HearMe.Personalized Music Recommender

Kazakh - HearMe

HearMe.Personalized Music Recommender

Шолу (Overview)

🏆 **Музыкалық ұсыныс жүйесін жасау бойынша жарысқа қош келдіңіздер!** 🎶

Біздің командамыз:

- **Джалиль** — hitter және Izi компаниясының Product Owner-ы, әрбір қолданушы өз күніне мінсіз саундтрек табуын қалайды 🎧
- **Дархан** — қолданушылардың орасан көп деректерін ақылды, жекелендірілген ұсыныстарға айналдыру міндеті жүктелген Data Scientist
- **Әділхан** — платформаның белсенді қолданушысы, жаңа музыкалық жаңалықтарды іздеуге құмар әрі қызығушылығы жоғары тыңдарман 🎵

Сіздің міндетіңіз: Дарханға әр қолданушының тыңдау тарихы мен қалауына негізделген ең релевантты 50 тректі болжайтын ұсыныс жүйесін құруға көмектесіңіз.

Датасет пайдаланушылар Hitter және Izi платформаларындағы күн сайын тыңдайтын, ұнататын және жүктеп алатын тректердің арасынан алынған. Сіз классикалық машиналық оқытудан бастап терең оқытуға немесе гибридті модельдерге дейін, әр түрлі тәсілдерді зерттей аласыз. Осыны Әділханды қайта-қайта «play» батырмасын басқызатын мінсіз плейлист құрылуы деп есептеңіз! 🚀

🎯 **Мақсат (Goal)**

Әрбір қолданушыға 50 ән ұсынатын модель құрыңыз, әрбір тыңдаушының қай әндерді көбірек ұнататынын және оларды қаншалықты тыңдайтынын болжауға тырысыңыз. Дарханға Әділхан сияқты қолданушыларды шынымен түсінетін, шикі деректерді үздіксіз, жекелендірілген музыкалық тәжірибеге айналдыратын модельмен Джалилге әсер етуге көмектесіңіз.🔥

📊 **Бағалау (Evaluation)**

Ұсыныс сапасын қалай өлшейміз:

1. Тректерді болжау
Сіздің моделіңіз әрбір қолданушыға (мысалы, Әділханға) ұнауы мүмкін 50 тректен тізімін болжайды.

2. Тыңдау жиілігін тексеру

Қолданушы әрбір ұсынылған трегің шынымен қаншалықты тыңдағанын тексереміз.

- 0.0 — мүлдем тыңдалмаған
- 0.25 — тректің төрттен бір бөлігі тыңдалған
- 0.5 — жартысы тыңдалған
- 0.75 — төрттен үш бөлігі тыңдалған
- 1.0 — толық тыңдалған (қайталама тыңдаулар қосымша есепке алынбайды)

3. Метриканы әр пайдаланушыға есептеу
Осы 50 трек бойынша жиналған үлестерің қосамыз → бұл қолданушының метрикасын береді (диапазоны: 0-ден 50-ге дейін).

4. Қолданушылар бойынша орташа мәнді есептейміз
Біз барлық қолданушылардың метриканың орташа мәнін есептейміз → бұл әрбір қолданушының орташа есеппен қанша трек тыңдағанын көрсетеді (диапазоны: 0-ден 50-ге дейін).

5. Қорытынды метрика
0 мен 1 арасында нормаланған метрика.

📈 Метриканы қалай түсіндіру керек

- **Метрика ≥ 0.25** → Қолданушылар ссіздің ұсыныстарыңызбен әрекеттесе бастайды. Тіпті жолдың төрттен бір бөлігі де шағын жеңіс болып саналады!
- **Метрика ~0.5** → Қолданушылар сіздің ұсыныстарыңыздың жартысына жуығын тыңдайды. Жарайсыз! Сіздің ұсыныстарыңыз қызықты екені анық.
- **Метрика ~0.75** → Көптеген тректер жолдың төрттен үш бөлігінде ойнатылады. Керемет жұмыс! Қолданушылар сіздің ұсыныстарыңызды шынымен бағалайды.
- **Метрика ~1.0** → Қолданушылар ұсынылған тректердің барлығын толығымен дерлік тыңдайды. Тамаша! Сіз бәрін дұрыс жасадыңыз.

