**Диплом НЕТОЛОГИИ**

<https://github.com/AbrMaria/Rec_Sys_PROJECT>   
  
цель, проблема, описание бизнес-процессов, архитектура решения, результаты  
  
**Проблема:**

* не понятно с чего начать/продолжить изучение DS или нового языка программирования
* изобилие ресурсов (курсов) усложняет выбор
* пользователи больше доверяют рекомендациям и оценкам других пользователей, чем рекламе самих ресурса

**Цель**

* выводить по запросу пользователя (по ключевым словам\параметрам) рекомендацию курса из базы
* сравнивать между собой курсы и подбирать похожие по содержанию
* сравнивать между собой курсы по описанию \ оценкам \ отзывам

Что пользователь получает на выходе - какую ценность и каким образом

* список курсов с наивысшими оценками и лучшими отзывами
  + после описания своих потребностей

**Описание бизнес-процессов**  
  
1. Определить что и кому рекомендуем

* что: курсы/лекции
* кому: студентам/вольнослушателям

2. Определить цель построения рекомендаций

* предложить новое (с чего начать изучение темы) - неперсонализированная рекомендация (5 лучших курсов по этой теме) - новичок
* предложить новое на базе уже изученного (дополнить, углубить и расширить знания в выбранной области) - персонализированная рекомендация (вместе с этим выбирают) - опытный
* сравнить между собой два/несколько схожих курсов и найти отличия по ключевым словам, оценкам, текстам отзывов - продвинутый

3. Определить метрики для достижения цели

* выдача 5-10 похожих

4. Определить метрики оценки системы (оффлайн и онлайн)

* переход по ссылке (?)

5. Подготовить данные

* бд курсов Coursera на Kaggle

6. Построить систему, оптимизирующую метрики

* ???

7. Определить способ отображения рекомендаций

* обернуть в бота с кнопками вопросно-ответной системы

8. Оценить качество рекомендаций

* ???

ИСТОЧНИК  
  
библиотека состоит из бд по курсам

**reviews\_by\_course.tsv**

CourseId - The course tag. This is in the URL of the course in the Coursera website. For example, in this URL, [machine-learning would be the course tag](https://www.coursera.org/learn/machine-learning).  
Review - A review in a specific course.  
Label - The rating of the course review.  
  
  
**Собрать информацию о курсах (спарсили с сайта Coursera)**

* название - сопоставить названия курсов с их Id
* раздел (область науки)
* средняя оценка
* количество оценок
* количество зарегистрированных на курс студентов
* количество рецензий
* автор\лектор
* язык
* к-во часов на изучение
* Об этом курсе - аннотация
* ПРИОБРЕТАЕМЫЕ НАВЫКИ - размеченные ключевые слова (теги)
* [Программа курса](https://www.coursera.org/learn/machine-learning#syllabus) - содержание (?)
* ранг курса - для пользователя какого уровня этот курс (новичок\базовый\продвинутый)
* “Пользователи, которым понравился этот курс, также просматривали“ - это было бы очень круто с моим результатом на выходе

**Пользовательский интерфейс - чатбот в Телеграм - вопросно-ответная система**  
  
приветствие  
получить информацию о пользователе:

* ник\имя
* отрасль\область его интересов\какая подтема интересна
* уровень продвинутости(?) в этой отрасли - новичок\опытный\продвинутый
* предложить лучший курс по его теме (лучший по оценкам и отзывам)
* спросить проходил ли этот курс, если “да”, то предложить следующий из выборки рекомендаци

на выход выдаёт

* новичку = название курса
* опытному = рекомендации - 5-10 самых похожих на тот курс, который он уже прошёл
* продвинутому = сравнение отличий и сходств по содержанию курсов

**Архитектура решения**  
  
0 часть - предобработка датасета

* разделение исходного датасета на русскоязычный и англоязычный

0.1 Дополнить базу курсов  
  
1 спарсить информацию о курсе

* спарсить описание и содержание курса в отдельную базу

2 обработать описание курса

* вытащить ключевые слова из описания

3 посчитать меры сходства и близости\похожести курсов

* TF\*IDF

**I NLP часть**  
  
1 обработать отзывы

* анализ текстов отзывов
* используя меру tf−idf, найти ключевые слова и биграмы для положительных и отрицательных отзывов
* определить настроение комментариев и сопоставить их с оценками курса (проверить т.о. их адекватность)
* построить несколько тематических моделей текстов отзывов

**II РекСис часть**  
Построение гибридной рекомендательной системы:

* Холодный старт (до 5 оценок) - рекомендуем наиболее популярные курсы
  + Популярность курса = нормированный райтинг \* нормированное количество оценок
* Теплый старт (от 5 до 10 оценок) - рекомендуем курсы на основе текстов отзывов
* Горячий старт (от 10 до бесконечности оценок) - коллаборативная фильтрация с блендингом (Item ??? based)
  + стекинг нескольких моделей
  + будем использовать данные по тем курсам, которым поставили больше 10 оценок

1 **Выбор и подготовка модели**  
  
**2 Рассмотрение различных вариантов рекомендательных систем   
(на базе библиотек Surprise, Spotlight, Implicit, Tensorrec, Lit2Vec)   
Сопоставление результатов и проверка контрольных метрик**  
  
  
  
**III пользовательский интерфейс - взаимодействие с ботом**  
  
  
[Бот](https://quip.com/VPdOADjG1Kzr)  
???????????????????????????????????????????????????????????  
Наиболее простым способом моделирования поведения чат-ботов является [конечный автомат](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82).  
<https://github.com/eternnoir/pyTelegramBotAPI>  
  
<https://groosha.gitbook.io/telegram-bot-lessons/>  
<https://groosha.gitbook.io/telegram-bot-lessons/chapter11>