

Неделя науки 2022

ЭНТРОПИЯ В МАТЕМАТИКЕ, ФИЗИКЕ И ИНФОРМАТИКЕ

Выполнил:
Гирфанов Вячеслав (гр. 3530903/10003)

Научный руководитель:
Доц. каф. "Высшая математика"
Филимоненкова Надежда Викторовна

ЭТИМОЛОГИЯ

Происходит от древне-греческого ἐντροπία «увёртка, уловка», из ἐν «в» + τρόπος «оборот, поворот; характер».

S

$$S = - \sum_{k=0}^n p_i \log_2 p_i$$

$$S = k \ln Z$$

Впервые введено в 1865 году Рудольфом Клаузиусом применительно к физике.

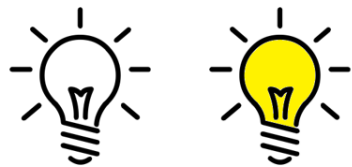
Статистический смысл установлен Людвигом Больцманом.

В 1948г. использовано Клодом Шенноном в статье «математическая теория связи».

$$S = \frac{1}{n} \log_2 \frac{\sum_{i=\rho}^n n}{n_1 n_2 n_3 \dots}$$

Абстрактная система лампочек

1 лампочка



2 состояния

Число состояний системы Ω в зависимости от числа лампочек n :

$$\Omega = 2^n$$

$$n = \log_2 \Omega$$

Это и называется энтропией

2 лампочки



4 состояния

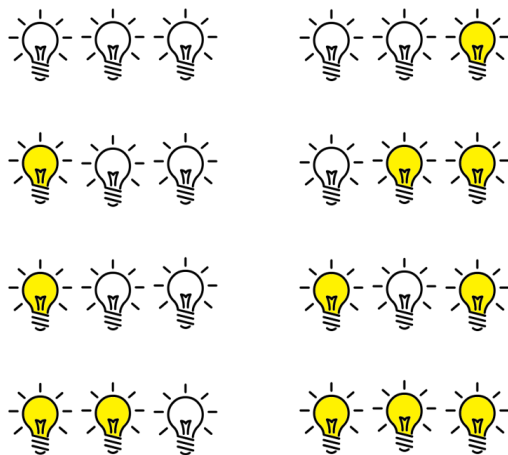
0 лампочек могут находиться только в одном состоянии:

$$\log 1 = 0$$

Линейная природа операций над энтропией:

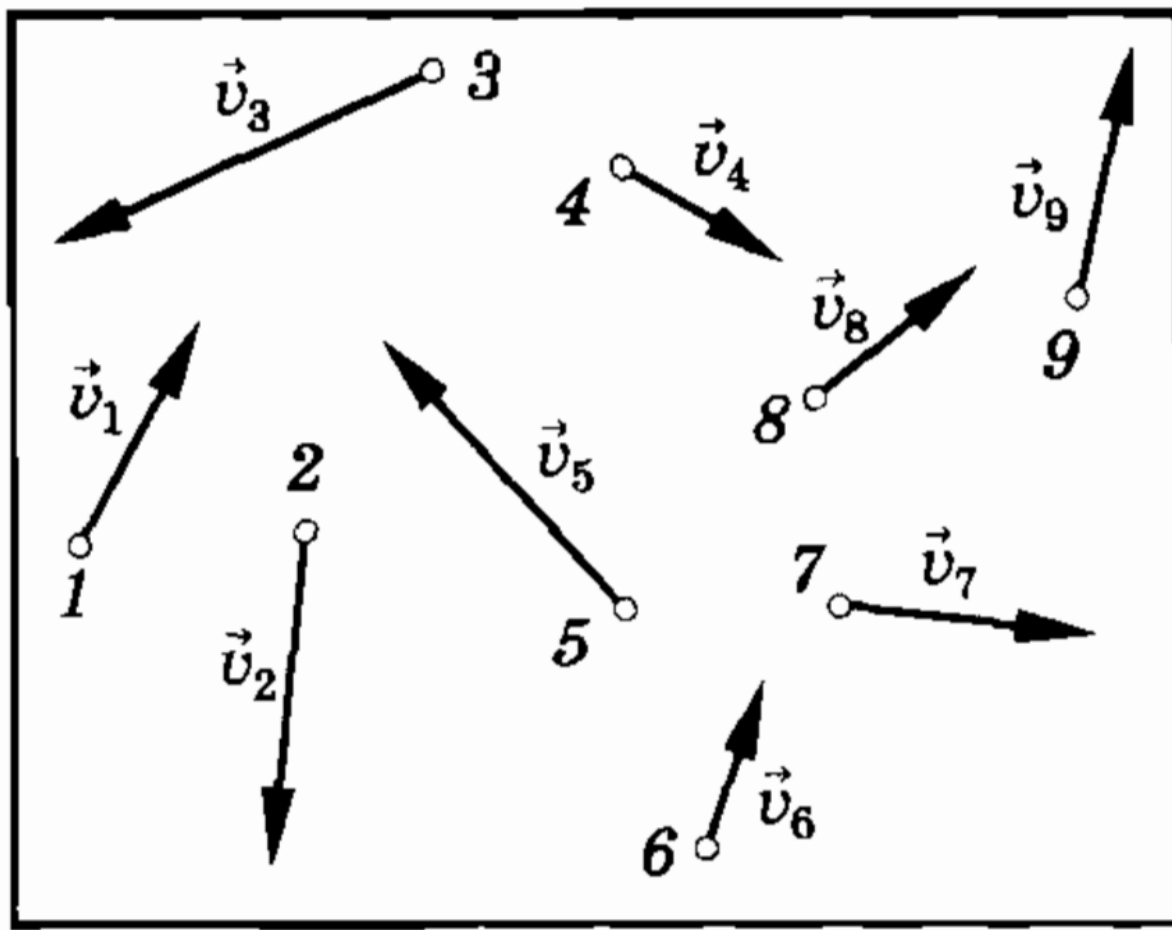
$$n_1 + n_2 = \log_2 \Omega_1 + \log_2 \Omega_2 = \log_2 \Omega_1 \Omega_2$$