📤 Шешімді жіберу (Submission)

Шолу

Сіздің шешіміңізде сынақ деректер жинағындағы әрбір қолданушы үшін рейтингіленген ұсыныстар болуы керек. Әрбір қолданушыда дәл 50 трек болуы керек, олардың әрқайсысында 1-ден 50-ге дейін сәйкес рейтингі болуы керек, мұндағы 1-ші рейтинг ең релевантты трек, ал 50-ші рейтинг ең аз релевантты трек болып табылады.

Файл форматы

- **Файл түрі:** CSV файлы

Бағандардың сипаттамасы

Баған	**Түрі**	**Сипаттама**	**Шектеулер**
id	Integer	Бірегей жол идентификаторы	Реттік, 0-ден бастап
user_id	Integer	Тест жинағынан қолданушы идентификаторы	Тест жинағынан user_ids-пен сәйкес келуі керек
item_id	Integer	Трек идентификаторы	Жарамды item_id болуы керек
rank	Integer	Ұсыныстың рангі	1-50 (1 = ең жоғары, 50 = ең төмен)

Талаптар

- Барлық бағандар міндетті — кез келген бағанның болмауы жіберу қатесіне әкеледі
- Тест жинағындағы әр қолданушыда дәл 50 ұсыныс болуы керек
- Әрбір қолданушы-трек жұбы үшін рейтингтер бірегей болуы керек (1-ден 50-ге дейін)
- Бір қолданушы үшін қайталанатын трек ұсыныстары болмауы керек
- id бағаны міндетті

id бағанын жасау үлгісі:

```
recos.insert(0, 'id', range(len(recos)))
```

Жіберу үлгісі

id	user_id	item_id	rank
0	1	123	1
1	1	312	2
2	1	321	3
3	1	123	31
4	1	123	12
5	1	1	6
6	1	73	7
7	1	998	…
…	…	…	…
49	1	70	50

Датасет сипаттамасы (Dataset Description)

Бұл датасет **hitter** және **Izi** музыкалық платформаларындағы **нақты қолданушы әрекеттерінен** алынған. Ол **қолданушы-трек әрекеттерін**, **трек метадеректерін** және **қолданушы метадеректерін** қамтиды.

Оқу жиынтығы қолданушылардың **2025-02-28** және **2025-08-30** аралығындағы тректерде өзара әрекеттесуін қамтиды (6 айлық кезең).

Тест жинағы 2025-09-01-ден 2025-09-15-ке дейінгі қолданушы әрекеттерін білдіреді, ұсыныс жүйесінің өнімділігін бағалау үшін қолданылады. Ол тек оқу әрекеттерінде кездесетін қолданушыларды қамтиды (**cold-start қолданушылары жоқ**).

> ⚠️ Ескертпе: interactions.csv файлында нақты деректерге тән шу, қайталанатын деректер немесе шағын қателер болуы мүмкін.

> ⚠️ Ескертпе: item_metadata.csv, user_metadata.csv файлдарында жетіспейтін немесе толық емес жазбалар болуы мүмкін.

Файлдар

- **interactions.csv** — Қолданушылардың тректермен әрекеттесу жазбалары
- **item_metadata.csv** — Тректер туралы метадеректер
- **user_metadata.csv** — Қолданушылар туралы метадеректер
- **test.csv** — Болжамдар жасалуы қажет қолданушылар тізімі

---

interactions.csv

Баған	**Сипаттама**	**Деректер түрі**	**Ескертпелер**
user_id	Қолданушының бірегей идентификаторы.	Integer	user_metadata.csv файлындағы қолданушыларға сілтеме жасайды.
item_id	Тректің бірегей идентификаторы.	Integer	item_metadata.csv файлындағы тректерге сілтеме жасайды.
listened_duration	Қолданушының осы әрекетте тректі тыңдаған ұзақтығы (секундпен).	Integer	Ішінара немесе шулы болуы мүмкін; өткізіп жіберулер/шығарлықтарды қамтиды.
listened_datetime	Әрекет болған уақыттың белгісі.	Datetime	Формат: YYYY-MM-DD HH:MM:SS.ssssss.

