

# Документация по Дипломному проекту: «Разработка рекомендательной системы»

## Оглавление

Введение .....	3
1. Бизнес-постановка задачи .....	4
1.1 Цель.....	4
1.2 Задачи .....	4
2. Техническая задача.....	5
2.1 Формат входных данных.....	5
2.1.1 Датасет "events" .....	5
2.1.2 Датасет "category_tree" .....	5
2.1.3 Датасет "item_properties".....	5
2.2 Трансформации данных.....	5
2.4 Создание факторов для модели.....	6
2.4.1 Генерация факторов, связанных с датой.....	6
2.4.2 Генерация факторов, связанных с айтемами .....	6
2.4.5 Генерация факторов юзер-айтем .....	6
2.3 Валидация .....	6
3. Проведение экспериментов .....	7
3.1 Коллаборативная фильтрация.....	7
3.2 Факторизационные машины .....	7
4. Docker.....	8
4.1 Скачивание образа из Docker Hub.....	8
4.2 Загрузка offline образа .....	8
4.3 Запуск.....	8
5. API Сервиса .....	9
5.1 Получение рекомендаций .....	9
Заключение .....	11



## Введение

Цель данного дипломного проекта заключается в разработке рекомендательной системы для интернет-магазина с целью повышения прибыли от допродаж на 20%. Проект включает в себя создание сервиса, способного предоставлять рекомендации на основе идентификатора пользователя, а также его интеграцию с главной страницей сайта.

# 1. Бизнес-постановка задачи

## 1.1 Цель

Разработать рекомендательную систему для интернет-магазина, цель которой - увеличение прибыли от допродаж на 20%.

## 1.2 Задачи

Разместить рекомендации на главной странице сайта в трех различных местах.

Создать сервис, предоставляющий рекомендации по идентификатору пользователя.

Обернуть сервис в Docker.

Написать документацию и презентацию для менеджера с описанием принципов работы.

## 2. Техническая задача

### 2.1 Формат входных данных

#### 2.1.1 Датасет "events"

#	Column	Non-Null	Count	Dtype	Описание
0	index	10000	non-null	int64	индекс
1	timestamp	10000	non-null	int64	время события
2	visitorid	10000	non-null	int64	идентификатор пользователя
3	event	10000	non-null	object	тип события (view, addtocart, transaction)
4	itemid	10000	non-null	int64	идентификатор объекта
5	transactionid	77	non-null	float64	идентификатор транзакции

#### 2.1.2 Датасет "category\_tree"

#	Column	Non-Null	Count	Dtype	Описание
0	categoryid	1669	non-null	int64	идентификатор категории
1	parentid	1644	non-null	float64	идентификатор родительской категории

#### 2.1.3 Датасет "item\_properties"

#	Column	Non-Null	Count	Dtype	Описание
0	index	10000	non-null	int64	индекс
1	timestamp	10000	non-null	int64	момент записи значения свойства
2	itemid	10000	non-null	int64	идентификатор объекта
3	property	10000	non-null	object	свойство (захешированно)
4	value	10000	non-null	object	значение свойства

### 2.2 Трансформации данных

Поскольку датасеты имеют большое количество записей и занимают большой объем оперативной памяти, пришлось ограничить их количество до 10000.

Также была выполнена оптимизация признаков следующим образом:

Признаки «event» и «property» преобразованы в категориальный тип данных. Числовые признаки преобразованы из «int64» и «float64» в меньший по объему памяти int8, int16, float32 и т.п. Это позволило уменьшить объем занимаемой памяти на 80% для «events» и на 40% для «item\_properties»

Произведено удаление дубликатов.

- Заполнение пропущенных значений и преобразование типов.
- Оптимизация факторов
- Создание списка идентификаторов посетителей.
- Разделение данных на тренировочный и валидационный датасеты.
- Удаление ненужных признаков.

