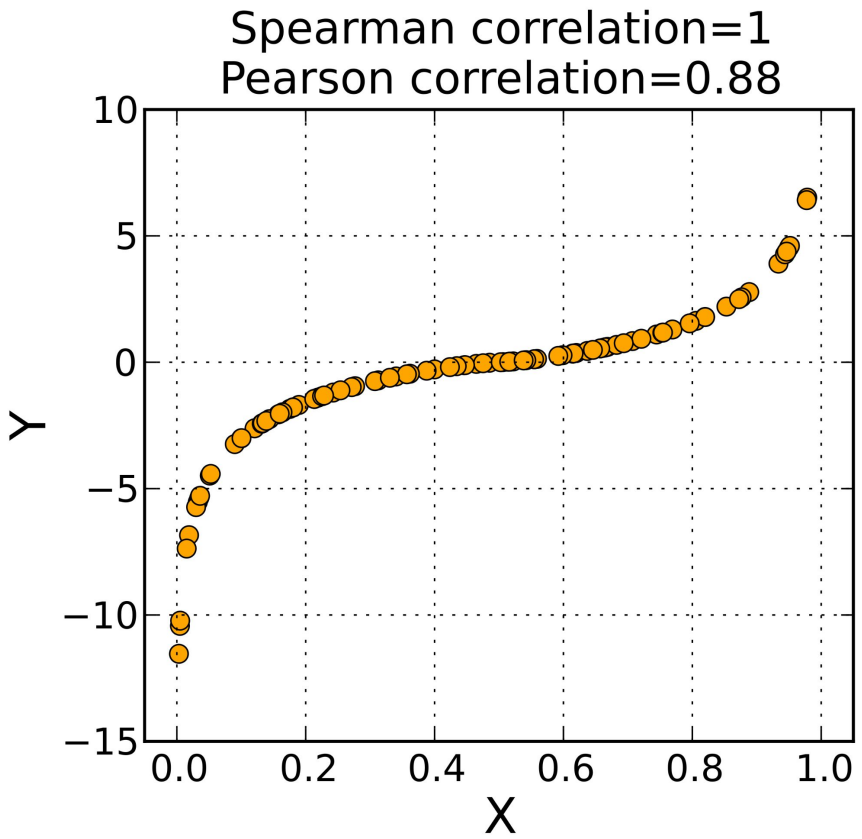


Старая добрая математическая статистика

Корреляция пирсона ловит только линейные зависимости

Что делать с нелинейными?

$$r_s = \rho_{\mathrm{rg}_X, \mathrm{rg}_Y} = \frac{\mathrm{cov}(\mathrm{rg}_X, \mathrm{rg}_Y)}{\sigma_{\mathrm{rg}_X} \sigma_{\mathrm{rg}_Y}},$$



Как же нам делать интерпретируемые алгоритмы?

1. Не обучать машины вовсе!
2. Обучать простые модели

Линейные модели

$$a(x) = w0 + w1 * x1 + w2 * x2 + \ldots$$

Линейные модели

$$a(x) = w0 + w1 * x1 + w2 * x2 + \ldots$$

y - цена квартиры

x1 - количество квадратных метров

x2 - находится ли квартира внутри МКАДа

Линейные модели

$$a(x) = w0 + w1 * x1 + w2 * x2 + \ldots$$

y - цена квартиры

x1 - количество квадратных метров

x2 - находится ли квартира внутри МКАДа

w1 - стоимость одного квадратного метра

w2 - переплата за нахождение внутри МКАДа

Линейные модели

$$a(x) = \text{sign}(w_0 + w_1 * x_1 + w_2 * x_2 + ...)$$

y - выживет ли пассажир

x1 - пассажир мужчина?

x2 - стоимость билета

Линейные модели

$$a(x) = \textbf{sigmoid}(w0 + w1 * x1 + w2 * x2 + ...)$$

y - выживет ли пассажир

x1 - пассажир мужчина?

x2 - стоимость билета

Линейные модели

$$a(x) = \textbf{sigmoid}(w0 + w1 * x1 + w2 * x2 + ...)$$

y - выживет ли пассажир

x1 - пассажир мужчина?