> 🕒 **Уақыт ауқымы:** 2025-02-28 → 2025-08-30 (6 айлық өндірістік деңгейдегі деректер)

---

item_metadata.csv

Баған	Сипаттама	Деректер түрі	Ескертпелер
item_id	Тректің бірегей идентификаторы.	Integer	Тректердің негізгі кілті.
track_name	Тректің атауы/тақырыбы.	String	Арнайы символдарды немесе ағылшын емес мәтінді қамтуы мүмкін.
artist_name	Орындаушы(лар) немесе атқарушы(лар) аты.	String	Бірлескен жұмыстарды қамтуы мүмкін (мысалы, "Artist feat. Other").
track_duration	Тректің толық ұзақтығы секундпен.	Integer	Тректің жалпы ұзындығы.
track_genres_list	Трекпен байланысты жанрлардың үтірмен бөлінген тізімі.	String	Мысал: ['Genre1', 'Genre2']; ағылшын емес жанрларды қамтуы мүмкін.
track_dislike_count	Трек алған жалпы дизлайктар саны.	Integer	Жинақталған метрика; жаңа тректер үшін 0-ден басталады.
track_like_count	Трек алған жалпы лайктар саны.	Integer	Жинақталған метрика; жаңа тректер үшін 0-ден басталады.
track_download_count	Тректің жалпы жүктелімдер саны.	Integer	Жинақталған метрика; жаңа тректер үшін 0-ден басталады.

---

user_metadata.csv

Баған	Сипаттама	Деректер түрі	Ескертпелер

user_id	Қолданушының бірегей идентификаторы.	Integer	Қолданушылардың негізгі кілті.
age_bin	Қолданушының жас тобы (ML моделіне негізделген).	String	Мысалдар: '18_29', '30_44', '45_60', '61_inf'.
children	Қолданушының балаларының саны (ML моделіне негізделген).	Integer	0–1
gender	Қолданушының жынысы (ML моделіне негізделген).	String	'M' — ер адам, 'F' — әйел.
top_genre	Қолданушының артық көретін немесе анықталған басты жанры.	String	Мысалдар: 'Поп', 'Джаз', 'Chill'; ағылшын емес тілде болуы мүмкін.
user_disliked_track_count	Қолданушы дизлайк қойған тректердің жалпы саны.	Integer	Қолданушы деңгейінде жинақталған метрика.
user_liked_track_count	Қолданушы лайк қойған тректердің жалпы саны.	Integer	Қолданушы деңгейінде жинақталған метрика.
user_downloaded_track_count	Қолданушы жүктеген тректердің жалпы саны.	Integer	Қолданушы деңгейінде жинақталған метрика.

---

test.csv

Баған	**Сипаттама**	**Деректер түрі**	**Ескертпелер**
user_id	Қолданушының бірегей идентификаторы.	Integer	Барлық қолданушылар оқу әрекеттерінде бар; cold-start қолданушылары жоқ.

🧭 **Уақыт ауқымы:** 2025-09-01 → 2025-09-15 (бағалау үшін әрекеттер кезеңі)

Болжам форматы (Prediction Format)

Тапсыру файлында тест жинағындағы әрбір пайдаланушыға арналған ұсыныстар болуы керек, олар өзектілігі бойынша рейтингке ие болуы керек.

Баған	**Сипаттама**	**Деректер түрі**
id	id реттілігі.	Integer
user_id	Қолданушы идентификаторы (test.csv-мен сәйкес келеді).	Integer
item_id	Ұсынылған трек идентификаторы.	Integer
rank	Ұсыныстың рейтингі (1-ден 50-ге дейін басталады).	Integer

Russian - HearMe

HearMe.Personalized Music Recommender

Обзор (Overview)

🏆 Добро пожаловать на соревнование по созданию музыкальной рекомендательной системы! 🎶

Наша команда:

- **Джалиль** — Product Owner в hitter и Izi, хочет, чтобы каждый пользователь нашёл идеальный саундтрек для своего дня 🎧
- **Дархан** — Data Scientist, которому поручено превратить горы пользовательских данных в умные персонализированные рекомендации 💡
- **Адильхан** — активный пользователь платформы, любознательный и жаждущий новых музыкальных открытий 🎵

Ваша миссия: помочь Дархану построить систему рекомендаций, которая предсказывает топ-50 наиболее релевантных треков для каждого пользователя на основе их истории прослушивания и предпочтений.

