feature enginering

Отличный вопрос! **Feature Engineering (FE)** — это искусство и наука превращения сырых данных в информативные признаки, которые позволяют моделям лучше понять закономерности. В рекомендательных системах это критически важно.

Вот **всеобъемлющий список методов FE**, структурированный по типам данных и этапам работы. Держите.

---

0. Философия и общие принципы

- **Цель:** Создать признаки, которые описывают:
  1. **Объекты (items):** Что это?
  2. **Пользователей (users):** Кто это?
  3. **Контекст (context):** Где, когда, при каких условиях?
  4. **Взаимодействие (interaction):** Как пользователь и объект встретились?
- **Правило:** Любую идею нужно проверять на **валидации**. Иногда простой признак бьет сложный.

I. FE для Объектов (Items)

1. Статистики по популярности (на основе исторических данных)

- `item_popularity` : Количество взаимодействий (кликов, покупок, просмотров) с объектом **за всё время**.
- `item_popularity_recent_1d/7d/30d` : Популярность за последний день/неделю/месяц (ловит тренды, "виральность").
- `item_popularity_trend` : Отношение популярности за последнюю неделю к популярности за предыдущую. Показывает рост/падение.
- `item_popularity_std` : Стандартное отклонение популярности по временным окнам. Показывает стабильность спроса.

2. Статистики по взаимодействиям (Implicit/Explicit Feedback)

- `item_avg_rating` : Средний рейтинг объекта (если есть рейтинги).

- `item_click_through_rate (CTR)`: `clicks / impressions` (если есть данные о показах).
- `item_purchase_conversion`: `purchases / views`.
- `item_success_ratio`: `positive_interactions / total_interactions`.
- **Распределение оценок:** Количество 5-звездочных, 1-звездочных оценок.

3. Временные признаки объекта

- `item_age`: Сколько дней/месяцев объект находится в системе (с момента создания).
- `days_since_last_interaction`: Как давно с объектом последний раз кто-то взаимодействовал.
- `is_new_item`: Флаг, `item_age < threshold` (например, 7 дней).

4. Признаки на основе "контента" объекта (метаданные)

- **Категориальные:**
  - `item_category`, `item_brand`, `item_author`, `item_genre`, `item_color`.
  - **Иерархические категории:** Разбить путь "Электроника -> Смартфоны -> Apple" на несколько признаков.
- **Текстовые:**
  - **Из названия/описания:** `item_title_length`, `item_description_length`, `language`, наличие ключевых слов (`"sale"`, `"new"`).
  - **TF-IDF / Word2Vec / BERT-эмбеддинги** названия. Затем можно делать `item_price`, `item_weight`, `item_size`.
- **Числовые:**
  - `item_price`, `item_weight`, `item_size`.
  - `is_discounted`, `discount_percentage`.
  - `item_price_per_unit` (например, цена за килограмм).

II. FE для Пользователей (Users)

1. Статистики по активности

- `user_activity_count`: Общее количество взаимодействий пользователя.
- `user_activity_recent_1d/7d/30d`: Активность в разных временных окнах.
- `user_activity_frequency`: `total_interactions / user_age_in_system`.
- `user_session_count`: Количество сессий.
- `avg_session_length`: Среднее количество взаимодействий за сессию.

2. Статистики по поведенческому профилю

- `user_avg_rating`: Средняя оценка, которую ставит пользователь (строгий/добрый).
- `user_preferred_category`: Самая частая категория в его истории.
- `user_preferred_price_segment`: Медианная цена товаров, с которыми он взаимодействовал.
- `user_diversity`: Энтропия по категориям в его истории (любит одно или разное).
- `user_recency`: Дней с последнего визита.

3. Временные и сессионные паттерны

- `user_time_of_day_avg`: В какое время суток (утро/день/вечер) чаще активен.
- `user_day_of_week_avg`: В какие дни недели чаще активен (будни/выходные).
- `user_is_weekend_user`: Флаг.

4. Демографические и статические признаки

- `user_age`, `user_gender`, `user_location` (город, страна), `user_registration_date`.
- `user_age_group`, `user_income_segment` (если есть).

III. FE для Пар [Пользователь x Объект] (User-Item Interactions)

Это самое важное! Модели ранжирования учатся на этих признаках.

1. Совпадение (Matching) профилей пользователя и объекта

- `category_match`: Взаимодействовал ли пользователь раньше с этой категорией? (`1/0`).
- `preferred_category_match`: Это его любимая категория? (`1/0`).
- `price_match`: Цена объекта близка к предпочитаемому пользователем ценовому сегменту? (Разница или бинаризованный признак).
- `brand_match`: Взаимодействовал ли он с этим брендом ранее?

2. Историческое взаимодействие с данным конкретным объектом

- `user_item_interaction_count`: Сколько раз пользователь взаимодействовал с **этим** объектом.
- `user_item_last_interaction_days_ago`: Как давно было последнее взаимодействие.

- `user_item_first_interaction_days_ago`: Как давно было первое взаимодействие.
- `user_item_avg_rating`: Средняя оценка пользователя этому объекту (если несколько).

