

Муниципальный этап Всероссийской олимпиады для школьников 9-11 классы

ОЛИМПИАДНАЯ ИНФОРМАТИКА. ПРОФИЛЬ ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

Задания выставлены на платформу Яндекс.Контест, настроена автоматическая проверка заданий, ссылка:

<https://contest.yandex.ru/contest/85721/enter/?retPage=standings>.

Требуется обязательный доступ к яндекс диску в месте проведения олимпиады.

Общая сумма баллов: 100.

Для выполнения заданий по программированию рекомендуется использовать следующие языки программирования: C++, Python, Java, Pascal, C#. Для решения задачи с табличными данными требуется табличный процессор.

Требуется ограничить в браузерах выход на другие сайты и ресурсы для обучающихся при прохождении олимпиады. Разрешен доступ только на платформу Яндекс.Контест и Яндекс-диск.

Разбор задач

Задача 1. Ансамбль решающих деревьев

В ансамбле (комитете) из 5 независимых моделей решающих деревьев проводится голосование по бинарному вопросу («да»/«нет»). После принятия решения каждое дерево даёт 1 балл за выбранный вариант. Каждое дерево с вероятностью 0,7 принимает правильное решение. После голосования, если сумма баллов $\geq T$ (порог) у варианта, у которого больше баллов, то принимается этот вариант, иначе ансамбль выдаёт ошибку. Какое целое T необходимо поставить, если нужна максимальная вероятность правильного решения ансамбля при условии, что вероятность неправильного решения не должна оказаться больше 0,1?

Решение:

пусть вероятность правильного решения ансамбля при заданном T – $P(V|T)$,
неправильного – $P(X|T)$

Для проверки вариантов посчитаем вероятность возможных распределений правильных и неправильных решений деревьев (V-X):

$$5-0: 0,7^5 = 0,16807$$

$$4-1: C_5^1 * 0,7^4 * 0,3^1 = 0,36015$$

$$3-2: C_5^2 * 0,7^3 * 0,3^2 = 0,3087$$

$$2-3: C_5^3 * 0,7^2 * 0,3^3 = 0,1323$$

$$1-4: C_5^4 * 0,7^1 * 0,3^4 = 0,02835$$

$$0-5: 0,3^5 = 0,00243$$

Ставить T ниже 3 не имеет смысла, т.к. у варианта, у которого больше баллов, их минимум 3.

T=3 нам тоже не подойдёт, т.к. вероятность неправильного решения будет $0,00243+0,02835+0,1323=0,16308>0,1$.

В остальном нам нужен минимальный возможный T.

Итого: T=4, $P(V|4)=0,52822$, $P(X|4)=0,03078$.

Ответ: 4

Баллы: 10

Задача 2. Байесовский классификатор спама

Пользователь хочет сделать свой фильтр спама по содержимому, имитирующий Байесовский. Он собрал информацию, что 20% всех приходящих ему писем — спам. Слово «акция» встречается в 80% спам-писем и только в 10% не-спам-писем. Не разобравшись, пользователь решил просто поставить вероятность отправки в спам письма, если в нём есть слово «акция». Найдите, при каком пороговом значении вероятности $P(\text{в_спам} | \text{«акция»})$ цена ошибки «пропустить спам» (не заметить спам) в 3 раза выше, чем цена ошибки «ложное срабатывание» (пометить хорошее письмо как спам)

(Ценность оценивается в количестве тех или иных писем, т.е. пропустить 1 спам-письмо также плохо, как и отправить в спам 3 не-спам-письма).