Датасет получен из реальных взаимодействий на платформах **hitter и Izi**, где пользователи ежедневно слушают, лайкают и скачивают треки. Вы можете изучить любой подход — от классического машинного обучения до глубокого обучения или гибридных моделей. Думайте об этом как о создании идеального плейлиста, который заставит Адильхана нажимать кнопку "play" снова и снова! 🚀

🎯 Цель (Goal)

Построить модель, которая рекомендует 50 треков для каждого пользователя, стремясь предсказать, какие песни каждому слушателю понравятся больше всего — и сколько они на самом деле будут их слушать. Помогите Дархану впечатлить Джалиля моделью, которая действительно понимает пользователей вроде Адильхана, превращая сырые данные в бесшовный персонализированный музыкальный опыт. 🔥

📊 Оценка (Evaluation)

Как мы измеряем качество рекомендаций:

1. Предсказываем треки
Для каждого пользователя (например, для Адильхана), ваша модель предсказывает список из 50 треков, которые ему могут понравиться.

2. Проверяем доли прослушивания
Для каждого рекомендованного трека мы проверяем, насколько пользователь на самом деле его прослушал:

- 0.0 — не прослушан вообще
- 0.25 — прослушана четверть трека
- 0.5 — прослушана половина
- 0.75 — прослушаны три четверти
- 1.0 — прослушан полностью (повторные прослушивания не учитываются дополнительно)

3. Считаем метрику на пользователя
Суммируем эти доли по 50 трекам → это даёт метрику пользователя (диапазон: от 0 до 50).

4. Считаем среднее по пользователям
Вычисляем среднее значение метрики всех пользователей → это говорит нам, сколько треков в среднем каждый пользователь на самом деле прослушал (диапазон: от 0 до 50).

5. Финальная метрика
Нормализованная метрика от 0 до 1.

📈 Как интерпретировать метрику

- **Метрика ≥ 0.25** → Пользователи начинают взаимодействовать с вашими рекомендациями. Даже четверть трека считается маленькой победой!
- **Метрика ~0.5** → Пользователи слушают примерно половину ваших рекомендаций. Хорошая работа! Ваши предложения явно интересны.
- **Метрика ~0.75** → Большинство треков прослушиваются на три четверти. Отличная работа! Пользователям действительно нравятся ваши рекомендации.
- **Метрика ~1.0** → Пользователи полностью прослушивают почти все рекомендованные треки. Превосходно! Вы попали в точку.

📤 Отправка решения (Submission)

Обзор

Ваше решение должно содержать ранжированные рекомендации для каждого пользователя в тестовом наборе данных. У каждого пользователя должно быть ровно 50 треков с

соответствующим рангом от 1 до 50, где ранг 1 — это наиболее релевантный трек, а ранг 50 — наименее релевантный трек.

Формат файла

- **Тип файла:** CSV файл

Спецификация колонок

Колонка	Тип	Описание	Ограничения
id	Integer	Уникальный идентификатор строки	Последовательный, начиная с 0
user_id	Integer	Идентификатор пользователя из тестового набора	Должен соответствовать user_ids из тестового набора
item_id	Integer	Идентификатор трека	Должен быть валидным item_id
rank	Integer	Ранг рекомендации	1-50 (1 = наивысший, 50 = наименьший)

Требования

- Все колонки обязательны — отсутствие любой колонки приведёт к ошибке отправки
- У каждого пользователя в тестовом наборе должно быть ровно 50 рекомендаций
- Ранги должны быть уникальными для каждой пары пользователь-трек (от 1 до 50)
- Не должно быть повторяющихся рекомендаций треков для одного пользователя
- Колонка id обязательна

Пример создания колонки id:

```
recos.insert(0, 'id', range(len(recos)))
```

Пример отправки

id	user_id	item_id	rank
0	1	123	1
1	1	312	2

2	1	321	3
3	1	123	31
4	1	123	12
5	1	1	6
6	1	73	7
7	1	998	…
…	…	…	…
49	1	70	50

Описание датасета (Dataset Description)

Этот датасет получен из **реальных пользовательских взаимодействий** на музыкальных платформах **hitter** и **Izi**. Он включает **взаимодействия пользователь-трек**, **метаданные треков** и **метаданные пользователей**.

