

Муниципальный этап ВсОШ по информатике профиль ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

Республика Коми 2025

Разбор заданий

Оглавление

7-8	2
Соревнование	2
Раскрась точку!	3
Выдача кредита	3
Размер окна	5
Ансамбль	5
Динамика акций	6
9-11	7
Кредитный риск	7
По грибы	8
Анализ данных	9
Выбор порога	9
Продажи мороженого	11
Лаборатория	11

7-8

Соревнование

Обозначим:

- LB — баллы LenaBagging
- MB — баллы MishaBoosting
- AS — баллы AlexStacking
- KV — баллы KateVoting

Из условий:

1. $LB = MB$
2. $AS = KV + 4$
3. $LB + MB = 3(AS + KV)$
4. $LB + MB + AS + KV < 70$
5. $KV \geq 3$
6. KV — минимальное возможное

Шаг 1: Из $LB = MB$ следует $LB + MB = 2LB$

Шаг 2: Из $AS = KV + 4$ следует $AS + KV = 2KV + 4$

Шаг 3: Из условия 3: $2LB = 3(2KV + 4) \rightarrow 2LB = 6KV + 12 \rightarrow LB = 3KV + 6$

Шаг 4: Подставляем в неравенство суммы:

$$2LB + (2KV + 4) < 70$$

$$2(3KV + 6) + 2KV + 4 < 70$$

$$6KV + 12 + 2KV + 4 < 70$$

$$8KV + 16 < 70$$

$$8KV < 54$$

$$KV < 6.75$$

Шаг 5: Из условия 5: $KV \geq 3$, значит $KV = 3, 4, 5, 6$

Шаг 6: Из условия 6 выбираем наименьшее $KV = 3$

Шаг 7: Находим LB :

- $KV = 3: LB = 3 \times 3 + 6 = 15$
- Проверяем сумму: $15 + 15 + 7 + 3 = 40 < 70 \checkmark$

Ответ: LenaBagging набрала 15 баллов.

Раскрась точку!

Точка	Истинный цвет	Ближайшая точка	Цвет ближайшей	Прогноз	Результат
A	красный	D	зелёный	зелёный	Ошибка
B	зелёный	E	зелёный	зелёный	Верно
C	синий	F	синий	синий	Верно
D	зелёный	G	красный	красный	Ошибка
E	зелёный	B	зелёный	зелёный	Верно
F	синий	C	синий	синий	Верно
G	красный	D	зелёный	зелёный	Ошибка
H	зелёный	I	синий	синий	Ошибка
I	синий	H	зелёный	зелёный	Ошибка
J	красный	H	зелёный	зелёный	Ошибка

Таким образом, точность предсказаний составляет 40%.

Выдача кредита

Получим предсказания для всех клиентов на дереве решений. На 5, 7 объектах дерево выдает ошибочный прогноз, таким образом, точность составляет 75%.

Клиент	Доход	Недвижимость	История	Кредит	Предсказание
1	1	1	1	1	1
2	1	0	1	1	1

Клиент	Доход	Недвижимость	История	Кредит	Предсказание
3	0	1	1	1	1
4	1	1	0	1	1
5	1	0	0	0	1
6	0	1	0	0	0
7	0	0	1	1	0
8	0	0	0	0	0

Таким образом, точность предсказаний составит $6/8 = 0,75$, что в процентах составляет 75%.

Размер окна

1. Формула количества окон

$$W = \left(\left\lfloor \frac{n-k}{3} \right\rfloor + 1 \right)^2$$

$$121 = \left(\left\lfloor \frac{50-k}{3} \right\rfloor + 1 \right)^2$$

2. Извлечение квадратного корня

$$\left\lfloor \frac{50-k}{3} \right\rfloor + 1 = 11$$

$$\left\lfloor \frac{50-k}{3} \right\rfloor = 10$$

3. Интерпретация

$$10 \leq \frac{50-k}{3} < 11$$

$$30 \leq 50 - k < 33$$

$$-20 \leq -k < -17$$

$$17 < k \leq 20$$

$$k \in \{18, 19, 20\}$$

4. Учёт нечётности k

Из $\{18, 19, 20\}$ нечётное: $k = 19$



Ансамбль

Ответ: 70%.