x2 - стоимость билета

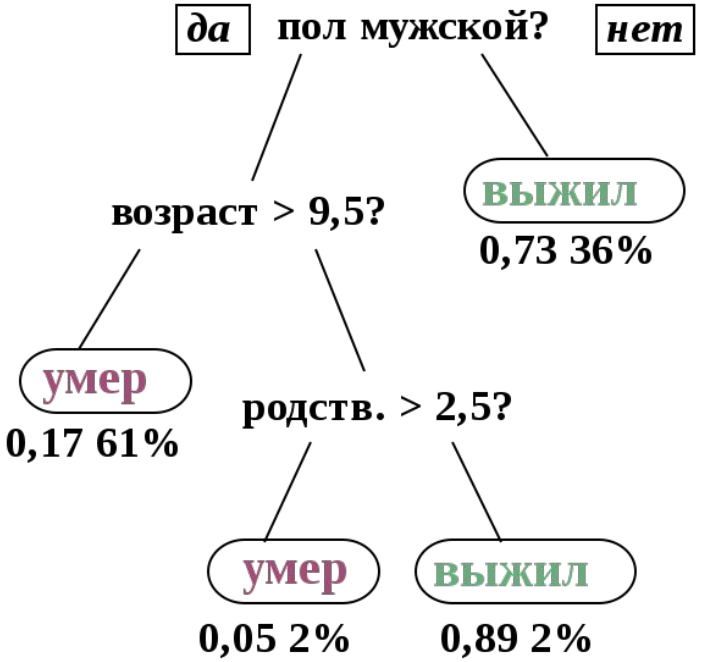
w1 - как сильно пол влияет на вероятность спастись

w2 - как сильно цена влияет на вероятность спастись

Как же нам делать интерпретируемые алгоритмы?