Обучающий набор охватывает **взаимодействия пользователей с треками** с **2025-02-28 по 2025-08-30** (6-месячный период).

Тестовый набор представляет **взаимодействия пользователей** с **2025-09-01 по 2025-09-15**, используемые для оценки качества рекомендаций. Он включает только пользователей, которые присутствуют в обучающем наборе данных (**нет cold-start пользователей**).

> ⚠️ Примечание: interactions.csv может содержать шум, дубликаты или незначительные ошибки, характерные для реальных данных.

> ⚠️ Примечание: item_metadata.csv, user_metadata.csv могут иметь отсутствующие или неполные записи.

Файлы

- **interactions.csv** — Записи взаимодействий пользователей с треками
- **item_metadata.csv** — Метаданные о треках
- **user_metadata.csv** — Метаданные о пользователях
- **test.csv** — Список пользователей, для которых необходимо сделать предсказания

---

interactions.csv

Колонка	Описание	Тип данных	Примечания
user_id	Уникальный идентификатор пользователя.	Integer	Ссылается на пользователей в user_metadata.csv.
item_id	Уникальный идентификатор трека.	Integer	Ссылается на треки в item_metadata.csv.
listened_duration	Длительность (в секундах), в течение которой пользователь слушал трек в этом взаимодействии.	Integer	Может быть частичной или содержать шум; включает пропуски/выбросы.
listened_datetime	Временная метка, когда произошло взаимодействие.	Datetime	Формат: YYYY-MM-DD HH:MM:SS.ssssss.

> 🕒 **Временной диапазон:** 2025-02-28 → 2025-08-30 (6 месяцев данных промышленного уровня)

---

item_metadata.csv

Колонка	Описание	Тип данных	Примечания
item_id	Уникальный идентификатор трека.	Integer	Первичный ключ для треков.
track_name	Название/заголовок трека.	String	Может включать специальные символы или текст не на английском.
artist_name	Имя исполнителя(ей) или артиста(ов).	String	Может включать коллаборации (например, "Artist feat. Other").
track_duration	Полная длительность трека в секундах.	Integer	Общая продолжительность трека.

track_genres_list	Список жанров, связанных с треком, разделённых запятой.	String	Пример: ['Genre1', 'Genre2']; может включать жанры не на английском.
track_dislike_count	Общее количество дизлайков, полученных треком.	Integer	Агрегированная метрика; начинается с 0 для новых треков.
track_like_count	Общее количество лайков, полученных треком.	Integer	Агрегированная метрика; начинается с 0 для новых треков.
track_download_count	Общее количество скачиваний трека.	Integer	Агрегированная метрика; начинается с 0 для новых треков.

---

user_metadata.csv

Колонка	Описание	Тип данных	Примечания
user_id	Уникальный идентификатор пользователя.	Integer	Первичный ключ для пользователей.
age_bin	Возрастная группа пользователя (на основе ML модели).	String	Примеры: '18_29', '30_44', '45_60', '61_inf'.
children	Количество детей у пользователя (на основе ML модели).	Integer	0–1
gender	Пол пользователя (на основе ML модели).	String	'M' для мужчин, 'F' для женщин.
top_genre	Предпочитаемый или определённый топ-жанр пользователя.	String	Примеры: 'Поп', 'Джаз', 'Chill'; может быть не на английском.
user_disliked_track_count	Общее количество треков, которым пользователь поставил дизлайк.	Integer	Агрегированная метрика на уровне пользователя.
user_liked_track_count	Общее количество треков, которым пользователь поставил лайк.	Integer	Агрегированная метрика на уровне пользователя.
user_downloaded_track_count	Общее количество треков, скачанных пользователем.	Integer	Агрегированная метрика на уровне пользователя.

test.csv

Колонка	Описание	Тип данных	Примечания
user_id	Уникальный идентификатор пользователя.	Integer	Все пользователи существуют в обучающих взаимодействиях; нет cold-start пользователей.

