A. Острова рекомендаций

Баллов за задачу: 100, по 20 за каждый вопрос
Формат сдачи ответа: ввод или загрузка файла, в зависимости от вопроса
Количество попыток: 10 на каждую подзадачу
Посылка в зачет: последняя

Условие

Вы работаете аналитиком в команде онлайн-маркетплейса. На сайте у каждого товара есть карточка с информацией (категория, цена, рейтинг, бренд, наличие) и блок «С этим также смотрят», где показаны другие товары, на которые пользователи часто переходят из данной карточки.

Вам выдали выгрузку двух таблиц в формате CSV:

- `items.csv` — список товаров и их свойства.
- `also_viewed.csv` — список переходов «с этим также смотрят».

Нужно ответить на несколько вопросов про товары и структуру рекомендательного графа. Ответы нужно получать с помощью программной обработки данных (например, на Python с использованием pandas и простых алгоритмов работы с графами).

Файл `items.csv` содержит информацию о товарах. Каждая строка – один товар. Поля:

- `item_id` – уникальный целочисленный идентификатор товара.
- `category` – категория товара (phones, accessories, laptops, books, home, toys).
- `price` – цена товара в условных единицах (целое число).
- `rating` – рейтинг товара по данным отзывов (вещественное число от 3.0 до 5.0 с шагом 0.1).
- `brand` – название бренда (строка).
- `in_stock` – 1, если товар есть в наличии, 0, если нет.

Файл `also_viewed.csv` описывает связи между товарами в блоке «с этим также смотрят». Каждая строка задаёт пару товаров (`item_from`, `item_to`), для которых зафиксировано, что пользователи часто переходят с одного на другой. В задачах, где речь идёт о «соседях» товара или о переходах между товарами, будем считать, что такая связь работает в обе стороны: если в таблице есть строка с парой товаров A и B (в любом порядке), то A и B считаются напрямую связанными рекомендациями. Для товара X его «соседями» считаются все товары, которые хотя бы в одной строке стоят в паре с X – неважно, указан X в `item_from` или в `item_to`.

Система оценивания

За эту задачу можно получить до 100 баллов. Каждый пункт стоит 20 баллов.

Результаты тестирования подзадач 1, 2, 3 и 5 **не доступны** во время проведения тура. Во всех подзадачах засчитывается последняя посылка.

A1 – Вопрос 1

Сколько всего товаров категории phones имеют рейтинг не ниже 4.5 (rating ≥ 4.5) и при этом есть в наличии (`in_stock` $= 1$)?

Формат вывода

Одно целое число – количество таких товаров.

Метрика оценивания точности ответа

Строгое совпадение введенного ответа.

A2 – Вопрос 2

Рассмотрим только товары категории laptops. Для каждого бренда посчитайте среднюю цену ноутбуков этого бренда. Какой бренд имеет максимальную среднюю цену среди ноутбуков?

Если несколько брендов имеют одинаковую максимальную среднюю цену, можно вывести любой из них.

Формат вывода

Одно слово – название бренда (строка brand из файла `items.csv`).

Метрика оценивания точности ответа

Строгое совпадение введенного ответа.

A3 – Вопрос 3

Команда маркетинга хочет разделить товары на три сегмента по цене и рейтингу:

- сегмент **premium** – если rating ≥ 4.5 и price ≥ 50000;
- сегмент **standard** – если rating ≥ 4.0 и price < 50000;
- сегмент **budget** – во всех остальных случаях.

Для каждого товара определите его сегмент (**premium** / **standard** / **budget**) по этим правилам. Среди товаров, которые есть в наличии (`in_stock` $= 1$), посчитайте, сколько товаров относится к сегменту **premium**.

Формат вывода

Одно целое число — количество товаров сегмента premium среди товаров с `in_stock` $=$ 1.

Метрика оценивания точности ответа

Строгое совпадение введенного ответа.

