

# **РАЗРАБОТКА БОТ-ПРИЛОЖЕНИЯ С СИСТЕМОЙ РЕКОМЕНДАЦИЙ НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ**

**Е.А. Павлив**

*Белорусский государственный университет, г. Минск;  
pavliv.liza@yandex.ru;  
науч. рук. – И.С. Войтешенко, канд. техн. наук, доц.*

Системы рекомендаций широко используются во многих отраслях, в том числе электронной коммерции, поскольку могут увеличить доход компании и быть способом выделиться среди конкурентов. Все более распространенными становятся бот-приложения, поскольку удобны в использовании, а взаимодействие с ними напоминает общение с человеком. Бот-приложение с системой рекомендаций может быть интегрировано, например, с веб-сайтом для просмотра фильмов, что позволяет пользователю избежать необходимости листать каталоги и тратить время на сложный выбор: достаточно задать вопрос на естественном языке, и система рекомендаций подберет наиболее подходящие ему кинокартины. В данной работе описывается процесс построения модели машинного обучения для системы рекомендаций, проектирование и разработка бот-приложения с использованием Cloud-технологий, умеющего распознавать естественный язык пользователя, и интеграция бот-приложения с системой рекомендаций. Адаптированное под систему рекомендаций тренингов приложение в настоящий момент используется в системе тренингов ИООО «Эксадел» для рекомендации тренингов сотрудникам в соответствии с их предпочтениями. Данная разработка позволяет пользователю, использующему систему тренингов, в режиме реального времени получить помощь в выборе курсов или вебинаров, задав вопрос на естественном языке.

**Ключевые слова:** система рекомендаций, машинное обучение, коллаборативная фильтрация, проблема холодного старта, бот-приложение, Cloud-технологии.

## **КОЛЛАБОРАТИВНАЯ ФИЛЬТРАЦИЯ**

Существует несколько типов систем рекомендаций, предназначенных для решения различных задач. Одним из наиболее распространенных является коллаборативная фильтрация, поскольку часто информации о пользователях нет, и удобнее использовать только имеющиеся данные о продуктах. В ходе коллаборативной фильтрации анализируются действия пользователей и ищется сходство между их предпочтениями в продуктах, при этом не нужно никакой информации о самих пользователях. То есть если у пользователей, совпали вкусы в прошлом, то предполагается, что они совпадут и в будущем. Главным преимуществом метода является то, что чем больше пользователей взаимодействуют с продуктами, тем более точными становятся новые рекомендации.

Одним из алгоритмов для реализации коллаборативной фильтрации является Field-aware Factorization Machines или машины факторизации с

учетом специфики поля. Взаимодействие пользователей с фильмами представляется в виде матрицы. Каждый столбец соответствует какой-то характеристике фильма. Последний столбец – выставленный пользователем рейтинг данному фильму. Поля – это независимые переменные, например, жанр фильма, а признаки – категориальные значения, которые принимает каждое поле, например, жанр комедия или боевик. В машинах факторизации изучается взаимодействие признаков [1]. Машины факторизации с учетом специфики поля хорошо использовать, когда при рекомендации учитываются характеристики объектов, их различные признаки, например, жанры фильма.

## ПРОЦЕСС ПОСТРОЕНИЯ МОДЕЛИ В ML.NET

Для создания модели машинного обучения была использована ML.NET – среда с открытым исходным кодом, разработанная для обучения, создания и оценки моделей машинного обучения [2].

Процесс построения модели в ML.NET начинается с сбора и загрузки данных. В качестве набора данных для обучения модели для системы рекомендаций фильмов использовался набор рейтинговых данных с сайта MovieLens, которые были собраны и предоставлены исследовательской лабораторией факультета компьютерных наук и инженерии университета Миннесоты GroupLens. Набор данных о фильмах содержит 27 миллионов пятизвездочных рейтингов. Каждый пользователь представлен анонимным идентификатором. Информация о каждом фильме представляет собой идентификатор, название фильма и жанры. Каждый фильм имеет один и более жанр. Далее данные делятся на обучающий и тестовый наборы в заданном соотношении (80 % на 20%). Затем создается обучающий конвейер, в котором выполняется преобразование набора тренировочных данных, поскольку алгоритмы машинного обучения ML.NET требуют того, чтобы входные данные или объекты находились в одном числовом векторе. Последним этапом в конвейер добавляется выбранный алгоритм обучения из каталогов ML.NET (FieldAwareFactorizationMachine).

Эффективность обученной модели можно оценить по тестовым данным. Для каждого алгоритма ML.NET предоставляет различные метрики для оценки. Наиболее часто используемой метрикой является точность, показывающая долю правильных прогнозов в тестовом наборе. Полученное значение метрики в разработанной модели: 78%. Всего для оценки модели было использовано 9 метрик, дающих приблизительно одинаковые значения эффективности модели.