1. Не обучать машины вовсе!
2. Обучать простые модели
3. Обучать деревья

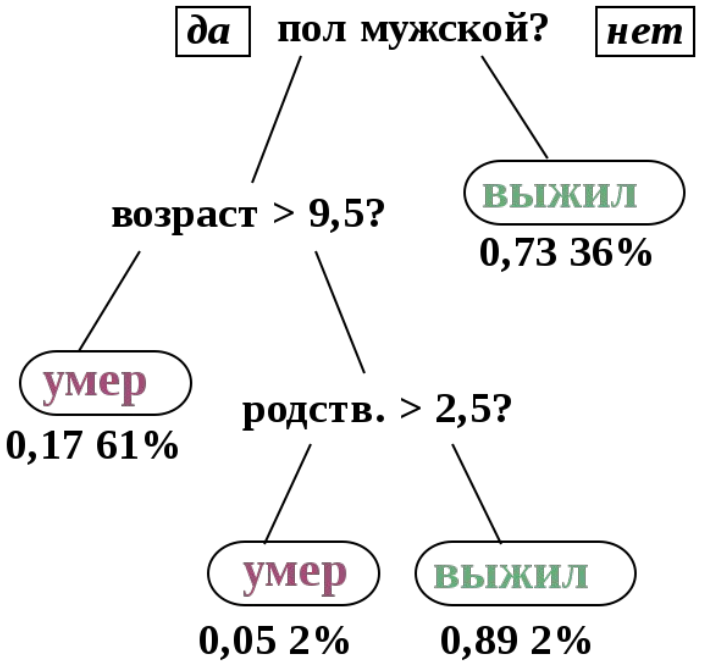
Дерево решений



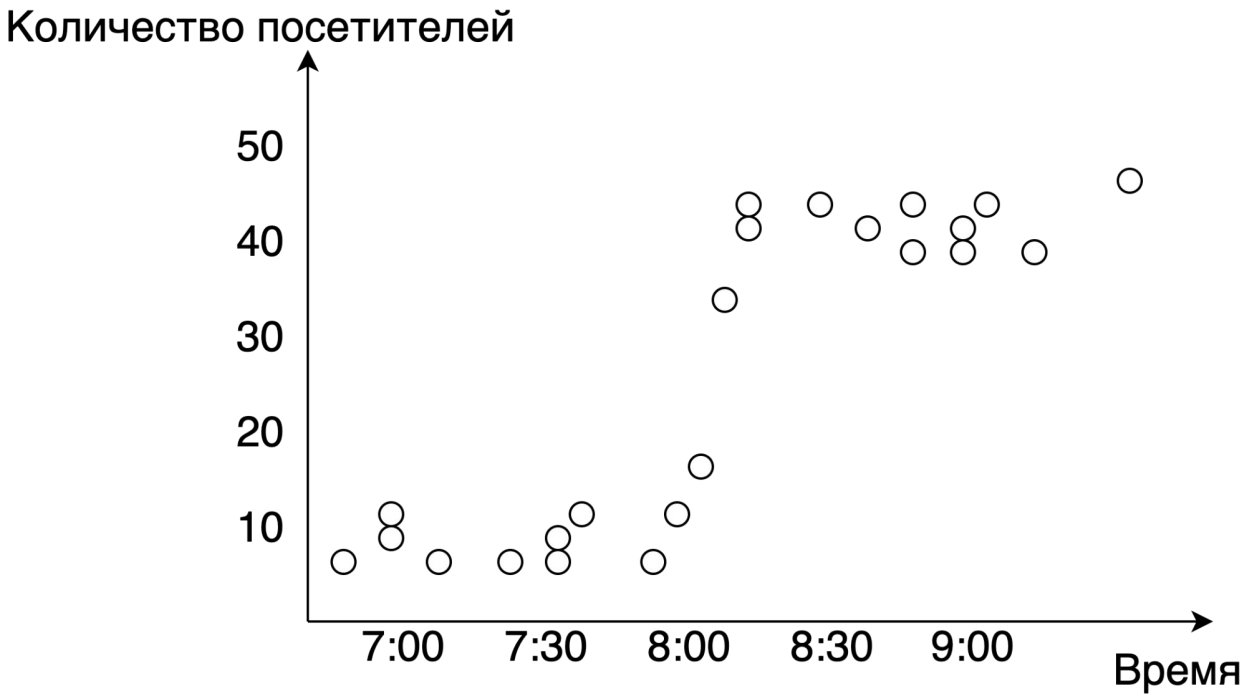
Дерево решений

Дает четкую структуру принимаемых решений

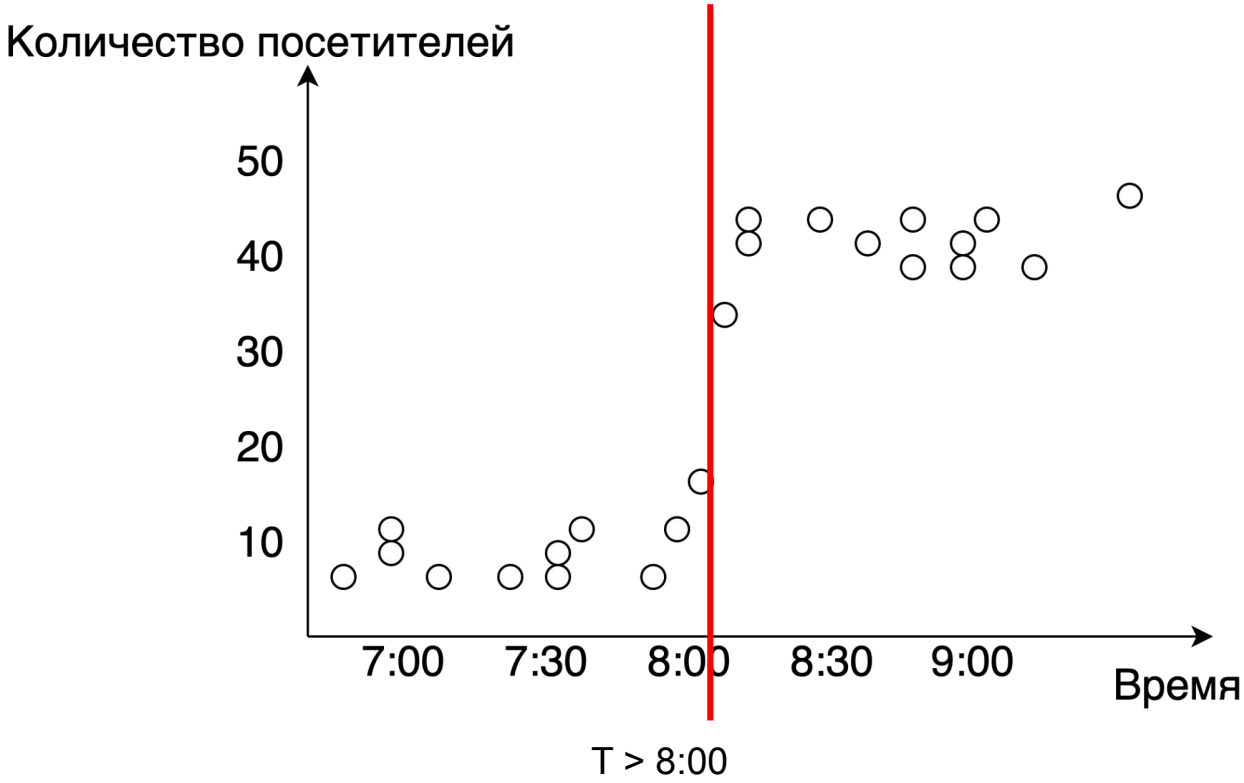
При этом умеют находить сложные зависимости