> 🧭 **Временной диапазон:** 2025-09-01 → 2025-09-15 (период взаимодействий для оценки)

Формат данных для отправки (Prediction Format)

Файл отправки должен содержать **рекомендации для каждого пользователя** в тестовом наборе, ранжированные по релевантности.

Колонка	Описание	Тип данных
id	Последовательность id.	Integer
user_id	Идентификатор пользователя (соответствует test.csv).	Integer
item_id	Идентификатор рекомендованного трека.	Integer
rank	Ранг рекомендации (начинается с 1 до 50).	Integer

Lost in the Museum

Kazakh - Lost in the Museum

Мұражайда

Сипаттама

«Мұражайда» атты жарысқа қош келдіңіз!

Сізге мұражайларға келушілер түсірген 1000 фотосурет берілген. Суреттердің сапасы, түсірілген ракурс әр-түрлі (қатты жақындатылған, жарқыраған, бұлыңғыр және нашар жарықтандырылған) болуы мүмкін. Сіздің міндетіңіз осы сапасы төмен фотосуреттердің әрқайсысы үшін музей архивтерінен алынған жоғары сапалы 10000 суреттердің коллекциясы арасында дәл сәйкестікті табу.

Фотоларға метадеректер, қолтаңбалар берілмеген. Тек пиксельдер ғана.

Бұл тапсырма берілген фотолармен архивтағы суреттер доменінің қатты сәйкессіздігі үшін күрделі болып табылады. Сондықтан сіздің модельіңіз күнделікті түсірілімнің «хаосы» мен сапасы жоғары суреттер арасындағы байланысты тиімді жолмен табуы тиіс.

Сізге барлығы **20 000** сурет берілген:

- **10000** - эталондық, студиялық сападағы музей коллекцияларынан суреттер (архив);
- **1000** - «сұраулар»: келушілер әртүрлі жағдайларда түсірген фотосуреттер (қозғалыс, жарықтар, кесу, төмен рұқсат, стандартты емес ракурстар);
- **9000** - «толтырғыштар»: басқа өнер туындыларын қамтитын немесе кескіндемеге мүлдем қатысы жоқ бөгде суреттер.

Нысандарды жіктеудің, анықтаудың немесе жұптарды қолмен таңдаудың қажеті жоқ. Оның орнына, сіз 20000 суреттің әрқайсысы үшін белгіленген ұзындықтағы бірыңғай вектор - эмбеддингті жасауыңыз керек. Эмбеддингтер келушілер түсірген және эталондық архивтегі суреттер арасындағы үлкен айырмашылыққа қарамастан, негізгі визуалды мазмұнды сенімді кодтау керек.

Сіз қандай суреттер - сұраулар, қайсысы - эталондық, қайсысы - «толтырғыштар» екенін білмейсіз. Сондықтан берілген барлық суреттерді олардың тек қана визуалды мазмұнына сүйене отырып, бірдей өңдеуіңіз қажет.

Басты мақсат - әрбір «әуесқой» фотосурет өзінің түпнұсқалық суретіне тікелей жақын, бірақ барлық басқа жұмыстардан алыс орналасқан белгілер кеңістігін құру.

Маңызды: Сіз іздеуді қолмен орындамайсыз. Сіз тек эмбеддингті жібересіз. Салыстыру және бағалау қатысушыларға қол жетімсіз жабық таңбаларды пайдалана отырып, ұйымдастырушылар жағында жүргізіледі.

Бағалау

Шешімдер Hit@3 метрикасы бойынша ұйымдастырушылар серверінде келесі түрде бағаланады.