Динамика акций

```
def main():  
    # Чтение количества дней  
    n = int(input().strip())  
  
    # Чтение параметра alpha  
    alpha = float(input().strip())  
  
    # Чтение n строк с ценами  
    prices = []  
    for _ in range(n):  
        price = float(input().strip())  
        prices.append(price)  
  
    # Инициализация сглаженных значений  
    smoothed = [prices[0]]  
  
    for t in range(1, n):  
        p_t = prices[t]  
        s_prev = smoothed[t-1]  
  
        #  $s_t = \alpha * p_t + (1-\alpha) * s_{t-1}$   
        s_t = alpha * p_t + (1 - alpha) * s_prev  
        smoothed.append(s_t)  
  
    # Вывод результатов с 3 знаками после запятой  
    for value in smoothed:  
        print(f"{value:.3f}")  
  
if __name__ == "__main__":  
    main()
```

Кредитный риск

Рассмотрим предсказания дерева на каждом объекте. На 5 объектах дерево выдает ошибочный прогноз, таким образом, точность составляет $11/16 = 0.6875 \approx 69\%$.

Клиент	Доход	Недвижимость	История	Эталон	Прогноз
1	120	8.5	780	3	3
2	85	5.2	720	2	3
3	45	3.1	650	1	1
4	25	1.2	580	0	0
5	150	12.0	810	3	3
6	60	2.8	620	1	2
7	35	0.8	550	0	0
8	95	6.5	740	2	3
9	40	4.5	680	1	1
10	28	0.9	520	0	0
11	110	7.8	760	3	3
12	70	3.5	630	1	2
13	22	0.5	480	0	0
14	130	9.2	790	3	3
15	55	2.2	610	1	2

Клиент	Доход	Недвижимость	История	Эталон	Прогноз
16	80	4.8	700	2	2

По грибы

1. Вероятности гипотез:

каждый пятый гриб оказывался съедобным, остальные – ядовитыми.

Это значит:

$$P(\text{Съедобный}) = 0.2$$

$$P(\text{Ядовитый}) = 0.8$$

2. Условные вероятности:

Если гриб съедобный, то приложение правильно определяет это в 80% случаев.

$$P(\text{"Съедобный"} \mid \text{Съедобный}) = 0.8$$

(ошибка для съедобных: 20%, т.е. говорит «ядовитый»)

Если гриб ядовитый, то приложение ошибается и говорит «съедобный» в 20% случаев.

$$P(\text{"Съедобный"} \mid \text{Ядовитый}) = 0.2$$

(правильно определяет ядовитый: 80%, т.е. говорит «ядовитый»)

3. Полная вероятность события «Приложение сказало "Съедобный"»:

$$P(\text{"Съедобный"}) = P(\text{Съедобный}) \cdot 0.8 + P(\text{Ядовитый}) \cdot 0.2$$

$$P(\text{"Съедобный"}) = 0.2 \times 0.8 + 0.8 \times 0.2$$

$$P(\text{"Съедобный"}) = 0.16 + 0.16 = 0.32$$

4. Вероятность, что гриб съедобный при условии «сказал "Съедобный"»:

$$P(\text{Съедобный} \mid \text{"Съедобный"}) = \frac{0.2 \times 0.8}{0.32}$$

$$P(\text{Съедобный} \mid \text{"Съедобный"}) = \frac{0.16}{0.32} = 0.5$$

5. Умножаем на 100 и округляем:

$$0.5 \times 100 = 50$$

Округляем $50 \rightarrow 50$ (целое уже).

Ответ:

Анализ данных

Подходящими строками являются **строка 14: (45,52,49)**, **строка 51: (49,53,45)**, **строка 97: (39,57,46)**, **строка 100: (54,41,50)**. Ответ: 4 строки.

