КУРСОВ ПРОЕКТ

Препоръчваща система за състезанието Dressipi Recsys2022

|  |
| --- |
| **Факултет по Математика и Информатика** **Студент:** Борислав Стоянов Марков **Факултетен номер:** **0MI3400048** **Учебен план:** **Изкуствен Интелект (редовно, магистър)** Курс: **Курс 1**; Група: **Група 1** **Активен период**: 2021/2022 летен, магистри **Дисциплина**: Препоръчващи системи |

# 1. Съдържание

[1. Съдържание 2](#_Toc104710675)

[2. Увод 3](#_Toc104710676)

[3 Корпус с данни 3](#_Toc104710677)

[4. Реализация с LightFM 3](#_Toc104710678)

[4.1 Алгоритъм 3](#_Toc104710679)

[4.2 Използване на приложението 3](#_Toc104710680)

[4.3 Оценка на резултатите 3](#_Toc104710681)

[5. Реализация с XXX 3](#_Toc104710682)

[5.1 Алгоритъм с XXX 3](#_Toc104710683)

[5.2 Използване на приложението с XXX 3](#_Toc104710684)

[5.3 Оценка на резултатите XXX 3](#_Toc104710685)

[6. Реализация с YYY 3](#_Toc104710686)

[6.1 Алгоритъм с YYY 3](#_Toc104710687)

[6.2 Използване на приложението с YYY 3](#_Toc104710688)

[6.3 Оценка на резултатите YYY 3](#_Toc104710689)

[4. Недостатъци и подобрения 3](#_Toc104710690)

[5. Източници и използвана литература 3](#_Toc104710691)

[Приложения 4](#_Toc104710692)

[1. Сорс код (Source code) 4](#_Toc104710693)

[2. За авторите 4](#_Toc104710694)

# 2. Увод

Проектът има за цел да се реализират получените от курса по препоръчващи системи знания. Dressipi е състезание от експерти по изкуствен интелект и участва ETH университета в Цюрих. Ние като студенти имаме възможността да сравним резултатите си със световни лидери по темата като тези резултати ще бъдат показани публично. Решихме, че екипната работа и дискусии помежду отборите ще подобри резултатите ни.

## 2.1 За състезанието Dressipi 2022

Дрессипи(Dressipi) е организация от експерти в областта на изкуствения интелект, която продава софтуер на онлайн магазини(ритейл мрежа). Те вече имат необходимия софтуер и препоръчваща система. Целта им е да подобрят софтуера, като организират подобно състезание всяка година.

Таз-годишното състезание е да се подобри препоръчващия механизъм за модни онлайн сайтове. Когато даден потребител разглежда няколко стоки, алгоритъмът на участниците трябва да предвиди акуратно какво е купил този човек накрая на потребителската си сесия. Потребителите са анонимизирани и са представени само като сесси от по максиум един ден. Съдържанието на данните са проверени от модни експерти и данните са от високо качество. На фигура 2.1.1 виждаме примерна сесия на потребител, който е разгледал няколко рокли и е купил подобна рокля.

|  |
| --- |
| Example Session and Purchase Data |

Фигура 2.1.1

Особености на задачата са:

* потребителите са 51% нови и няма исторически данни и предложените стоки трябва да се основават само на текущата сесия
* за тези потребители, за които има исторически данни, знаем, че си сменят вкуса и предпочитанията много рязко и това се обяснява със спецификата на домейна – модата.
* Сесиите са твърде кратки и трябват акуратни предложения за потребителя, преди да е напуснал сайта

Като част от задачата Дрессипи са реализирали набор от данни от 1.1 милиона ритейл сесии, които са приключили с поръчка. Всички стоки са етикирани по категории. Данните са стоките ще наричаме фийчъри на стоките. Всички данни са анонимизирани.

* Сесия: стоките, които са прегледани от потребител, максимално за един ден.
* Поръчка: Поръчка, която е станала накрая на сесията, максимално е дадена една стока за поръчка.
* Фийчър на стоката: Етикета на някой атрибут на стоката, например : „цвят:зелен“, „деколте: V-образно деколте“ и т.н.

