Примерное содержание:  
Титульный лист - 1л;  
Содержание - 1л;  
Введение и постановка задачи - 1л;  
Описание данных и их особенностей - 1л;  
Описание обработки данных и разделение на трейн и тест - 1л;  
Описание возможных решений и планируемая архитектура - 1л;  
Описание обучения - 1л;  
Описание итогового результата и если есть применения на практике - 1л;  
Заключение с выводами и планами на дальнейшее развитие - 1л;  
Список литературы - 1л.

Титульный лист - 1л;

Пояснительная записка к дипломной работе

«Рекомендательная система»

Содержание - 1л;

# Введение и постановка задачи - 1л;

Основная идея

Задача рекомендательной системы:

* отранжировать какой-то набор объектов согласно какому-то критерию
* предсказать оценку объекта

**Самый типовой случай** - персонализированное ранжирование.

2 действующих лица - пользователь и объект (статья, книга, фильм и тд), которые “взаимодействуют” друг с другом

Постановка задачи

* **u**ser - пользователь
* **i**tem - объект
*  - оценка объекта **i** пользователем **u**

Задача - найти для пользователя **u** объект **i** с максимальной оценкой **r**

**Что такое рекомендательная система**

Рекомендательная система автоматически предсказывает товары/фильмы/музыку, которые могут заинтересовать пользователя на основе:

* прошлого поведения
* похожести на других пользователей
* похожести товаров/фильмов/музыки

**Зачем нужны рекомендательные системы**

Зачем нужны бизнесу?

Значительная часть покупок/просмотров/взаимодействий с сайтом происходит благодаря алгоритмам рекомендаций:

* Netflix: **2/3** фильмов просматриваются из рекомендаций
* Google News: рекомендации генерируют на **38%** больше просмотров
* Amazon: **35%** покупок совершаются через рекомендации

Зачем нужны пользователю?

* *"Люди читают около 10 МБ материалов в день, слушают около 400 МБ в день, и просматривают 1 МБ информации каждую секунду"* - The Economist, November 2006.
* *"В 2015 потребление достигнет 74 ГБ в день"* - UCSD Study 2014.

Вокруг слишком много информации и услуг. Хорошая рекомендательная система предлагает пользователю то, что ему понравится и тем самым экономит время пользователя.

# Описание данных и их особенностей

Используем датасет онлайн библиотеки книг

Данные состоят из 3 файлов:

Items

Users

Interactions

Items содержит данные о книгах: id, название, автор, жанр, год выпуска

В базе 59 599 уникальных книг

Users содержит данные о пользователях: id, возраст, пол

В базе 142 888 пользователя

Распределение пола пользователей:

Мужчины – 91991 человек

Женщины – 44635 человек

Пол неизвестен – 6262 человек

Распределение возраста пользователей:

18-24 года – 55181 человек

25-34 года – 30143 человек

35-44 года – 17600 человек

55-64 года – 16525 человек

45-54 года – 14327 человек

65 лет и более – 8966 человек

Возраст неизвестен – 146 человек

Interactions содержит данные о взаимодействиях: id пользователя, id книги, дата начала чтения книги, прогресс чтения, поставленная оценка

Всего в датасете 1 533 078 взаимодействий, из них 262625 поставленных оценок

# Описание обработки данных

Найдем самые популярные книги для рекомендации новым пользователям, которых еще нет в базе (Холодный старт):

Добавим к датафрейму оценок признаки пользователей с помощью функции merge по полю user\_id

Добавим к получившемуся датафрейму признаки книг с помощью функции merge по полю item\_id

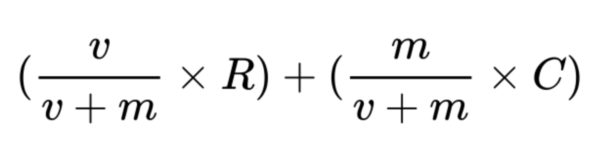
Найдем среднюю оценку, число поставленных оценок и число прочтений для каждой книги, используя функции groupby и agg

Отсортируем книги по средней оценке и посмотрим первые 10. В топ попали только книги, имеющие малое число оценок(1-3) по 5 баллов. Для рекомендаций такие книги не подходят.

Посмотрим самые читаемые книги. В топе присутствуют книги с относительно невысоким средним рейтингом(4.6), тогда как книги с рейтингом более 4.95, но имеющие меньшее число прочтений, в рекомендации не попадают.