Дерево решений



Дерево решений



Как же нам делать интерпретируемые алгоритмы?

1. Не обучать машины вовсе!
2. Обучать простые модели
3. Обучать деревья
4. Обучать все что угодно и раскладывать по Шепли

Вектор Шепли

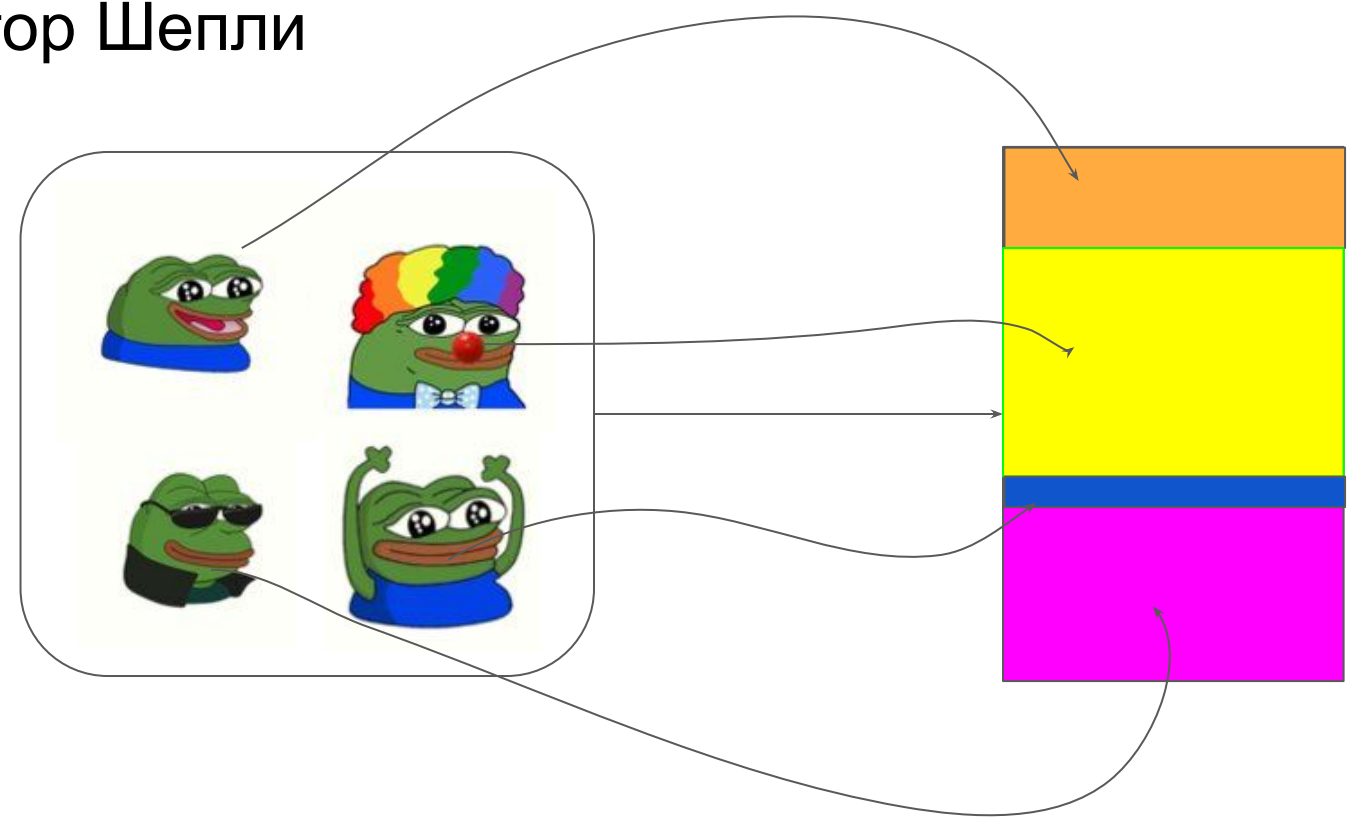
Вы с друзьями участвуете в командном хакатоне