Примерна атрибутирана стока е показана в фигура 2.1.2

|  |
| --- |
| Example Content Data |

Фигура 2.1.1

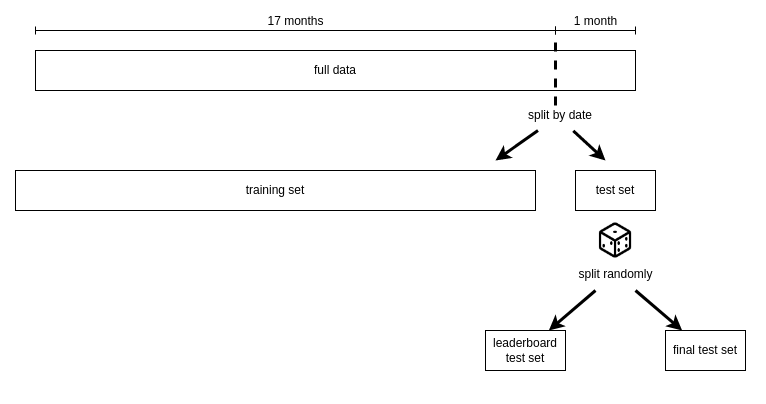
Повече информация за състезанието и правилата можете да намерите в [1]. Сега да се спрем на данните и тяхното подробно описание.

# 3 Корпус с данни

Описание на данните може е описано на сайта на Dressipi [2]. Данните са събрани от 1.1 милиона сесии от ритейл магазини за период от 18 месеца. Всички те са сесии в които има поръчка накрая. Артикулите, които се разглеждат, са облекло и обувки. Целта е да се предвиди какво е купено накрая на сесията. Най-общо можем да кажем какво означават сесия, поръчка и атрибут на стоката:

* Сесия: стоките, които са прегледани от потребител, максимално за един ден.
* Поръчка: Поръчка, която е станала накрая на сесията, максимално е дадена една стока за поръчка.
* Атрибут(фийчър) на стоката: етикета на някой атрибут на стоката, например : „цвят:зелен“, „деколте: V-образно деколте“ и т.н.

Разделянето на тестово и обучаемо множество става като тестовото множесто е от 17месеца и данните от последният месец се ползва за тестово множество, като се разделя още на две: стена на лидерите, и финални данни. Задачата е да се направят 100 предположения за всяка сесия. Тренировъчните сесии са 1 милион а тестовите „стена на лидерите“ и „финално множество“ са по 50 хиляди сесии.



Фигура 3.1

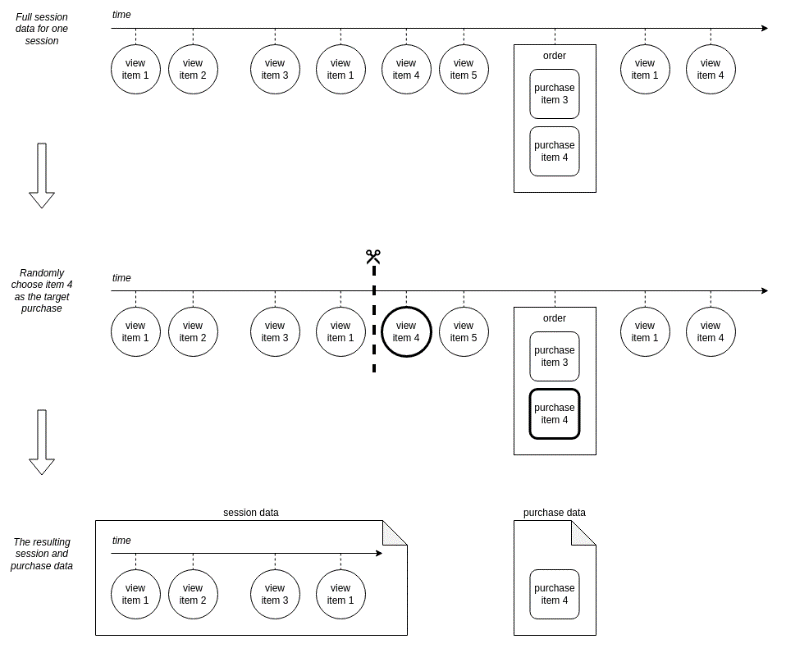
Подаването на решение се изисква във формат описан в таблица 3.2

|  |
| --- |
| session\_id,item\_id,rank  1,100,1  1,105,2  1,107,3  ...  1,101,100  2,108,1  2,107,2  ... |