## 2.4 Создание факторов для модели

### 2.4.1 Генерация факторов, связанных с датой

Из признака «date» извлечены новые признаки:

- Hour
- Month
- day\_of\_week
- is\_weekend
- is\_holiday
- time\_of\_day

### 2.4.2 Генерация факторов, связанных с айтемами

Используя матричную факторизацию с алгоритмом ALS, сгенерированы дополнительные 20 факторов, связанных с айтемами.

### 2.4.5 Генерация факторов юзер-айтем

Используя разреженную матрицу user-item, транспонированную относительно предыдущей матрицы были созданы дополнительные 20 факторов для айтемов и пользователей с использованием алгоритма ALS.

## 2.3 Валидация

После объединения датасетов «events» и «item\_properties» был создан валидационный датасет, разбив данные по времени.

Для этого были взяты последние четырнадцать дней данных в качестве валидационного периода.

Использование метрики Precision@3 для оценки качества рекомендаций.

### 3. Проведение экспериментов

Перед каждым новым экспериментом проводилась очистка памяти, путем удаления неиспользуемых переменных.

#### 3.1 Коллаборативная фильтрация

Обработанные данные были переданы в модель LightFM с параметрами:

```
cf_model = LightFM(  
    loss          = 'warp',  
    learning_rate = 0.05,  
    item_alpha    = 0.0001,  
    user_alpha    = 0.0001,  
    no_components = 60,  
    random_state  = 42)
```

И обучена с параметрами:

```
cf_model.fit(csr_user_item_train, epochs=100, num_threads=1)
```

Получено значение метрики **Precision@3 = 0.0114**

#### 3.2 Факторизационные машины

На обработанных данных построены сводные таблицы, преобразованные в разреженные матрицы взаимодействий пользователей с товарами.

На основе этих матриц была обучена модель «AlternatingLeastSquares» с 10 факторами. При подачи ID пользователя, модель выдает необходимые рекомендации.

## 4. Docker

### 4.1 Скачивание образа из Docker Hub

Для скачивания образа, необходимо в окне терминала ввести команду:

```
docker pull dmitriymakovetskiy/recsys
```

### 4.2 Загрузка offline образа

1. Скачайте файл образа по этой [ссылке](#) на локальный компьютер.
2. В окне терминала выполните команду (если архив лежит в той же директории):

```
docker load -i recsys.tar
```

3. Либо введите абсолютный путь к архиву:

```
docker load -i /полный/путь/к/recsys.tar
```

### 4.3 Запуск

Запуск осуществляется в окне терминала посредством ввода команды:

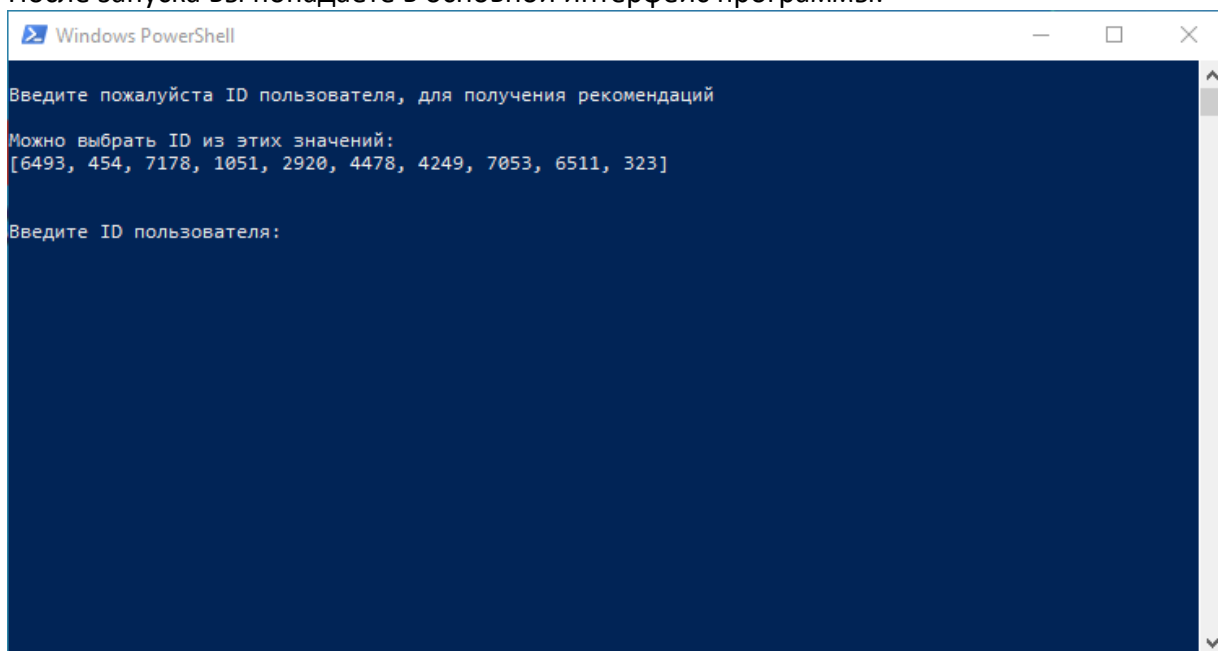
```
docker run -it dmitriymakovetskiy/recsys
```