Решение:

Запишем через вероятности условие про ценность:

$$3 * P(V | \text{спам}) * P(\text{спам}) = P(X | \text{не_спам}) * P(\text{не_спам})$$

Тогда всё остальное решается просто:

$$P(V | \text{спам}) = P(\text{не "акция"} | \text{спам}) + (1 - P) = 1,2 - P$$

$$P(X | \text{не_спам}) = P(\text{"акция"} | \text{не_спам}) = 0,1$$

$$3 * (1,2 - P) * 0,2 = 0,1 * 0,8$$

$$P \approx 1,067 > 1$$

Поэтому ставим $P=1$ (получатся вероятности 0,04 и 0,08)

Ответ: 1 ровно

Баллы: 12

Задача 3. Случайный лес и информативность признаков

В «Случайном лесу» для выбора лучшего признака при разбиении узла случайно выбирается подмножество из k признаков из общего числа n . Пусть существует ровно 3 «сильных» признака, которые идеально разделяют данные. Какова вероятность, что среди случайно выбранных k признаков окажется хотя бы один из этих трех, если $k=5$, $n=20$?

(Случайный лес — это популярный алгоритм машинного обучения, который используется для задач классификации (например, «отнести рекламу к запрещённой или допустимой») и регрессии по принципу использования различных вариантов сравнения / решения (признаков) и выбора общего. Узлы — группы данных для анализа лесом, после каждого разбиения на подгруппы (дочерние узлы) в них будут более чистые данные (в одном узле почти вся реклама разрешена, в другом почти вся запрещена))

Решение:

Легче найти вероятность противоположного события — «ни одного сильного признака».

Всего способов выбрать 5 признаков из 20:

$$C_{20}^5 = \frac{20!}{15!5!} = \frac{16*17*18*19*20}{1*2*3*4*5} = 16*17*3*19 = 15504$$

Способов выбрать 5 признаков только из 17 слабых:

$$C_{17}^5 = \frac{17!}{12!5!} = \frac{13*14*15*16*17}{1*2*3*4*5} = 13*14*2*17 = 6188$$

Вероятность отсутствия сильных признаков:

$$P = \frac{C_{17}^5}{C_{20}^5} = \frac{13*14*2*17}{16*17*3*19} = \frac{13*7}{4*3*19} \approx 0,399$$

Искомая вероятность: $1 - P \approx 0,601$.

Ответ: 0.601 в округлении до 3 знаков.

Баллы: 10

Задача 4. Размещение данных на серверах

ИИ-систему хотят обучить распределять задач по серверам для параллельных вычислений. Для составления первой обучающей выборки необходимо просчитать все возможные варианты распределения 10 идентичных задач по 4 различным серверам. Также было сразу решено ввести ограничение на вместимость серверов: первый сервер может принять не более 3 задач, второй — не более 4, третий — не более 5, четвёртый — не более 6 задач.

а) Сколькими способами можно это сделать, если каждый сервер может получить любое приемлемое количество задач (включая 0)?

б) Сколько существует способов, если каждый сервер должен получить хотя бы одну задачу?

Вследствие идентичности задач для первой обучающей выборки варианты отличаются только количеством задач на том или ином сервере, например, один из вариантов для пункта а) 3 – 4 – 3 – 0, т.е. первые 2 сервера заполнены, а все остальные задачи попали на третий сервер.

Решение:

Распределение n объектов в k групп (без учёта порядка (БУП)) = Выбор для каждого из n объектов (БУП) одной из k групп (выбранные группы могут совпадать) = Выбор n групп (БУП) из k доступных, но с повторениями = $\bar{C}_k^n = C_{k+n-1}^n$

$$\text{а) } n=10, k=4 - \bar{C}_4^{10} = C_{13}^{10} = \frac{13!}{10!3!} = \frac{13*12*11}{6} = 286 = K$$

Далее исключаем варианты, которые невозможны по условию вместимости (и возвращаем учтённые дважды):

$$K - K(4-X-X-X) - K(X-5-X-X) - K(X-X-6-X) - K(X-X-X-7) + K(4-5-X-X) + K(4-X-6-X) \\ = 286 - \bar{C}_3^6 - \bar{C}_3^5 - \bar{C}_3^4 - \bar{C}_3^3 + \bar{C}_2^1 + \bar{C}_2^0 = 286 - 28 - 21 - 15 - 10 + 3 + 1 = 216$$