3. Историческое взаимодействие с "похожими" объектами (агрегации)

- Берем N самых похожих объектов (по эмбеддингам ALS или по контенту) и агрегируем:
  - `user_avg_rating_on_similar_items`
  - `user_interaction_count_with_similar_items`

4. Популярность объекта для "похожих" пользователей

- Берем K ближайших соседей пользователя (по коллаборативной фильтрации) и считаем:
  - `avg_popularity_of_item_among_similar_users`
  - `avg_rating_of_item_among_similar_users`

5. Ранговые признаки (которые очень любят бустинги)

- `item_popularity_rank_in_user_category`: Ранг популярности объекта внутри категории, которую любит пользователь.
- `user_activity_rank_among_all_users`: Ранг пользователя по активности.

IV. FE на основе Контекста (Context)

- **Время:** `hour_of_day`, `day_of_week`, `is_weekend`, `is_holiday`, `part_of_day` (утро/день/вечер/ночь).
- **Сессия/визит:**
  - `session_length_so_far`: Сколько действий уже совершено в текущей сессии.
  - `time_spent_in_session_so_far`.
  - `items_in_session_already_viewed`: Список или счетчик.
- **Устройство/платформа:** `platform` (ios/android/web), `screen_size`, `app_version`.
- **Геолокация:** `city`, `country`, `user_distance_to_item_store` (для e-commerce).
- **Социальный контекст:** `number_of_friends_used_item`, `trending_in_your_network`.

V. FE на основе Внешних данных (External Data)

- **Погода:** `temperature`, `is_rainy` (может влиять на рекомендации товаров, фильмов, еды).
- **Экономические индикаторы:** Курс валют, индекс потребительских настроений.
- **События:** Спортивные матчи, концерты, праздники (`is_valentines_day`, `is_black_friday`).
- **Тренды соцсетей/поиска:** Частота запросов ключевых слов в Google Trends.

VI. Продвинутые / Синтетические методы FE

1. Генерация признаков через матричные разложения (Embeddings)

- **ALS/WMF (implicit):** Обучаете модель на матрице взаимодействий, получаете эмбеддинги пользователей (`user_embedding_1...d`) и объектов (`item_embedding_1...d`).
  - Их можно использовать напрямую как **d признаков**.
  - Можно считать **косинусную близость** между эмбеддингом пользователя и объекта -> мощнейший признак `als_similarity`.
  - Можно искать ближайших соседей.
- **Node2Vec/Graph FE:** Если построить граф (пользователи-объекты), можно получить эмбеддинги вершин.

2. Агрегации с различными окнами и условиями (ленивые вычисления)

- **Скользящие окна (rolling windows):** `item_avg_rating_last_100_interactions`.
- **Экспоненциальное сглаживание:** Более свежим взаимодействиям придается больший вес.
- **Агрегации с условием:** `user_avg_rating_when_price_above_median`.

3. Целевое кодирование (Target Encoding)

- **Очень мощный, но опасный метод (риск утечки).**
- `item_avg_target`: Средний целевой показатель (например, клик или покупка) по всем взаимодействиям с объектом.
- `user_avg_target`: Средний целевой показатель по всем взаимодействиям пользователя.

- `user_category_avg_target`: Средний целевой показатель пользователя в данной категории.
- **ВАЖНО:** Рассчитывать строго на исторических данных (до момента взаимодействия), чтобы избежать data leakage. Чаще всего используют **out-of-fold** или **expanding mean** схемы.

4. Взаимодействия признаков (Feature Crosses)

- Прямое перемножение или конкатенация категориальных признаков:
  - `user_country` x `item_category` -> новый категориальный признак.
  - `part_of_day` x `day_of_week`.
- **Полиномиальные признаки:** `price^2`, `price * popularity`.
- Автоматически это умеют делать модели типа **Factorization Machines** или **Deep & Cross Networks**, но ручные crosses тоже работают.

VII. Процесс и практические советы

1. **Итеративность:** Начните с 5-10 самых очевидных признаков, обучите модель. Добавляйте группы признаков по одному и смотрите на прирост на валидации.
2. **Приоритет:** Сначала **пользователь-объект взаимодействия**, затем **статистики по объектам и пользователям**, затем контекст и внешние данные.
3. **Борьба с проклятием размерности:** Для категориальных признаков с большим количеством уникальных значений (например, `item_id`) используйте **таргет-энкодинг** или **эмбеддинги**, а не One-Hot.
4. **Работа с временем:** Всегда используйте **скользящее или расширяющееся окно** для подсчета статистик, чтобы не было утечки из будущего в прошлое.
5. **Инструменты:**
   - Для агрегаций: `pandas` (`groupby`, `transform`, `rolling`), `pyspark`.
   - Для автоматизации: библиотеки `featuretools` (для автоматического создания агрегированных признаков) и `category_encoders` (для разнообразного кодирования).
6. **Анализ важности признаков:** После обучения модели (особенно tree-based) смотрите `feature_importances_`. Это скажет, какие признаки реально работают, а какие — шум.

Главный секрет: Лучшие признаки часто имеют простую **интерпретацию**. Если вы как человек можете объяснить, почему "средний рейтинг товара среди похожих на меня пользователей" должен влиять на рекомендацию, то, скорее всего, это будет хороший признак.

Это ваш арсенал. Комбинируйте, экспериментируйте и обязательно **валидируйте**!