Будем использовать взвешенный рейтинг, учитывающий как среднюю оценку книги, так и число оценок. При большом количестве оценок взвешенный рейтинг будет практически равен средней оценке книги , тогда как при малом количестве оценок он будет стремиться к среднему рейтингу по всем объектам. Принцип такой: чем больше оценок, тем больше мы можем доверять их усреднённому значению. Такой рейтинг часто применяется в различных онлайн каталогах, например IMDb (Internet Movie Database)

Взвешенный рейтинг считается по формуле:

  
Здесь:

* **v** – количество оценок
* **m** – минимальное количество голосов, необходимое для включения в рейтинг
* **R** – среднее арифметическое оценок **конкретного** объекта
* **C** – среднее арифметическое оценок **всех** объектов

Лучшие книги по взвешенному рейтингу будем использовать в качестве рекомендаций для новых пользователей. Также добавим фильтр по полу и возрасту пользователей, чтобы рекомендовать книги, популярные среди пользователей определенного пола и возраста. Выборка для некоторых возрастов может получиться недостаточной для корректных рекомендаций, поэтому разобьем возраст на 2 приблизительно равных диапазона: 18-34 года и 35+ лет. При неизвестных поле и возрасте будут рекомендоваться книги, популярные среди всех пользователей.

Описание возможных решений и планируемая архитектура - 1л;

Виды рекомендательных систем

Верхнеуровнево можно выделить два подхода:

* Не персонализированные
* Персонализированные

Персонализированные подходы можно разделить на следующие:

* Content-based filtering
* Collaborative filtering
* Hybrid

Collaborative filtering

**Основная идея** - использовать историю взаимодействий пользователей с объектами для получения векторных представлений.

Neighbour-based сollaborative filtering

**Основная идея** - использовать строки или столбцы из матрицы оценок как векторное представление пользователя или объекта.

**Два подхода:**

* Item-item - матрица схожести объектов
* User-user - матрица схожести пользователей

**Как использовать:**

* Найти похожие объекты на то, с чем пользователь взаимодействовал
* Рекомендовать объекты из тех, с которыми взаимодействовали похожие пользователи

Model-based сollaborative filtering

**Основная идея** - построить внутренние векторные представления для пользователей и объектов на основе матрицы оценок.

**Основной подход** - матричные разложения.

Двухэтапная архитектура. Фичи

После генерации кандидатов получается каркас нашего датасета:

* user\_id
* item\_id
* date
* target

На этот каркас через join’ы присоединяются фичи

Описание обучения - 1л;

Описание итогового результата и если есть применения на практике - 1л;

Метрики

Метрики в рекомендательных системах можно разделить на следующие группы

* Регрессионные
* Классификационные
* Ранжирующие

Регрессионные метрики применяются для оценки качества предсказанных моделью значений

Классификационные метрики оценивают качество топ-N рекомендаций с точки зрения бинарной классификации. Все считается на основе 4 базовых случаев:

* True positive (TP) - модель рекомендовала объект, с которым пользователь провзаимодействовал
* False positive (FP) - модель рекомендовала объект, с которым пользователь не провзаимодействовал
* True negative (TN) - модель не рекомендовала объект, с которым пользователь не провзаимодействовал
* False negative (FN) - модель не рекомендовала объект, с которым пользователь провзаимодействовал

Классификационные метрики уже неплохо показывают качество наших топ-К рекомендаций, но они учитывают только попадания. А мы также хотим, чтобы наши релевантные рекомендации находились как можно выше.

Здесь нам и помогут ранжирующие метрики, которые будут оценивать наши попадания, но с весами:

Заключение с выводами и планами на дальнейшее развитие - 1л;

Проверка качества

Два этапа тестирования:

* Offline - на ретро данных
  + Метрики: Precision@K, MAP, MRR, NDCG...
* Online - в реальном мире
  + Метрики: Conversion rate, CTR, Retention...

Основная цель - подобрать и оптимизировать такие offline метрики, которые улучшат online метрики.

Как будет использоваться модель?

* Рекомендации будут считаться раз в какой-то период
* Онлайн-рекомендации

Какие технические ограничения?

* Время на обучение и время на построение рекомендаций
* Доступность данных в онлайн-режиме

Список литературы - 1л.