б) Сначала дадим каждому серверу по одной задаче. Осталось распределить $10 - 4 = 6$ задач. Далее по аналогии:

$$n=6, k=4 - \bar{C}_4^6 = C_9^6 = \frac{9!}{6!3!} = \frac{9*8*7}{6} = 84$$

Далее исключаем варианты, которые невозможны по условию вместимости (возвращений не будет):

$$K - K(4-X-X-X) - K(X-5-X-X) - K(X-X-6-X) - K(X-X-X-7) = \\ 84 - \bar{C}_3^3 - \bar{C}_3^2 - \bar{C}_3^1 - \bar{C}_3^0 = 84 - 10 - 6 - 3 - 1 = 64$$

Ответ: 21664

Баллы: 16

Задача 5. Векторное позиционирование автономного робота-строителя

Автономная система ИИ для строительного робота-сборщика анализирует геометрию кубического модуля. Из одной вершины куба исходят три взаимно перпендикулярных вектора (ребра).

Системе идентифицированы два из них:

$$\bar{a} = 7\bar{i} + 6\bar{j} - 6\bar{k}$$

$$\bar{b} = 6\bar{i} + 2\bar{j} + 9\bar{k}$$

Формальная постановка: известно, что векторы $\bar{a} = 7\bar{i} + 6\bar{j} - 6\bar{k}$, $\bar{b} = 6\bar{i} + 2\bar{j} + 9\bar{k}$ могут быть ребрами куба. Найдите координаты третьего вектора.

Решение:

Векторное позиционирование автономного робота-строителя

Найдем третье ребро $\vec{c} = (x; y; z)$ куба. Так как $\vec{a} \perp \vec{c}$, то $\vec{a} * \vec{c} = 0$ т.е, $7x + 6y - 6z = 0$; Так как $\vec{b} \perp \vec{c}$, то $\vec{b} * \vec{c} = 0$ т.е, $6x + 2y + 9z = 0$; из равенств $|\vec{c}| = |\vec{a}| = |\vec{b}| = 11$ вытекает, что $\sqrt{x^2 + y^2 + z^2} = 11$. Для нахождения координат вектора решим систему уравнений

$$\begin{cases} 7x + 6y - 6z = 0 \\ 6x + 2y + 9z = 0 \\ x^2 + y^2 + z^2 = 121 \end{cases}$$

Из первых двух уравнений выражаем x и y через z ($x = -3z, y = \frac{9}{2}z$) и подставляем их значения в третье уравнение системы: $9z^2 + \frac{81}{4}z^2 + z^2 = 121$. Отсюда находим, что: $z = \pm 1$, тогда $x = \pm 6$ и $y = \pm 9$. Следовательно $\vec{c} = \pm(6\vec{i} - 9\vec{j} - 2\vec{k})$.

Ответ: 692

Баллы: 10

Задача 6. Данные для выполнения этого задания находятся в файле электронной таблицы. Вы можете скачать файл в одном из трёх форматов (XLSX, ODS, CSV) по ссылке https://disk.yandex.ru/d/MvnnrySVTQQG_w

Вы — инженер по ML. Ваша команда проводит предобработку данных и EDA (разведочный анализ) перед обучением модели, которая будет классифицировать клиентов банка по кредитоспособности. Этап предобработки данных включает проверку на выбросы, которые могут повлиять на качество модели классификации.

Постановка задачи:

Вам предоставили файл с данными. В первом столбце указана заработная плата клиента банка, а во втором столбце — его категория (на английском языке), которая была предварительно присвоена на основе уровня дохода (базовый(C), стандартный(B), премиум(A)).

Определите общее количество выбросов по всем категориям клиентов. Значение считается выбросом, если выполняется неравенство $|x - \bar{x}| > \bar{x}$

$\bar{x}$ – среднее арифметическое значение зарплаты в категории.

x – заработная плата клиента.