Далее модель можно импортировать в различные форматы, в том числе ZIP и ONNX (Open Neural Network Exchange) – открытый и совместимый стандартный формат для представления моделей машинного обучения, который позволяет сохранять обученные модели, созданные на любой платформе, и запускать их на других платформах, включая .NET.

## ПРОБЛЕМА ХОЛОДНОГО СТАРТА

Проблема холодного старта – невозможность предоставить новому пользователю точные рекомендации. Основная стратегия работы с новыми пользователями – попросить предоставить их некоторую информацию о своих предпочтениях в продукте для создания первоначального профиля пользователя: таким образом система будет иметь некоторую отправную точку для работы с ним. В разработанном приложении новому пользователю выводится список из 20 лучших фильмов в системе, ему предлагается оценить какие-нибудь из этих фильмов выставлением рейтингов по шкале от 1 до 5.

Как ранее говорилось, в ходе коллаборативной фильтрации ищутся пользователи со схожими интересами. В связи с этим возникает проблема нахождения степени «похожести» нового пользователя приложения на других пользователей из набора данных. Изначально для каждого пользователя составляется вектор предпочтений. Если пользователь оценил продукт, в вектор записывается выставленный рейтинг, если нет – ноль. Для измерения сходства векторов использовалось косинусное сходство. Чтобы найти человека с похожими предпочтениями, нужно найти косинусы между вектором рейтингов исходного пользователя и векторами пользователей из набора данных. Так как косинусное сходство определяется как косинус между двумя векторами, то результат всегда будет в диапазоне от -1 до 1, где -1 соответствует углу 180 градусов (вектора противоположно направлены) и показывает, что два пользователя не похожи, а 1 – углу 0 градусов (вектора сонаправлены), значит, два пользователя похожи. Поэтому из всех найденных косинусов выбирается максимальный.

## РАЗРАБОТКА БОТ-ПРИЛОЖЕНИЯ

Боты могут делать то же самое, что и другие типы программного обеспечения, а их ключевое отличие заключается в своеобразной имитации действий людей и использовании механизмов, обычно предназначенных для общения между людьми. Именно из-за растущей популярности, актуальности и повсеместного использования ботов, для интеграции с ранее разработанной системой рекомендаций было выбрано бот-приложение.

Для его создания были использованы сервисы Azure Bot Service и Bot Framework, которые предоставляют средства для сборки, тестирования, развертывания и управления интеллектуальными ботами в единой среде [3]. Бот взаимодействует с Bot Framework Service для отправки и получения сообщений и событий. Для расширения функционала бота использовались компоненты, предоставленные платформой облачных сервисов Azure Cloud Services. Бот-приложение развертывается с помощью Azure App Service и выполняется в Azure. В процессе работы оно отправляет данные телеметрии в сервис App Insights, которые в дальнейшем используются для анализа работы приложения. Бот использует сервис распознавания естественного языка LUIS, который применяет искусственный интеллект к тексту для прогнозирования намерения пользователя (задавались примеры фраз для каждого намерения для обучения модели). Бот-приложение интегрируется с ранее обученной моделью системы рекомендаций. Для получения и сохранения информации о предпочтениях (выставленных рейтингах) пользователя используется облачное хранилище ключей/значений Azure Table storage. Приложение загружает постеры фильмов для интерактивного отображения рекомендаций, используя MovieLens API. Пользователь взаимодействует с ботом через канал, который используется для подключения бота к различным пользовательским приложениям, таким как Web Chat, Skype и другим. Для канала Web Chat был создан виджет веб-чата, который может быть интегрирован на любой веб-сайт.

### **Библиографические ссылки**

1. *Peng Yan, Xiaocong Zhou, Yitao Duan. E-Commerce Item Recommendation Based on Field-aware Factorization Machine* // ResearchGate. URL: [https://www.researchgate.net/publication/282846395\\_E-Commerce\\_Item\\_Recommendation\\_Based\\_on\\_Field-aware\\_Factorization\\_Machine](https://www.researchgate.net/publication/282846395_E-Commerce_Item_Recommendation_Based_on_Field-aware_Factorization_Machine) (date of access: 04.10.2020).
2. What is ML.NET and how does it work? // Microsoft Docs. URL: <https://docs.microsoft.com/en-us/dotnet/machine-learning/how-does-mldotnet-work> (date of access: 25.09.2020).
3. About Azure Bot Service // Microsoft Docs. URL: <https://docs.microsoft.com/en-us/azure/bot-service/bot-service-overview-introduction?view=azure-bot-service-4.0> (date of access: 19.03.2021).