1-қадам: Косинустық жақындық бойынша сәйкестік:

- 1000 жабық "сұраныс" суретінің әрқайсысы үшін оның эмбеддингі мен галереядағы барлық 19 000 суреттің эмбеддингі арасындағы косинустық жақындық есептеледі:
- 10 000 жоғары сапалы (HQ) суреттер
- және 9000 «толтырғыштар»

Сұраныс **q** эмбеддингі мен галерея **g** эмбеддингі арасындағы косинустық жақындық мына түрде анықталады:

$$\text{similarity}(\mathbf{q}, \mathbf{g}) = \frac{\mathbf{q} \cdot \mathbf{g}}{|\mathbf{q}|_2 \cdot |\mathbf{g}|_2}$$

мұнда:

q * g — векторлардың скаляр көбейтіндісі,

$|\mathbf{q}|_2$ және $|\mathbf{g}|_2$ — олардың L2 нормалары.

Жүйе нормаланбаған эмбеддингтерді қабылдаса да, ең жақсы нәтижелер үшін L2-нормаланған векторларды жіберген абзал.

2-қадам: Үш ең жақсы кандидатты таңдау

Әрбір сұраныс үшін барлық 19 000 сурет косинустық жақындық бойынша төмендеу ретімен сұрыпталады. Үш ең жоғары жақындық мәндері бар суреттер сәйкестік үміткерлері ретінде таңдалады.

3-қадам: Hit@3 метрикасын есептеу

Егер шын сәйкес сурет (ұйымдастырушылардың жабық маркировкасынан) берілген сұраныс үшін үш ең жақсы кандидаттың ішінде болса, онда балл беріледі. Соңғы нәтиже келесі формуламен есептелінеді:

$$\text{Hit@3} = \frac{\text{Number of queries with correct match in Top-3}}{1000}$$

Метрика 0,0-ден 1,0-ге дейінгі мәндерді қабылдайды. Неғұрлым жоғары болса, соғұрлым жақсы модель. Қатысушылар қоғамдық лидерлер тақтасында олардың Hit@3 кемуі бойынша орналасады.

Маңызды: Сайыс аяқталғаннан кейін барлық финалистерден нәтижені қайта тексеру үшін коды бар жұмыс дәптерін (kaggle notebook) тапсыру тиіс.

Файлдар

Сізге `submission.csv` атты бір CSV файлын жасап, жүктеу керек. Файлда дәл 20 000 жол болуы керек:

- дерек жиынындағы әрбір сурет үшін бір жол
- қателер, қайталаулар немесе қалдырылған жазбалар болмауы тиіс.

Әрбір жол келесі мәліметтерді қамтиды:

ID — нөмірлеу (сурет файлының атауы болуы мүмкін)

image_name — сурет файлының атауы (мысалы, `00001.png`);

Эмбеддингтер — `feature_0`, `feature_1`, ..., `feature_{D-1}`, мұнда D — сіз таңдаған эмбеддинг ұзындығы (сапа мен тиімділіктің оңтайлы тепе-теңдігі үшін 256 немесе 512 ұсынылады).

Файл пішімі:

```
ID,image_name,feature_0,feature_1,feature_2,feature_3,...,feature_D-1
00001.png,00001.png,0.1234,-0.5678,0.9101,...,0.4421
00002.png,00002.png,-0.3312,0.8876,-0.0045,...,1.2039
...
```

Барлық 20 000 сурет көрсетілуі керек — ешқандай ерекшелік, сүзгілеу немесе сұрыптау болмауы тиіс.

Russian - Lost in the Museum

Заблудившиеся в музее

Описание

Добро пожаловать в «Заблудившегося в музее» — соревнование, где компьютерное зрение сталкивается с искусством, а ваша модель превращается в самого проницательного куратора на свете.

Перед вами — 1 000 реальных фотографий знаменитых полотен, сделанных посетителями музеев: с тенями, сильным приближением, бликами, размытием и плохим освещением. Ваша задача — для каждой из этих «неказистых» фотографий найти точное соответствие среди безупречной коллекции из 10 000 высококачественных изображений картин из музейных архивов.

Никаких подписей. Никаких метаданных. Только пиксели — и ничего больше.

Это одна из самых сложных задач поиска изображений в условиях сильного несоответствия доменов: вашей модели предстоит научиться «видеть сквозь хаос» повседневной съёмки и находить связь с идеальным цифровым аналогом.