Решение:

Шаг 1. Определим среднее арифметическое для каждой категории. Внесём формулы в ячейки:

G1 =СРЗНАЧЕСЛИ(\$B\$1:\$B\$1200;"A";\$A\$1:\$A\$1200)

G2 =СРЗНАЧЕСЛИ(\$B\$1:\$B\$1200;"B";\$A\$1:\$A\$1200)

G3 =СРЗНАЧЕСЛИ(\$B\$1:\$B\$1200;"C";\$A\$1:\$A\$1200)

Используя автозаполнение, распространим формулы на все строки с данными.

Шаг 2. В ячейки C1, D1, E1 запишем формулы для определения наличия выброса в категории A, B и C соответственно

=ЕСЛИ(И(B1="A";ABS(A1-\$G\$1)>\$G\$1);1;0)

=ЕСЛИ(И(B1="B";ABS(A1-\$G\$2)>\$G\$2);1;0)

=ЕСЛИ(И(B1="C";ABS(A1-\$G\$3)>\$G\$3);1;0)

Используя автозаполнение, распространим формулы на все строки с данными.

Шаг 3. Определим общее количество выбросов по всем категориям, используя формулу =СУММ(C1:E1200)

Ответ: 22

Баллы: 14

Задача 7. Валидация цифрового следа

Привет, ИИ-стажёр! Сегодня мы обучаемся алгоритмической задаче на обработку текста: классифицировать строки на основе корректности расстановки скобок. Твоя цель — анализировать входную строку и выводить бинарный результат: "Корректно" (если все пары скобок правильно открыты/закрыты с учётом вложенности) или "Некорректно" (в противном случае). Скобки: круглые () и квадратные [] — ключи к балансу.

Пример 1

Ввод	Вывод
Nkldjdfs dfkjdfs	Корректно
Пример 2	
Ввод	Вывод
Некорректно(	Некорректно

Решение:

```
def check_brackets(s):
    stack = []
    bracket_pairs = {'(': ')', '[': ']', '{': '}'

    for char in s:
        if char in '([': # Открывающая скобка
            stack.append(char)
        elif char in ')]': # Закрывающая скобка
            if not stack or stack[-1] != bracket_pairs[char]:
                return False
            stack.pop()

    return len(stack) == 0

input_string = input()
if check_brackets(input_string):
    print("Корректно")
else:
    print("Некорректно")
```

Баллы: 14

Задача 8. Модель-классификатор сделала предсказание на валидационной выборке. Популярным способом собрать показатели *precision* и *recall* в одну метрику является метрика *F1*. Напишите программу, для оценки качества модели, используя метрику *F1*.

Метрика *F1* – это среднее гармоническое метрик *precision* и *recall*

TP (true positive)

FP (false positive)

TN (true negative)

FN (false negative)

$F_1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$	$Precision = \frac{TP}{TP + FP}$	$Recall = \frac{TP}{TP + FN}$
--	----------------------------------	-------------------------------

Формат входных данных:

- первая строка содержит одно целое число n – количество предсказаний ($n \geq 6$);
- следующие строки содержат истинные и предсказанные значения, разделённые пробелом.

Формат выходных данных

В ответе укажите значение метрики $F1$, округлив значение до 6-х знаков после запятой.

ВВОД	ВЫВОД
7 1 1 1 0 0 0 0 1 1 0 1 1 1 1	0.666667
12 1 1 0 1 1 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0 1 0 1 1	0.571429

Программа (решение):

```
n=int(input())
y_true=[]
y_pred=[]
```

```
for i in range(n):
    x1,y1 = list(map(int,input().split()))
    y_true.append(x1)
    y_pred.append(y1)
tp=fp=fn=0
for i in range(n):
    if y_pred[i]==y_true[i]==1:
        tp=tp+1
    if y_pred[i]==1 and y_true[i]==0:
        fp=fp+1
    if y_pred[i]==0 and y_true[i]==1:
        fn=fn+1
presision =tp/(tp+fp)
recall = tp/(tp+fn)
f1=2/(1/presision+1/recall)
print(f"{f1:.6f}")
```

Баллы: 14