

Муниципальный этап Всероссийской олимпиады для школьников 7-8 классы

ОЛИМПИАДНАЯ ИНФОРМАТИКА. ПРОФИЛЬ

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

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

<https://contest.yandex.ru/contest/85709/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#.

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. Модель-классификатор сделала предсказание на валидационной выборке. Популярным способом собрать показатели *precision* и *recall* в одну метрику является *F1*.

Для оценки качества модели, используйте метрику *F1*. В ответе значение метрики округлите до 4-х знаков после запятой. При вводе ответа в качестве разделителя целой и дробной части укажите точку.

y_{true} – истинные значения

y_{pred} – предсказанные значения

y_{true}	1	1	0	0	1	1	1	0	0	1	0	0	0	1	0
y_{pred}	1	0	0	1	0	1	1	1	0	1	0	1	0	1	0

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

TP (true positive) - верно предсказанные положительные объекты

FP (false positive) - отрицательные объекты, ошибочно предсказанные как положительные

TN (true negative) — верно предсказанные отрицательные объекты

FN (false negative) — положительные объекты, ошибочно предсказанные как отрицательные

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

Решение:

TP = 5 - модель предсказала класс 1 для случаев, где истинным является 1

FP = 3 - модель поставила класс 1, для истинного значения 0

TN = 5 - модель поставила класс 0 для класса 0

FN = 2 - модель поставила класс 0 для класса 1

$$Precision = \frac{5}{5 + 3} = \frac{5}{8} \quad Recall = \frac{5}{5 + 2} = \frac{5}{7}$$

$$\frac{1}{Recall} = \frac{7}{5} \quad \frac{1}{Precision} = \frac{8}{5}$$

$$F_1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} = \frac{2}{\frac{7}{5} + \frac{8}{5}} = \frac{2}{\frac{15}{5}} = \frac{2 * 5}{15} = \frac{2}{3} \approx 0.6667$$

Ответ: 0.6667

Баллы: 10

Задача 2. Байесовский классификатор спама

Байесовский фильтр спама анализирует письма на наличие двух слов: «акция» и «выигрыш». В спаме: $P(\text{«акция»}|\text{спам})=0,8$, $P(\text{«выигрыш»}|\text{спам})=0,6$.

В не-спаме: $P(\text{«акция»}|\text{не спам})=0,1$, $P(\text{«выигрыш»}|\text{не спам})=0,01$ (вертикальная черта «|» читается как «если»).

Пришло новое письмо, содержащее оба слова.

Предполагая условную независимость слов при наличии класса (спам/не спам), найдите вероятность, что это письмо — спам.

Решение:

применим формулу Байеса.

Пусть S — событие «письмо является спамом», A — событие «письмо содержит слова "акция" и "выигрыш"». Дано: $P(S) = 0,2$, $P(\text{не-}S) = 0,8$, $P(A|S) = 0,8*0,6=0,48$, $P(A|\text{не-}S) = 0,1*0,01=0,001$.

Найти: $P(S|A)$.

По формуле Байеса:

$$\begin{aligned} P(S|A) &= \frac{P(A|S) * P(S)}{P(A|S) * P(S) + P(A|\text{не-}S) * P(\text{не-}S)} = \frac{0,48 * 0,2}{0,48 * 0,2 + 0,001 * 0,8} = \\ &= \frac{0,096}{0,096 + 0,0008} = \frac{0,096}{0,0968} = \frac{120}{121} \approx 0,9917 \end{aligned}$$

Ответ: 0.992 в округлении до 3 знаков

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: 13

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.

а) Сколькими способами можно это сделать, если каждый сервер может получить любое 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) $10 - 0 - 0 - 0$, т.е. все задачи попали на первый 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 = $\overline{C}_k^n = C_{k+n-1}^n$

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

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

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

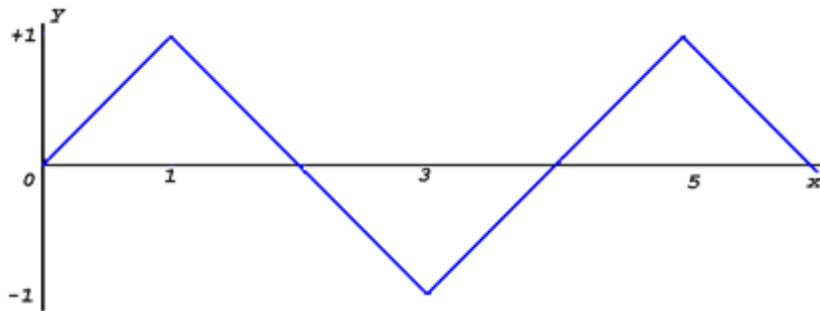
Ответ: 28684

Баллы: 17

Задача 5. Калибровка сенсорной сети

Вы — разработчик в команде по созданию автономного города. Вам необходимо запрограммировать модуль калибровки для сенсорной сети. Модуль получает "сырое" показание датчика x и должен вернуть

скорректированное значение y согласно заданному алгоритму, график которого был получен при тестировании оборудования.