После 3 бессонных ночей вы выигрываете X денег

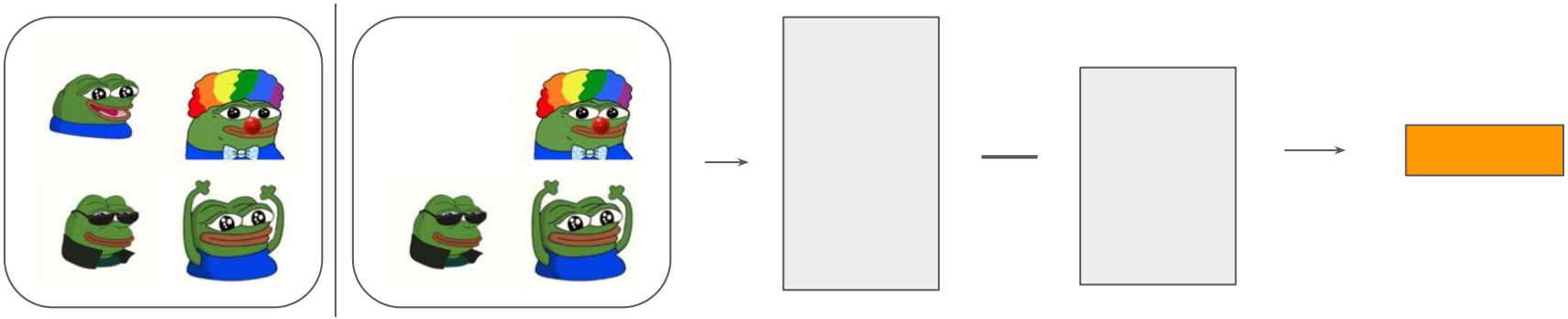
Вопрос - как *честно* поделить их между вами?