A4 – Вопрос 4

Рассмотрим файл `also_viewed.csv`. Найдите все товары, которые хотя бы один раз встречаются в поле item_to (то есть товары, которые хотя бы раз были показаны в блоке «С этим также смотрят»). Для каждой категории посчитайте, сколько разных товаров из этой категории встречается в `item_to` хотя бы один раз.

Нужно подготовить таблицу с двумя столбцами:

- `category` – название категории;
- `cnt` – количество разных товаров этой категории, которые встречаются в `item_to`.

В таблицу следует включить все категории, которые есть в файле `items.csv`, даже если для какой-то категории `cnt` = 0. Строки в таблице нужно отсортировать по названию категории в алфавитном порядке.

Формат вывода

Текстовый файл `answer4.csv` в формате CSV с заголовком и двумя колонками:

```
category,cnt
```

Файл должен содержать ровно по одной строке для каждой категории.

Метрика оценивания точности ответа

Доля категорий `category` в вашем файле-ответе для которых количество `cnt` совпадает с количеством `cnt` в эталонном файле-ответе.

A5 – Вопрос 5

Будем рассматривать связи между товарами, как описано в разделе «Описание датасета»: два товара считаются напрямую связанными, если в `also_viewed.csv` есть строка, где они стоят парой (в любом порядке).

Назовём «островом рекомендаций» любое множество товаров, внутри которого из любой карточки можно добраться до любой другой карточки, переходя по прямым связям между товарами (по соседям). Если два товара относятся к разным островам рекомендаций, то никакой цепочкой таких переходов из одного к другому попасть нельзя.

Нас интересуют такие острова рекомендаций, в которых одновременно есть хотя бы один товар категории phones и хотя бы один товар категории accessories.

Сколько таких островов рекомендаций существует в наших данных?

Формат вывода

Одно целое число – количество «островов рекомендаций», в которых есть и хотя бы один phones, и хотя бы один accessories.

Метрика оценивания точности ответа

Строгое совпадение введенного ответа.

B. Кластеризация

Баллов за задачу: 100
Формат сдачи ответа: загрузка файла-ответа в формате .csv
Количество попыток: 20
Посылка в зачет: последняя

Условие

По дороге на региональный этап ВсОШ ИИ Миша нашел флешку с брелком, на котором написано “методкомиссия”. На флешке оказался табличный файл с названием `data.csv`. Поскольку целевой переменной в `csv` файле Миша не обнаружил, он справедливо заключил, что это должна быть задача на кластеризацию. Однако, информацию про количество кластеров Мише обнаружить нигде не удалось. Помоги Мише понять количество кластеров и правильно кластеризовать данные.

Формат ввода

К задаче прикреплены файлы:

- `data.csv` - содержит матрицу объекты-признаки (каждая строка таблицы - объект, каждая колонка - признак). Колонка `id` - идентификатор объекта. Остальные колонки - признаки.
- `baseline.ipynb` - ноутбук с базовым решением задачи.
- `submission.csv` - пример решения, которое вам нужно отправить в тестирующую систему.

Формат вывода

Вам нужно отправить как посылку файл `submission.csv`, содержащий две колонки:

- `id` - идентификатор объекта из `data.csv`.
- `cluster` - предсказанный вами кластер объекта (целое положительное число).

Система оценивания

За эту задачу можно получить до 100 баллов.

Данные разбиты на публичную и приватную части. Когда вы отправляете submission.csv, вам показывается результат на публичной части. После завершения контеста ваш результат будет пересчитан на приватной части.

После окончания этапа ваша метрика будет приведена к 100-балльной шкале по следующему правилу:

- результат **baseline-решения** (ARI=0.0) оценивается в **0 баллов**;
- результат **авторского решения** (ARI=0.9814) оценивается в **100 баллов**;
- результаты между этими точками распределяются линейно.