3 лампочки



8 состояний

Энтропия в физике



Искали это:

$$S = \int \frac{\delta Q}{T}$$

А нашли вот это:

$$S = k \ln \Omega$$

Энтропия в информатике

1. Информация субъективна

2. Информация имеет формы выражения

2 подхода к измерению информации

Субъективный

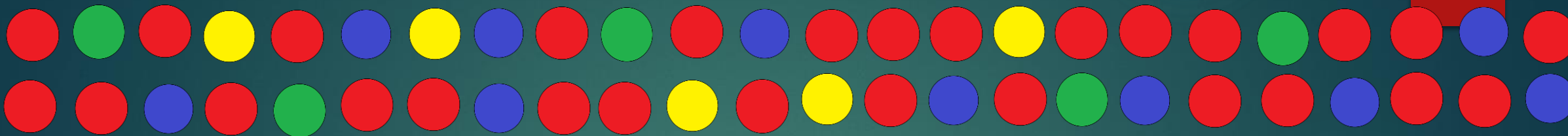
Алфавитный



Формула Хартли: $k = \log_2 N$

Количество информации в битах называется двоичной энтропией

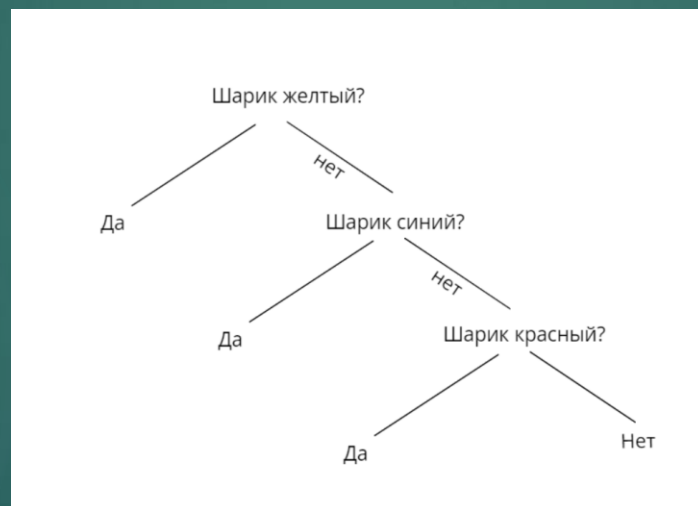
Формула Шеннона



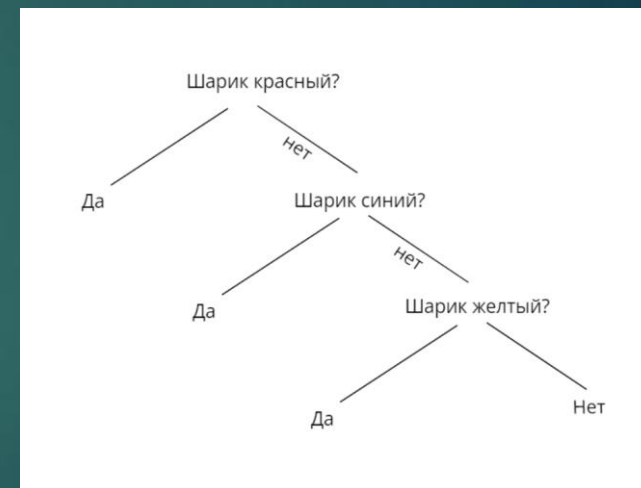
Наивный способ узнать цвет шарика:

Шарик зеленый?
Шарик желтый?
Шарик синий?
Шарик красный?

Неоптимальный способ узнать цвет шарика:



Оптимальный способ узнать цвет шарика:



Формула Шеннона:

$$i = \sum p_i \log_2 \frac{1}{p_i}$$

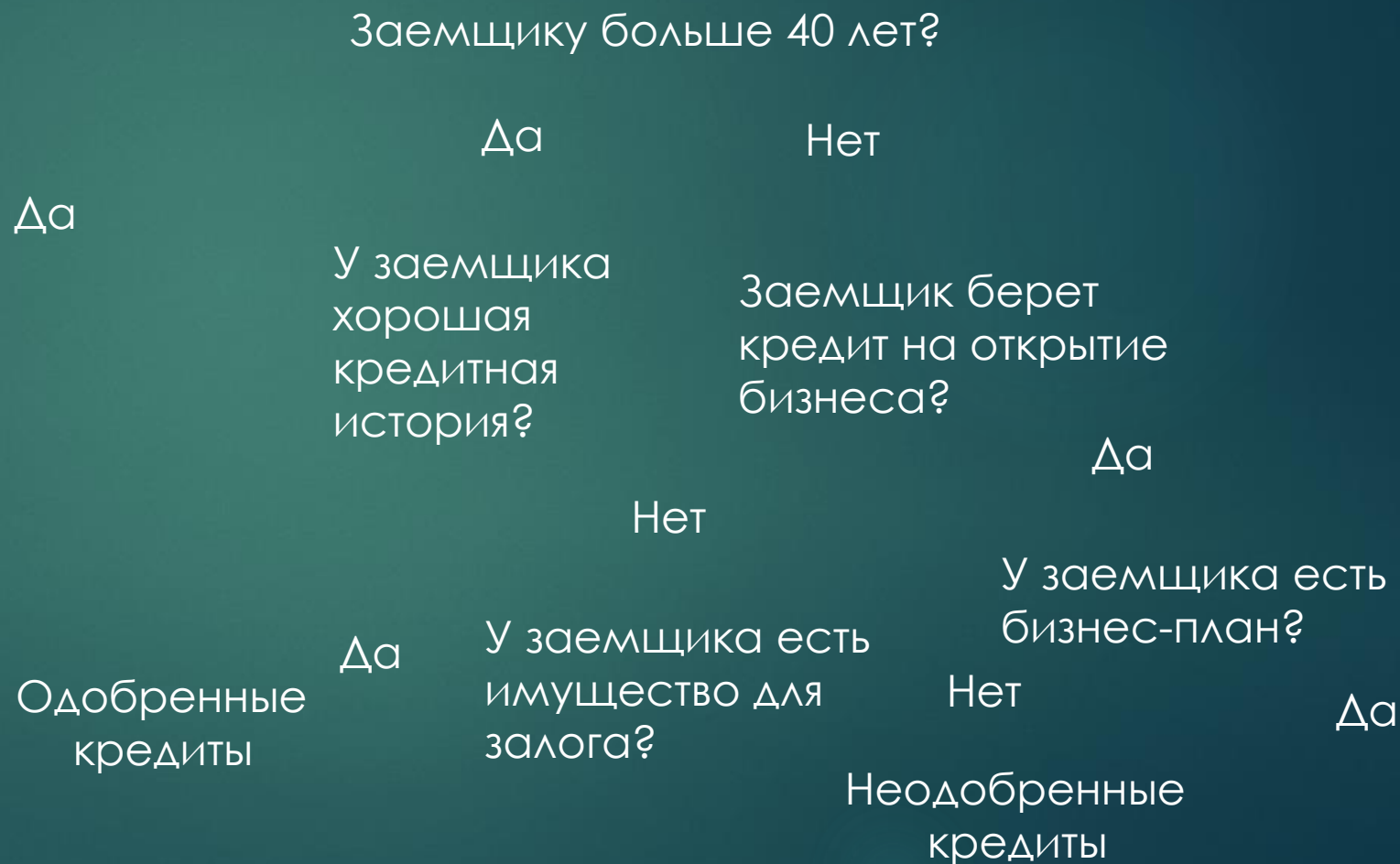
Описание моделируемой ситуации

Заемщик берет кредит в банке. По его анкете принимается решение, выдавать ли ему кредит.

Алгоритм классификации посредством последовательной проверки простых условий называется деревом решений

Классы:

- 1) Одобренные кредиты
- 2) Неодобренные кредиты




```
import math
import random

#Создание обучающей выборки:

Creditors_Names = ["Вася", "Петя", "Ваня", "Илья", "Сергей", "Павел", "Егор", "Коля", "Васиссуалий"]

Ammount = [] #Список для хранения числа кредиторов с параметрами:
Ammount += [50] #50 человек старше 40 лет
Ammount += [30] #30 человек берут кредит на открытие бизнеса
Ammount += [20] #Из них 20 имеют бизнес-план
Ammount += [80] #У 80 человек хорошая кредитная история
Ammount += [60] #60 человек обладают имуществом для залога
Ammount += [25] #У 25 человек имущество застраховано

#Создание выборки из 200 человек с заданным числом обладателей параметров в виде массива с названием Learning_Kit:
Learning_Kit = [[random.choice(Creditors_Names),0,0,0,0,0,0] for k in range(0, 200)]
Counter = Ammount #Переменная-счетчик людей с параметрами

#Заполнение выборки в соответствии с числом кредиторов:

def Filing(Parameter, Condition=0):#функция заполнения
    cond = 1
    while Counter[Parameter-1]:
        k = random.randint(0, 199)
        if Condition: cond = Learning_Kit[k][Condition]
        if not Learning_Kit[k][Parameter] and cond:
            Learning_Kit[k][Parameter] = 1
            Counter[Parameter-1] -= 1

#Вызов заполнения для всех параметров
Filing(1)
Filing(2)
Filing(3, 2)
Filing(4)
Filing(5)
Filing(6)
```


#Построение оптимального дерева решений:

Predicate = [0, 0, 0, 0, 0, 0] #Массив для хранения найденных предикатов по порядку ветвления в дереве

#Наивный алгоритм:

```
def Naive(Creditor):  
    if not Creditor[1]:  
        if Creditor[2]:  
            if Creditor[3]: return True  
        if Creditor[4]: return True  
        if Creditor[5] or Creditor[6]: return True  
    return False
```

#Расчёт комбинаторной энтропии на основе данных наивного алгоритма:

```
def Entropy(a): #a - предоставляемый функции массив выборки  
    N = 0  
    for k in range(len(a)):  
        if Naive(a[k]): N +=1  
    if len(a) == 0: return 0  
    if N/len(a) == 1: return 0  
    return -( (N/len(a))*math.log(N/len(a),2) + (1-N/len(a))*math.log(1-N/len(a), 2) )
```

PositiveAnswer = [] #Множество одобренных кредитов

NegativeAnswer = [] #Множество кредитов с отказом

Predicates = [] #Массив для хранения номеров предикатов (параметров, по которым делим множество кредиторов)

#Первое деление по параметру:

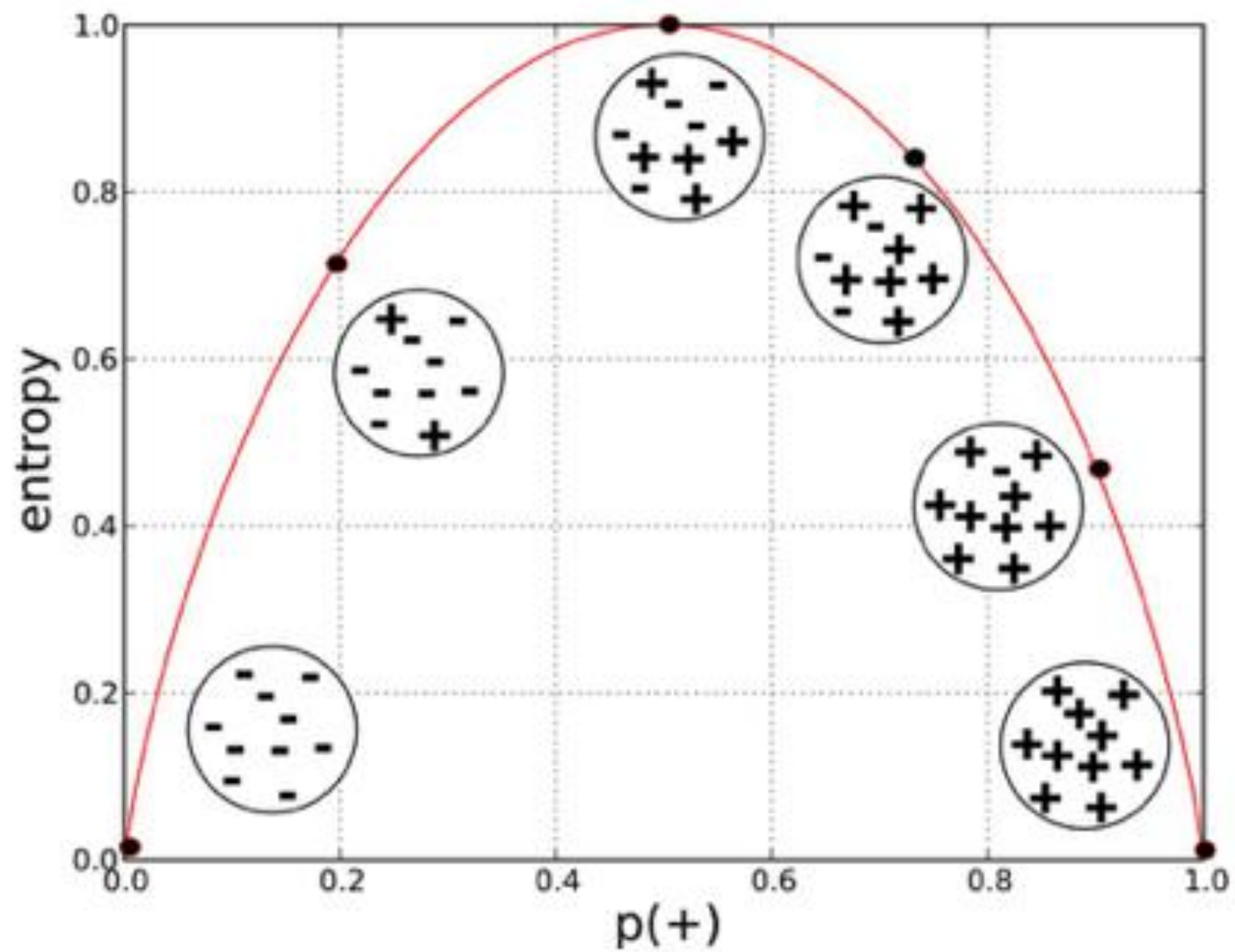
OriginalEntropy = Entropy(Learning_Kit)

```
for k in range(0, 200):  
    if Learning_Kit[k][1]:  
        PositiveAnswer += [Learning_Kit[k]]  
    else:  
        NegativeAnswer += [Learning_Kit[k]]
```

MaxEntropy = OriginalEntropy - (Entropy(PositiveAnswer) + Entropy(NegativeAnswer)) #Максимальная энтропия

pred = [1, 2, 3, 4, 5, 6] #Список номеров предикатов

Принцип поиска наибольшей убыли энтропии при делении по классам



Принцип Ландауэра

В любой вычислительной системе, независимо от её физической реализации, при потере 1 бита информации выделяется теплота в количестве по крайней мере W джоулей:

$$W = k_B T \ln 2,$$

Связывает количество информации
с её воплощением в реальном
мире

НО

Принцип физический, зависит от архитектуры хранения информации и основан на следующем положении:

"A simple binary device consists of a particle in a bistable potential well"