

Рекомендательная система для сервиса “Московское Долголетие”.

Команда Сквозь Код

При построении рекомендательной модели помимо интересов пользователя важно было учесть также географический фактор. В таблице *users.csv* был указан адрес проживания пользователя, а в таблице *groups.csv* адрес проведения занятия. С помощью OpenStreetMap API мы разместили данные и для каждого пользователя из *users.csv* добавили параметр *district* (район проживания). Далее модифицировали разметку по районам из *groups.csv*, чтобы районы в двух таблицах совпадали. Также некоторые районы объединили в географические кластеры по принципу близости, так как многие пользователи посещают группы в соседних районах. Географический кластер включает в себя несколько соседних районов, при этом данный объект существенно меньше административного округа, так как в рамках одного округа группа и пользователь могут находиться друг от друга дальше, чем группа и пользователь в рамках соседних административных округов. Также важным параметром мы посчитали время дня, поэтому в таблицы были добавлены данные о предпочтениях времени дня пользователя на основе их посещаемости (день, утро, вечер).

По заданию в качестве рекомендации для пользователя необходимо выдавать конкретную группу, но мы решили обучить модель находить интересы пользователя, а потом уже по полученному списку интересов выдавать подходящие по географическому и временному признакам группы. В качестве *item* выступало направление занятия третьего уровня

(направление 3, далее - *dir3*) из таблицы *groups.csv*. Каждому направлению был присвоен свой уникальный номер(*item_id*). Далее была составлена таблица интересов *user-item*, где на пересечении *user* и *item* находится число - общее количество посещений занятий данного направления данным пользователем вне зависимости от группы. Таким образом таблица является матрицей рейтингов, причем она учитывает не только сам факт, но и степень интереса пользователя к данной активности.

В качестве кандидатов для обучения были выбраны алгоритмы SVD (реализация *scipy*) и ALS (реализация *implicit*). В ходе тестирования алгоритм ALS оказался сильнее, и было принято решение обучать и подбирать гиперпараметры под него. В ходе тестов лучшая модель имела следующие гиперпараметры:

- размерность векторов: 12;
- регуляризация: 0.01;
- количество итераций обучения: 142.

Для обучения использовались разреженные матрицы, из-за некоторых особенностей реализации *scipy* пришлось переопределить *user_id* для модели. Данный “маппинг” *user_id* из таблицы *users.csv* и *ml_id* для модели не влияет на общую работу сервиса. Модель принимает на вход *ml_id* пользователя и выдает вектор *item_id* направлений активностей. Данная модель уже учитывает посещаемые направления занятий и не выдаёт посещаемые пользователем.

[!] Если пользователь преимущественно посещал онлайн-занятия, то модель преимущественно выдаст направления различных онлайн-занятий. Мы реализовали функционал, который позволяет добавлять в такую подборку направления офлайн занятий, так как очный формат является приоритетным.

После получения вектора направлений активностей для пользователя подбирается подходящая группа

- с учетом его места проживания и предпочтительного времени дня для посещения занятия, если рекомендуется очный формат занятия;

- с учетом предпочтительного времени дня для посещения занятия, если рекомендуется онлайн формат занятия.

Если по заданным территориальному и временному фильтру подходит несколько групп, то рекомендованная группа выбирается из списка случайно.

[!] Оставляю важные комментарии с точки зрения продукта:

1. В идеале стоит выдавать ближайшую к адресу пользователя группу, но тогда датасет нужно обогащать дополнительными геоданными, чтобы просчитывать расстояние между точками на карте, либо обращаться к API одного из сервисов карт;

2. В погоне за метрикой `ar@k` мы могли бы выдавать не случайную группу из выборки, а самую популярную с точки зрения посещаемости, это способствовало бы увеличению метрики, но при масштабировании данной задачи на огромный сервис для людей мы таким подмешиванием будем искусственно направлять трафик на одну группу, что приведет к сильному смещению распределения пользователей по группам в сторону часто посещаемых, а остальные группы останутся ни с чем.

Также модель умеет выдавать направления занятий, похожие на заданное. Например, если пользователь просматривает группу, которая занимается английским языком, то ему на детальной странице будут предложены группы, похожие на эту по направлению занятий, например, французский язык, немецкий язык итд.

Файл [test_answers.csv](#) содержит списки групп для пользователей из *test.csv* и представлен в репозитории.

При тестировании сервиса вы можете вводить любые ФИО и дату рождения, тогда вы авторизуетесь с случайным *id* пользователя.

По рекомендации экспертов мы дополнили датасет случайными ФИО для удобства входа в аккаунт пользователя с конкретным *id*. Если вы желаете пройти авторизацию под таким пользователем, то воспользуйтесь данной ссылкой для получения ФИО и даты рождения пользователя:

<http://api.sqwozcode.ru/getUser?uid=XXXXXXXXXX>

где вместо XXXXXXXXXXXX подставьте уникальный *id* пользователя.