Всего в датасете 20 000 изображений:

- **10 000** — эталонные, студийного качества изображения картин из музейных коллекций;
- **1 000** — «запросы»: реальные фото, сделанные посетителями в самых разных условиях (движение, блики, обрезка, низкое разрешение, нестандартные ракурсы);
- **9 000** — «приманки»: посторонние изображения, которые могут включать другие произведения искусства или вообще не относиться к живописи.

Вам **не нужно** классифицировать, детектировать объекты или вручную подбирать пары. Вместо этого вы должны сгенерировать для **каждого** из 20 000 изображений единый вектор фиксированной длины — эмбеддинг, — который надёжно кодирует визуальное содержание, несмотря на огромную разницу между «грязной» реальностью и студийным совершенством.

Вы не будете знать, какие изображения — запросы, какие — эталонные, а какие — приманки. Все изображения следует обрабатывать одинаково, полагаясь исключительно на их визуальное содержание.

Главная цель — построить такое пространство признаков, в котором каждая «любительская» фотография оказывается в непосредственной близости от своей оригинальной картины, но далеко от всех остальных работ.

Важно: вы не выполняете поиск вручную. Вы отправляете только эмбеддинги. Сопоставление и оценка производятся на стороне организаторов с использованием закрытой разметки, недоступной участникам.

Оценка

Решения оцениваются по метрике **Hit@3**, которая рассчитывается на сервере организаторов следующим образом.

Шаг 1: Сопоставление по косинусной близости

Для каждого из 1 000 закрытых «запросных» изображений вычисляется косинусная близость между его эмбеддингом и эмбеддингами всех 19 000 изображений из галереи:

— 10 000 высококачественных (HQ) картин

— и 9 000 «приманок» (фиктивных изображений).

Косинусная близость между эмбеддингом запроса **q** и эмбеддингом изображения из галереи **g** определяется как:

$$\text{similarity}(\mathbf{q}, \mathbf{g}) = \frac{\mathbf{q} \cdot \mathbf{g}}{|\mathbf{q}|_2 \cdot |\mathbf{g}|_2}$$

где:

- **q** * **g** — скалярное произведение векторов,

- $|\mathbf{q}|_2$ и $|\mathbf{g}|_2$ — их L_2 нормы.

Хотя система принимает и оценивает и ненормализованные эмбеддинги, для наилучших результатов рекомендуется отправлять L2-нормализованные векторы.

Шаг 2: Выбор трёх лучших кандидатов

Для каждого запроса все 19 000 изображений сортируются по убыванию косинусной близости. Три изображения с наибольшими значениями близости выбираются в качестве кандидатов на соответствие.

Шаг 3: Расчёт метрики Hit@3

«Попадание» засчитывается, если истинная соответствующая картина (из закрытой разметки организаторов) оказывается среди трёх лучших кандидатов для данного запроса. Итоговый результат вычисляется по формуле:

$$\text{Hit@3} = \frac{\text{Number of queries with correct match in Top-3}}{1000}$$

Метрика принимает значения от 0.0 до 1.0. Чем выше значение — тем лучше модель справляется с поиском. Участники ранжируются в публичном лидерборде по убыванию их Hit@3.

Важно: после завершения соревнования от всех финалистов потребуется предоставить рабочий ноутбук с кодом для проверки воспроизводимости результата и исключения возможного жульничества.

Файлы

Вам необходимо сгенерировать и загрузить **один CSV-файл** с именем `submission.csv`. Файл должен содержать **ровно 20 000 строк** — по одной для каждого изображения из датасета — без пропусков, дубликатов или пропущенных записей.

Каждая строка должна включать:

- ID – нумерация (может быть имя файла изображения)

- image_name — имя файла изображения (например, `00001.png`);

- Фиксированное число признаков — `feature_0`, `feature_1`, …, `feature_{D-1}`, где **D** — выбранная вами размерность эмбеддинга (рекомендуем 256 или 512 для оптимального баланса между качеством и эффективностью).

Формат файла:

```
ID,image_name,feature_0,feature_1,feature_2,feature_3,...,feature_D-1
00001.png,00001.png,0.1234,-0.5678,0.9101,...,0.4421
00002.png,00002.png,-0.3312,0.8876,-0.0045,...,1.2039
...
```

Все **20 000 изображений** должны быть представлены — **без исключений**, фильтрации, сортировки или пропусков.