Выбор порога

```
def main():  
    # Чтение количества точек  
    n = int(input().strip())  
    points = []  
  
    # Чтение точек  
    for _ in range(n):  
        x, y, label = map(int, input().strip().split())  
        points.append((x, y, label))  
  
    min_errors = n  
    best_T1 = 101  
    best_T2 = 101
```

```

# Перебираем все возможные пороги от -100 до 100
for T1 in range(-100, 101):
    for T2 in range(-100, 101):
        errors = 0
        for x, y, true_label in points:
            # Классификация:  $x < T1$  И  $y < T2$  -> класс 0, иначе -> класс 1
            pred = 0 if (x < T1 and y < T2) else 1

            if pred != true_label:
                errors += 1

        # Обновляем лучший результат
        if errors < min_errors:
            min_errors = errors
            best_T1 = T1
            best_T2 = T2
        elif errors == min_errors:
            # При равенстве ошибок выбираем минимальный T1
            if T1 < best_T1:
                best_T1 = T1
                best_T2 = T2
            elif T1 == best_T1 and T2 < best_T2:
                # При одинаковом T1 выбираем минимальный T2
                best_T2 = T2

# Вывод только чисел через пробел
print(min_errors, best_T1, best_T2)

if __name__ == "__main__":
    main()

```

Продажи мороженого

Сводная таблица прогнозов:

День	Факт	Простое среднее	Скользящее среднее	Эксп. сглаживание
1	12	23.5	-	12.0
2	18	23.5	-	15.0
3	24	23.5	18.0	19.5
4	30	23.5	24.0	24.75
5	36	23.5	30.0	30.375
6	21	23.5	29.0	25.6875

Итоговые MAE за дни 4-6:

Модель	MAE
Простое среднее	7.167
Скользящее среднее	6.667
Экспоненциальное сглаживание	5.188

Ответ: 7.167 6.667 5.188

Лаборатория

```
def classify_point(x, y, a, b):
```

```
    """Классифицирует точку относительно прямой  $y = ax + b$ """
```

```
    if y > a * x + b:
```

```
        return 1 # Выше прямой
    else:
        return 0 # Ниже прямой
```

```
def calculate_f1_score(points, labels, a, b):
```

```
    """Вычисляет F1-меру"""
```

```
    true_positive = 0
```

```
    false_positive = 0
```

```
    false_negative = 0
```

```
    for (x, y), true_label in zip(points, labels):
```

```
        predicted_label = classify_point(x, y, a, b)
```

```
        if true_label == 1 and predicted_label == 1:
```

```
            true_positive += 1
```

```
        elif true_label == 0 and predicted_label == 1:
```

```
            false_positive += 1
```

```
        elif true_label == 1 and predicted_label == 0:
```

```
            false_negative += 1
```

```
    # Вычисление precision и recall
```

```
    if true_positive + false_positive > 0:
```

```
        precision = true_positive / (true_positive + false_positive)
```

```
    else:
```

```
        precision = 0
```

```
    if true_positive + false_negative > 0:
```

```
        recall = true_positive / (true_positive + false_negative)
```

```
    else:
```

```
        recall = 0
```

```
    # Вычисление F1-меры
```

```
    if precision + recall > 0:
```

```
f1 = 2 * precision * recall / (precision + recall)
```

```
else:
```

```
f1 = 0
```

```
return f1
```

```
def main():
```

```
    # Чтение входных данных
```

```
    n = int(input().strip())
```

```
    points = []
```

```
    labels = []
```

```
    for _ in range(n):
```

```
        data = input().strip().split()
```

```
        x = int(data[0])
```

```
        y = int(data[1])
```

```
        label = int(data[2])
```

```
        points.append((x, y))
```

```
        labels.append(label)
```

```
    # Параметры прямой
```

```
    a = float(input().strip())
```

```
    b = float(input().strip())
```

```
    # Вычисление F1-меры
```

```
    f1_score = calculate_f1_score(points, labels, a, b)
```

```
    print(f"{'f1_score':.3f}")
```

```
if __name__ == "__main__":
```

```
    main()
```