Таблица 3.2

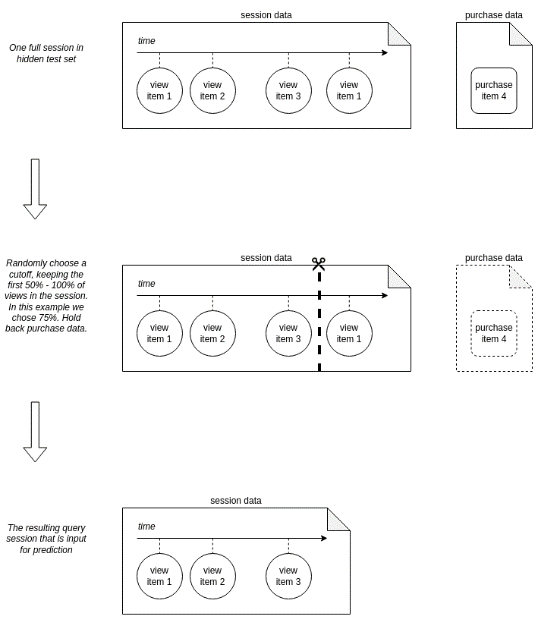
Метриката за оценяване е MRR (среден реципрочен ранк) Повече може да се прочете на [3].

Характеристиките на данните можем да кажем с няколко изречения. Данните са анонимизирани. При покупка на няколко стоки се взема предвид само първата. Много от сесиите имат само по едно две преглеждания преди покупка, но това е поради реалността и поради особеността как е конструирано множеството от данни. А именно при поръчка от няколко неща и продължаване на разглеждане, бъдещите интеракции не влизат в множеството за състезанието. А също и ако има разглеждане на артикула, който впоследствие е купен, той не влиза в данните за обучение. На следващата фигура е показана схема как се получават данните за сесиите и данните за покупките.



Фигура 3.3

Конструирането на данните за тестовите сесии се събира по следния начин. За всякак сесия се определя случаен процент х% между 50% и 100% и сесията се отрязва до там. Нарочно не се включва първият преглед на артикула, който е закупен накрая. Смисълът на това е да се предложи на потребителя стоки, които той би си купил и това предложение да е на по-ранен етап. Описаният алгоритъм за получаване на тестови сесии е показан на фигура 3.4.



Фигура 3.4

Сваляне на архива с данните за състезанието става само след регистрация. В архива, който се сваля има csv файлове и те изглеждат по този начин:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Таблица 3.5

# 4. Анализ на данните

Анализът е направен с помощта на python и Jupyter Notebook средата. Можем един по един да покажем какво представляват CSV файловете, след като ги заредим в pandas.DataFrame, използвайки функцията “df.info()”. Ще покажем само само за по-важните „train\_purchases.csv” и “ train\_sessions.csv” в Таблица 4.1

|  |
| --- |
| <class 'pandas.core.frame.DataFrame'>  RangeIndex: 1000000 entries, 0 to 999999  Data columns (total 3 columns):  # Column Non-Null Count Dtype  --- ------ -------------- -----  0 session\_id 1000000 non-null int64  1 item\_id 1000000 non-null int64  2 date 1000000 non-null datetime64[ns]  dtypes: datetime64[ns](1), int64(2)  memory usage: 22.9 MB  train\_purchases\_df.info() : None |
| <class 'pandas.core.frame.DataFrame'>  RangeIndex: 4743820 entries, 0 to 4743819  Data columns (total 3 columns):  # Column Dtype  --- ------ -----  0 session\_id int64  1 item\_id int64  2 date datetime64[ns]  dtypes: datetime64[ns](1), int64(2)  memory usage: 108.6 MB  train\_sessions\_df.info() : None |

Таблица 4.1

Сега следва да видим как са разпределени прегледите по сесии и преглежданията по артикули – таблица 4.2.