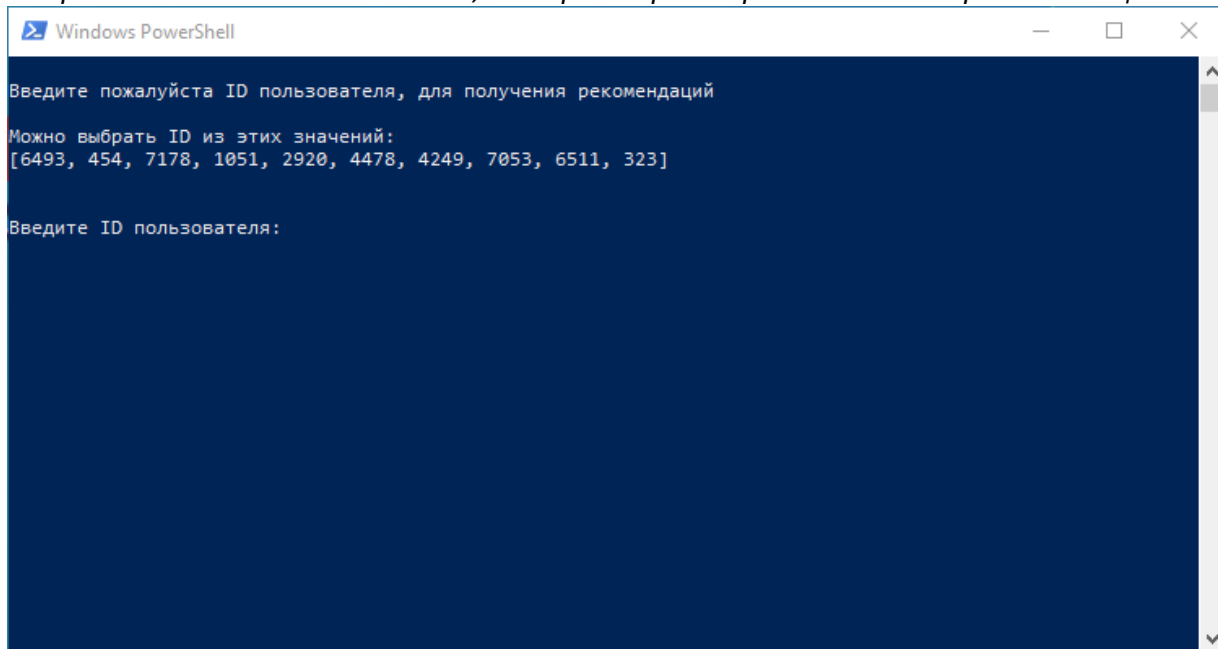
## 5. API Сервиса

### 5.1 Получение рекомендаций

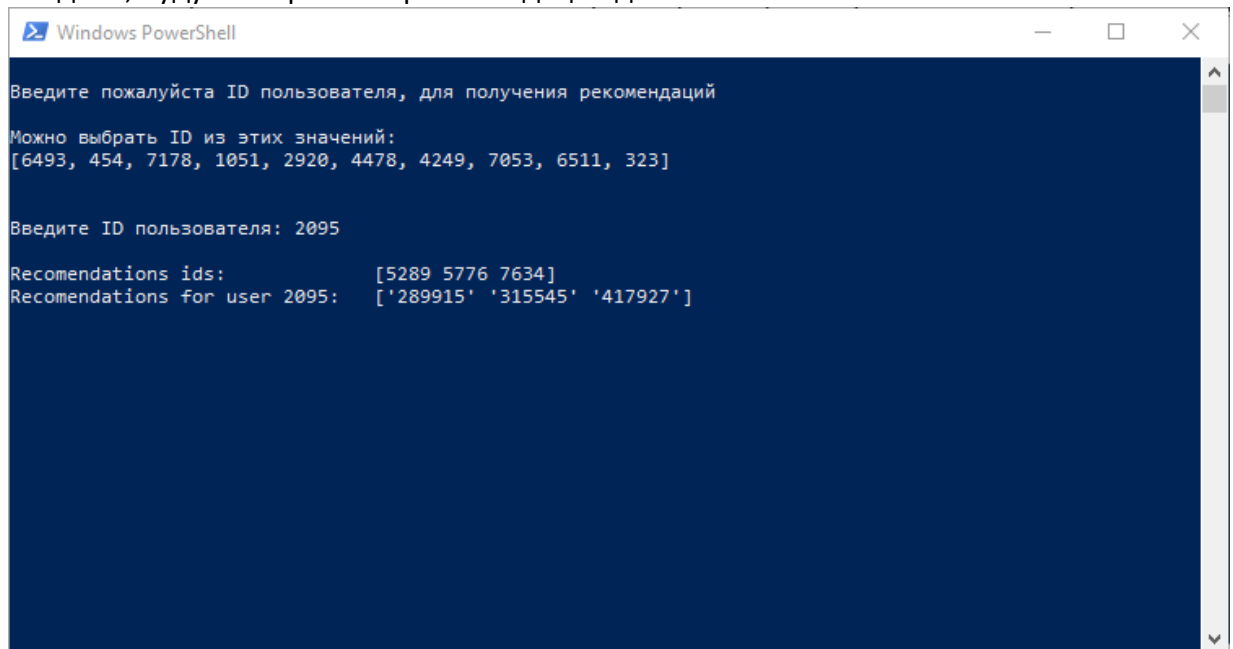
- После запуска вы попадаете в основной интерфейс программы:



- Для получения рекомендации, необходимо ввести ID пользователя.  
**Примечание.** Поскольку модель обучена лишь на части данных, будет предложено выбрать ID пользователя из тех, которые гарантированно есть в рекомендациях.



- Введя ID, будут отображены рекомендации для пользователя



```
Windows PowerShell

Введите пожалуйста ID пользователя, для получения рекомендаций

Можно выбрать ID из этих значений:
[6493, 454, 7178, 1051, 2920, 4478, 4249, 7053, 6511, 323]

Введите ID пользователя: 2095

Recomendations ids:      [5289 5776 7634]
Recomendations for user 2095: ['289915' '315545' '417927']
```

## Заключение

Данная документация предоставляет полное описание постановки задачи, технических аспектов, экспериментов и инструкций по использованию Docker и API сервиса. Она является основой для понимания и внедрения рекомендательной системы в интернет-магазин.