Пример 1

Ввод

0.5

Вывод

0.5

Пример 2

Ввод

2

Вывод

0

Решение:

```
x = float(input().strip())
t = x % 4
if 0 <= t < 1:
    y = t
elif 1 <= t < 3:
    y = 2 - t
else: # 3 <= t < 4
    y = t - 4
print(y)
```

Баллы: 16

Задача 6. Оптимизация вектора внимания

Вы — инженер по машинному обучению. Вам поручено реализовать модуль пост-обработки для вектора весов в механизме внимания ИИ. Алгоритм должен выполнить следующую последовательность действий над целочисленным вектором $\vec{n}$:

1. Найти экстремумы: в векторе $\vec{n}$ необходимо найти минимальный (min) и максимальный (max) элементы.

2. Обработать особый случай: необходимо предусмотреть и корректно обработать ситуацию, когда все элементы вектора равны между собой. В этом случае минимальный и максимальный элементы совпадают, и сортировка между ними не требуется.

3. Отсортировать подмассив: все элементы вектора, расположенные между найденными минимальным и максимальным элементами (не включая их самих), должны быть отсортированы в порядке возрастания.

Формальная постановка:

в целочисленном векторе $\vec{n}$ найти минимальный и максимальный элементы. Предусмотреть случай, когда таких элементов нет (все элементы вектора равны между собой). Элементы вектора, расположенные между ними, отсортировать по возрастанию.

Пример

Ввод

7 -2 1 3 -5 5 2 0 10 1

Вывод

7 -2 1 3 -5 0 2 5 10 1

Решение:

```
numbers = list(map(int, input().split()))
```

```
if numbers:
```

```
    min_val = min(numbers)
```

```
    max_val = max(numbers)
```

```
    if min_val != max_val:
```

```
        min_index = numbers.index(min_val) #
```

```

max_index = numbers.index(max_val)
left = min(min_index, max_index)
right = max(min_index, max_index)
numbers[left+1:right] =
sorted(numbers[left+1:right])

print(*numbers)

```

Баллы: 20

Задача 7. Строительство аптек

На карте России появился новый город для молодых учёных и их семей «ИИНОВОГРАД». В городе уже построены три микрорайона «Градиент», «Кластер», «Нейрон» (рис.1). В каждом микрорайоне должна быть своя аптека. Принадлежность аптеки к микрорайону определяется по алгоритму k -ближайших соседей (для $k=9$).

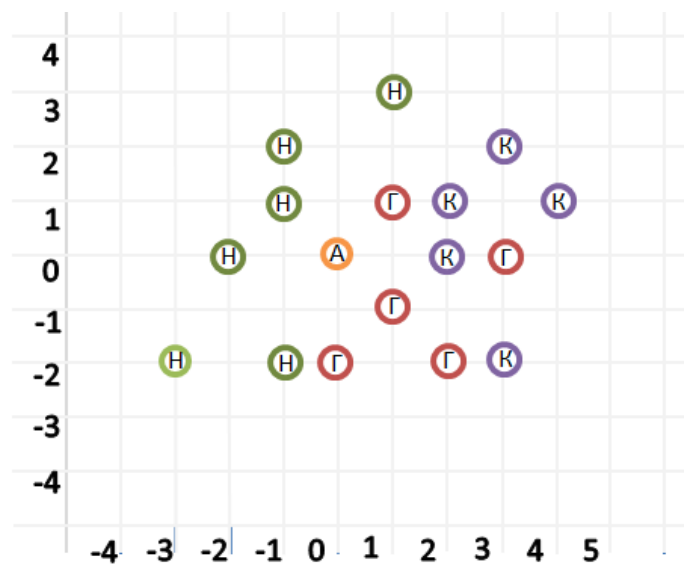


Рис.1 Расположение домов микрорайонов

Расположение домов в каждом микрорайоне на карте определяется координатами:

микрорайон «Нейрон»: (-1, -2), (-2, 0), (-1, 1), (-1, 2), (1, 3), (-3, 2)

микрорайон «Кластер»: (3, -2), (2, 0), (2, 1), (3, 2), (4, 1)

микрорайон «Градиент»: (0,-2), (1,-1), (1,1) (2,-2) (3,0)

Аптека: (0,0)

В ответе укажите в алфавитном порядке первые буквы названий микрорайонов, для которых надо построить аптеки. Пример ответа – *АБ*, где А- первая буква названия *микрорайона 1*, Б - первая буква названия *микрорайона 2*.

Решение:

Определяем квадраты расстояния между аптекой и домами по формуле:

$r_i^2 = (y_i - y_A)^2 + (x_i - x_A)^2$, $((x_i, y_i)$ – координаты расположения домов (x_A, y_A) – координаты аптеки.

Н: (-1, -2) - 5; (-2, 0) - 4, (-1, 1) - 2, (-1,2) - 5, (1, 3) -10, (-3, 2) -13

К: (3, -2) - 13, (2, 0) - 4, (2, 1) -5 (3,2)- 13 (4,1) – 17

Г: (0,-2) - 4, (1,-1) - 2, (1, 1) – 2, (2,-2) - 8 , (3, 0) - 9

Выписываем в порядке возрастания расстояния

$r^2 = 2$: Н, Г, Г

$r^2 = 4$: Н, К, Г

$r^2 = 5$: Н, Н,К

$r^2 = 8$: Г

Видим, что из 9 ближайших соседей на более близком расстоянии к аптеке находится больше всего домов микрорайона «Нейрон». Значит, аптеки нужно построить для Градиента и Кластера.

Ответ: ГК

Баллы: 12