Метрика оценивания точности ответа

В этой задаче используется метрика **ARI (Adjusted Rand Index)**. Чем большее количество пар объектов Миша правильно распределяет по кластерам (например, если оба объекта находятся в разных кластерах и Миша также разделяет их по

разным кластерам, ИЛИ оба объекта находятся в одном кластере и Миша тоже их определяет в один кластер), тем выше эта метрика. ARI принимает значение 0 для случайного разбиения на кластеры, значение 1 для идеально правильного разбиения, и может принимать отрицательные значения в случае неудачного разбиения хуже случайного.

Пример расчета метрики **ARI** на *Python*:

```python
from sklearn.metrics import adjusted_rand_score

labels_true = [0, 0, 1, 1, 2, 2]
labels_pred = [1, 1, 0, 0, 2, 2]

ari = adjusted_rand_score(labels_true, labels_pred)
print("ARI =", ari)
```

C. Стоимость аренды квартир

Баллов за задачу: 100
Формат сдачи ответа: загрузка файла-ответа в формате .csv
Количество попыток: 20
Посылка в зачет: последняя

Условие

Пока Семён готовился к решению регионального этапа ВсОШ и мечтал, как получит свой БВИ, он решил прикинуть, какую квартиру он сможет снять на деньги, накопленные на ML олимпиадах, если его не поселят в общежитие рядом с университетом. Для этого он соскрэйпил данные с сайтов про аренду недвижимости и решил построить модель предсказания стоимости аренды, чтобы затем найти самые выгодные предложения. Однако, из-за того, что данные собирались не слишком аккуратно и с разных сайтов, датасет получился достаточно "грязным". Помоги Семёну аккуратно обработать данные и получить наилучшее качество прогноза стоимости аренды.

Формат ввода

К задаче прикреплены файлы:

- `train.csv` - колонка `price` - целевая переменная. Остальные колонки - признаки.
- `test.csv` - колонка `id` - идентификатор объекта. Остальные колонки - признаки.
- `baseline.ipynb` - ноутбук с базовым решением задачи.
- `submission.csv` - пример решения, которое вам нужно отправить в тестирующую систему.

Формат вывода

Вам нужно отправить как посылку файл `submission.csv`, содержащий две колонки:

- `id` - идентификатор объекта из `test.csv`.
- `price` - предсказанная вами целевая переменная.

Система оценивания

За эту задачу можно получить до 100 баллов.

Данные разбиты на публичную и приватную части. Когда вы отправляете `submission.csv`, вам показывается результат на публичной части. После завершения контеста ваш результат будет пересчитан на приватной части.

После окончания этапа ваша метрика будет приведена к 100-балльной шкале по следующему правилу:

- результат **baseline-решения** (RMSE=21.046) оценивается в **0 баллов**;
- результат **авторского решения** (RMSE=13.8) оценивается в **100 баллов**;
- результаты между этими точками распределяются линейно.

Метрика оценивания точности ответа

В этой задаче используется метрика **RMSE**.

Строгое математическое определение метрики **RMSE**:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

y_i — истинное значение, $\hat{y}_i$ — предсказание, n — число объектов.

Пример расчета метрики **RMSE** на *Python*:

```python
from sklearn.metrics import root_mean_squared_error

y_true = [3.0, -0.5, 2.0, 7.0]
y_pred = [2.5,  0.0, 2.1, 7.8]

rmse = root_mean_squared_error(y_true, y_pred)
print("RMSE =", rmse)
```

D. Марсианский Архивариус

Баллов за задачу: 100
Формат сдачи ответа: загрузка файла-ответа в формате .csv
Количество попыток: 20
Посылка в зачет: последняя

Условие

Во время третьего года экспедиции в долине Маринер на Марсе наш знакомый инженер Андрей — специалист по внеземным системам — наткнулся на нечто невероятное: идеально сохранившийся кристаллический модуль памяти, скрытый глубоко под поверхностью каньона.