Вектор Шепли

Вектор Шепли

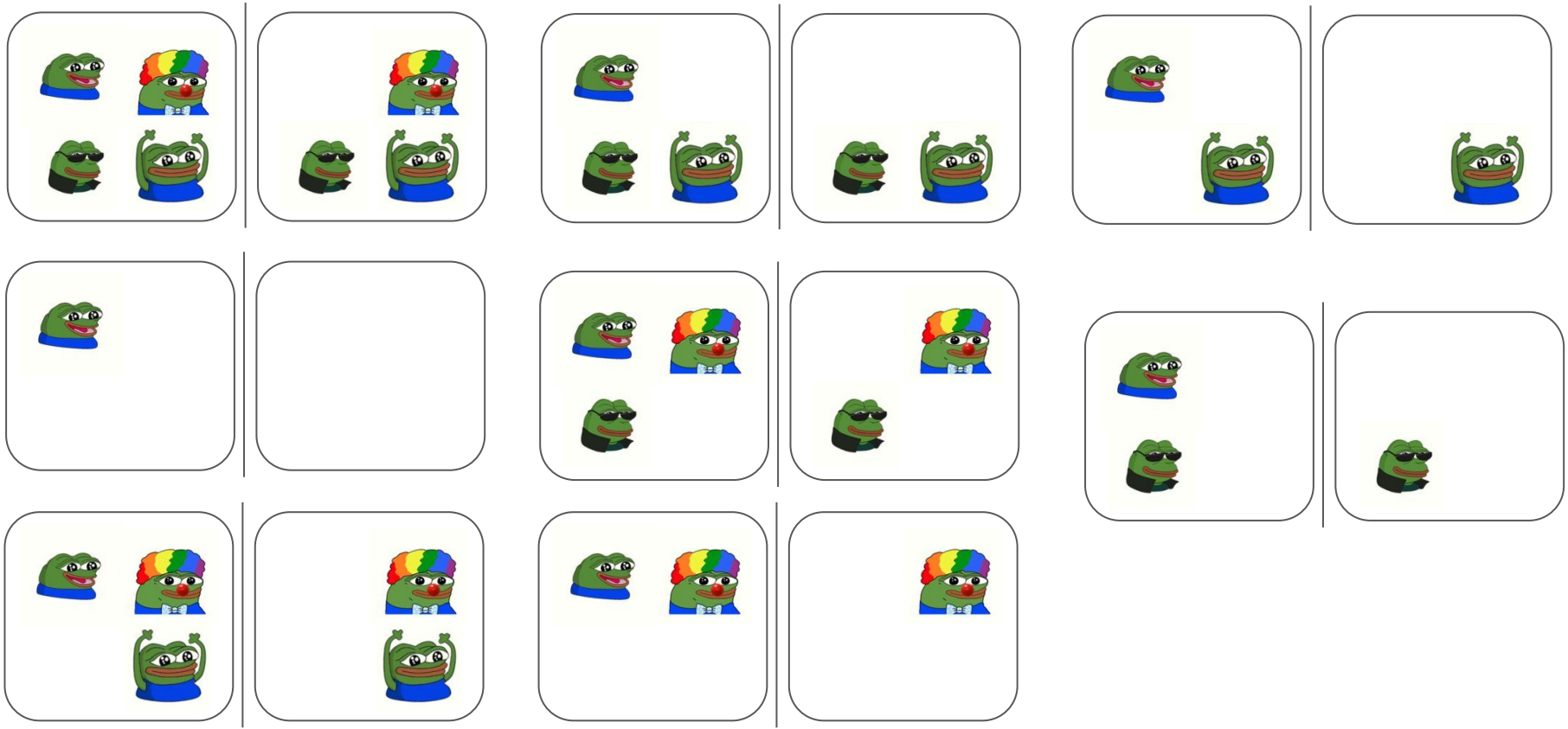


Вектор Шепли

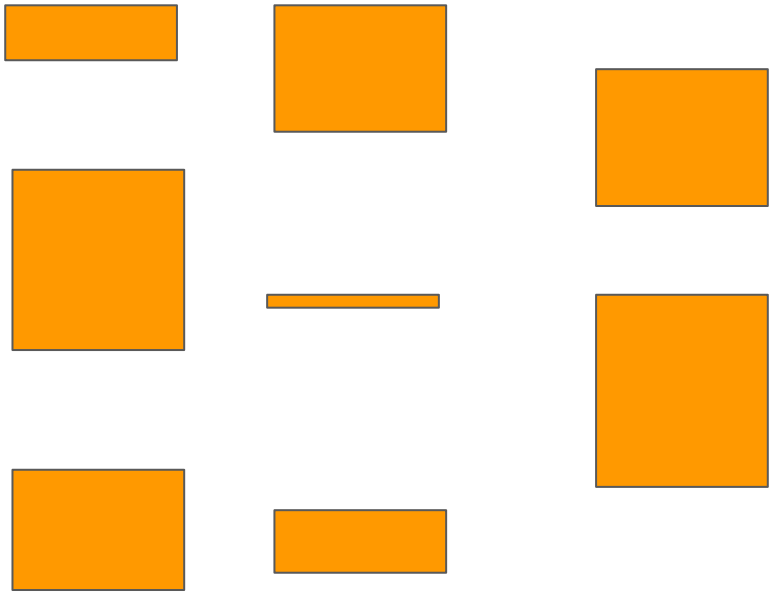


Вклад первого игрока в группу

Вектор Шепли

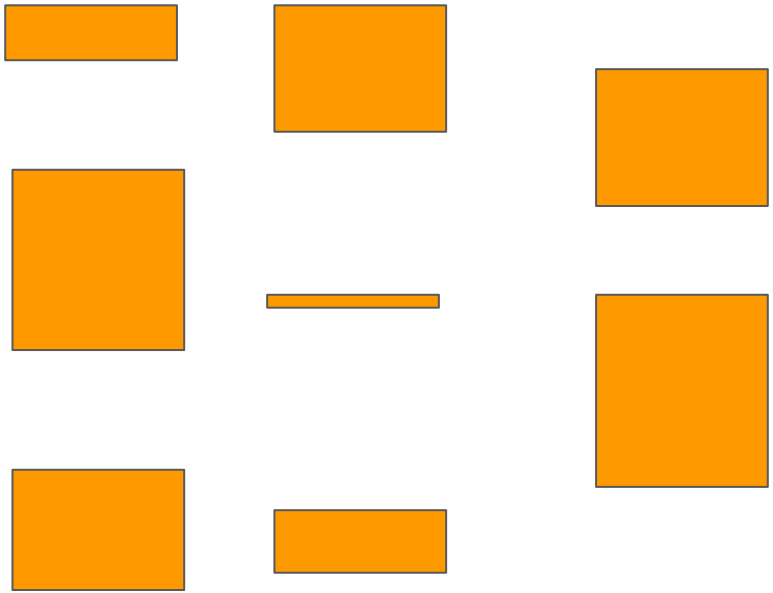


Вектор Шепли



Вклады первого игрока в каждую из возможных групп

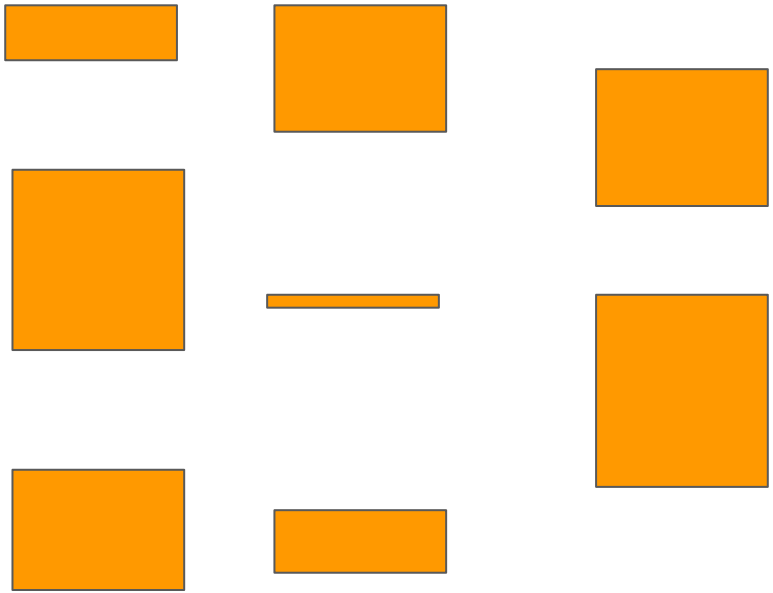
Вектор Шепли



Вклады первого игрока в каждую из возможных групп

Значение Шепли для первого игрока - это **среднее этих вкладов**

Вектор Шепли



Вклады первого игрока в каждую из возможных групп

Значение Шепли для первого игрока - это **среднее этих вкладов**

Теорема Шепли -
сумма Шепли каждого игрока равна итоговому суммарному выигрышу + еще несколько полезных свойств (которые нам сейчас не интересны)

Вектор Шепли

Признаки модели - игроки

Результат работы модели - итоговый выигрыш

Соответственно значение Шепли для конкретного признака - это **вклад этого конкретного признака** в итоговый результат