|  |  |
| --- | --- |
| а) интеракции по сесии | б) брой прегледи по артикул |

Таблица 4.2

Средно имаме 4.74 интеракции във всяка сесия. От подточка а) виждаме, че най-много сесии има с по един преглед. От подточка б) Виждаме, че имаме някои много популярни артикули, докато други са преглеждани минимално и не са търсени. Можем да видим и как са прегледите по артикули:

|  |
| --- |
| train\_sessions\_df['item\_id']**.**value\_counts()**.**describe()  count 23496.000000  mean 201.899047  std 362.441815  min 1.000000  25% 14.000000  50% 95.000000  75% 263.000000  max 14714.000000  Name: item\_id, dtype: float64 |

Таблица 4.3

Прегледите по месеци и покупките по месеци са показани в таблица 4.4.

|  |  |
| --- | --- |
| *# session views per month*  train\_sessions\_df**.**groupby([train\_sessions\_df["date"]**.**dt**.**year, train\_sessions\_df["date"]**.**dt**.**month])**.**count()**.**plot(kind**=**"bar") | *# Purchases per month*  train\_purchases\_df**.**groupby([train\_purchases\_df["date"]**.**dt**.**year, train\_purchases\_df["date"]**.**dt**.**month])**.**count()**.**plot(kind**=**"bar") |
|  |  |

Таблица 4.4

Прегледите по година и час се виждат на следващата таблица. Силните часове за 2020г и 2021г са 19,20 и 21 часа. Часовете между 9 и 17 изглеждат еднакво натоварени. Останлите часове са слаби на преглеждания.

|  |
| --- |
| *# session views per hour*  train\_sessions\_df**.**groupby([train\_sessions\_df["date"]**.**dt**.**year, train\_sessions\_df["date"]**.**dt**.**hour])**.**count()**.**plot(kind**=**"bar") |
|  |

Таблица 4.5

Анализът на данни може да се види от „notebooks/EDA.ipynb“ в Приложения.

## 4.1 Отправна точка(baseline)

Като отправна точка за всеки алгоритъм е да се сравни с някой друг и това става като например даваме случайни предложения и измерим MRR (среден реципрочен ранк). За целта е разработена специална ноутбук папка „notebooks/Baseline.ipynb“ и там е реализиран случаен алгоритъм и замерване на 5000(поради ограничената RAM памет) елемента на случайно подбрана 5% извадка от поръчките. Вижда се, че MRR граничи около 0.01, което е и минимума, който е описан в състезанието.

# 5. Реализация с LightFM

LightFM е библиотека на python за хибридни препоръчващи системи. Тя работи добре при студен старт, както имаме в текущите данни, тъй като нямаме информация за потребителите и техните вкусове. Ноутбук папката е „notebooks/lightfm.ipynb“ и там е кодът реализиращ препоръчваща система. Като цяло библиотеката работи с атрибути на артикулите(item\_features), атрибути на потребителите(user\_features) и потребителски взаимодействия(user\_interactions).

## 5.1 Алгоритъм

## 5.2 Използване на приложението

## 5.3 Оценка на резултатите

# 6. Реализация с XXX

## 6.1 Алгоритъм с XXX

## 6.2 Използване на приложението с XXX

## 6.3 Оценка на резултатите XXX

# 7. Реализация с YYY

## 7.1 Алгоритъм с YYY

## 7.2 Използване на приложението с YYY

## 7.3 Оценка на резултатите YYY

# 8. Недостатъци и подобрения

# 9. Източници и използвана литература

[1] The RecSys Challenge 2022, Dressipi, Bruce Ferwerda (Jönköping University, Sweden), Saikishore Kalloori (ETH Zürich, Switzerland), and Abhishek Srivastava (IIM Jammu, India).  
<http://www.recsyschallenge.com/2022/#participation>

[2] RecSys Challenge 2022 Dataset,   
<http://www.recsyschallenge.com/2022/dataset.html>

[3] Mean Reciprocal Rank, Wikipedia,  
<https://en.wikipedia.org/wiki/Mean_reciprocal_rank>

# Приложения

## 1. Сорс код (Source code)

Кодът е публичен и качен в платформата Гитхъб.

https://

## 2. За авторите