Когда модуль осторожно извлекли и подключили к питанию, он пробудил древний марсианский ИИ, назвавший себя «Архивариус блок F» — хранителем знаний исчезнувшей цивилизации. Архивариус поведал, что в его памяти содержатся обширные сведения о кристаллах, которые древние марсиане использовали в своих лабораториях и энергетических реакторах.

Каждый кристалл был описан эмбеддингом — вектором длины 16, отражающим его структуру, состав, резонансные свойства и ещё множество характеристик, которые люди пока не умеют интерпретировать напрямую. Помимо эмбеддинга, Архивариус хранил и класс (из 25 возможных) — тип или функциональное назначение кристалла.

Но за тысячи лет под марсианской пылью Архивариус был повреждён.

Для многих кристаллов информация о классе оказалась полностью утеряна.

Для других сохранилась только частично: вместо одного точного класса Архивариус выдавал несколько возможных вариантов, иногда разумных, а иногда — совершенно случайных. Похоже, что структуры данных внутри модуля перемешались, и никакой простой метод восстановления информации не работает.

Андрея и его команду чрезвычайно интересуют древние знания о марсианских кристаллах — понимание их свойств может стать прорывом в энергоёмких технологиях и материаловедении.

Вот почему они обращаются к вам.

Ваша задача — помочь Архивариусу восстановить истинные классы тех кристаллов, для которых информация была утеряна или повреждена. Вам будут предоставлены:

- эмбеддинги кристаллов,
- корректные классы для части из них,
- неоднозначные списки возможных классов для остальных,
- а также набор кристаллов, чьи классы предстоит предсказать.

Как и древний ИИ, вам придётся работать в условиях неопределённости и неполной информации. Однако современные методы машинного обучения дают шанс восстановить значительную часть утраченных знаний — если применить их достаточно аккуратно и изобретательно.

Формат ввода

К задаче прикреплены файлы:

- `train.csv` - содержит информацию о кристаллах, для которых известны истинные или возможные метки. Поля:
  - `id` — уникальный идентификатор объекта.
  - `F{i}`, где $i \in \{1, \ldots, 16\}$ — компоненты эмбеддинга.
  - `labels` — набор возможных классов для данного объекта (истинный класс может присутствовать среди них, но может и отсутствовать).
- `test.csv` — файл с эмбеддингами кристаллов, чьи классы необходимо предсказать. Гарантируется, что каждый объект относится ровно к одному из 25 классов.
- `baseline.ipynb` — ноутбук с базовым решением задачи.
- `submission.csv` — пример решения, которое вам нужно отправить в тестирующую систему.

Формат вывода

Вам нужно отправить как посылку файл `submission.csv`, содержащий две колонки:

- `id` — идентификатор объекта из test.csv.
- `class` — предсказанная моделью метка класса

Система оценивания

Максимум за задачу — 100 баллов.

Данные тестовой выборки разделены на публичную и приватную части.

После отправки решения система показывает результат на публичной части.

Окончательный результат после завершения контеста будет рассчитан по приватной части. После окончания этапа ваша метрика будет приведена к 100-балльной шкале по следующему правилу:

- результат **baseline-решения** (Accuracy=0.2717) оценивается в **0 баллов**;
- результат **авторского решения** (Accuracy=0.8) оценивается в **100 баллов**;
- результаты между этими точками распределяются линейно.

Метрика оценивания точности ответа

В этой задаче используется метрика **Accuracy**. Она считается как доля объектов тестовой выборки, для которых класс предсказан верно.

Строгое математическое определение метрики **Accuracy**:

$$\text{Accuracy} = \frac{\text{число верных ответов}}{\text{общее число тестовых кристаллов}}.$$

Пример расчета метрики **Accuracy** на *Python*:

```
from sklearn.metrics import accuracy_score

y_true = [0, 1, 2, 2, 1]
y_pred = [0, 2, 1, 2, 1]

acc = accuracy_score(y_true, y_pred)
print("Accuracy =", acc)
```