

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**РАЗРАБОТКА ПРИЛОЖЕНИЙ ДЛЯ ЭЛЕКТРОННОЙ КОММЕРЦИИ
С ПРИМЕНЕНИЕМ МАШИННОГО ОБУЧЕНИЯ И
ОБРАБОТКИ БОЛЬШИХ ДАННЫХ**

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 273 группы

направления 01.04.02 Прикладная математика и информатика

факультета КНиИТ

Гасанова Орхана Имран оглы

Научный руководитель:

Д.т.н, профессор _____ А.В. Фалькович

подпись, дата

Зав. кафедрой:

К.ф.-м.н., доцент _____ М.В. Огнева

подпись, дата

Саратов 2019

ВВЕДЕНИЕ

Электронная коммерция в 21-м веке стала одной из самых быстрорастущих отраслей. Успех электронной коммерции включает в себя получение большого количества данных и принятия лучших решений на основе этих наборов данных. Конечная задача состоит в том, чтобы понять все эти данные, и машинное обучение помогает достичь этой цели.

Машинное обучение - это применение искусственного интеллекта, который предоставляет системам возможность автоматически учиться и улучшаться из опыта, не будучи явно запрограммированным. Машинное обучение базируется на разработке компьютерных программ, которые получают доступ к данным и используют их для изучения. Основная цель состоит в том, чтобы позволить компьютерам учиться автоматически без вмешательства человека и соответствующим образом корректировать действия.

Процессы, связанные с машинным обучением, аналогичны процессам интеллектуального анализа данных и прогнозирования. Оба требуют поиска данных, чтобы искать шаблоны и соответственно корректировать действия программы. Примером может послужить покупка в интернете товаров, и объявление которое система показывает клиенту во время покупки. Это происходит потому, что механизмы рекомендаций и объявлений используют машинное обучение для персонализации доставки и делают это в режиме реального времени. Помимо персонализированного маркетинга, другие распространенные случаи использования машинного обучения включают обнаружение мошенничества, фильтрацию спама, обнаружение угрозы сетевой безопасности, интеллектуальное обслуживание и создание новостных лент.

Цель магистерской работы – Целью магистерской работы является создание платформы электронной коммерции, в которую интегрированы машинное обучение и обработка больших данных.

Поставленная цель определила **следующие задачи:**

1. Обзор литературы по машинному обучению;
2. Обзор областей электронной коммерции в которых наиболее востребовано применение машинного обучения;
3. Исследование и нахождение наиболее подходящего набора данных для чатботов используемых в электронной коммерции;
4. Структурирование выбранного набора и сохранение в базу данных для дальнейшей тренировки модели;
5. Обзор алгоритмов машинного обучения и выбор наиболее подходящего для электронной коммерции;
6. Тренировка модели наборами данных выбранными ранее;
7. Обзор фреймворков и выбор наиболее оптимального для разработки фронтенд модуля приложения;
8. Разработка модуля mycart-be;
9. Разработка модуля mycart-fe;
10. Исследование технологий для учета количества товаров и выбор наиболее оптимального;
11. Разработка приложения mycart-kafka-connect;
12. Развертывание и тестирование kafka connect с использованием коннекторов.

Методологические основы разработки приложений для электронной коммерции с применением машинного обучения и обработки больших данных представлены в работах Алпайдин Э. , Флах П., Narkhede N., Shapira G., Palino T. , Kleppmann M, Kumar M., Singh C., Wieruch R., Bertoli M., Arora H., Kashyap P.

Теоретическая значимость магистерской работы

В работе описаны проблемы, с которыми сталкиваются платформы электронной коммерции в современном мире, и даны теоретические основы для решения данных проблем. Выделены 2 основные проблемы: проблема обработки больших данных и автоматизации процесса общения с клиентом. Для решения вышеуказанных проблем предложены брокер сообщений для обработки данных в режиме реального времени и машинное обучение. Был проведён подробный обзор и сравнения существующих технологий, и выбор наиболее оптимального для практического применения. Теоретический материал представленный в работе, может быть взят за основу для решения подобных проблем в платформах электронной коммерции.

Практическая значимость магистерской работы

В работе представлены детальные шаги разработки приложений для решения проблем в платформе электронной коммерции. Первым приложением является чатбот Liya. Данной чатбот обучен на данных со службы поддержки twitter. Приложение служит для автоматизации процесса общения с клиентом. Второе приложение mycart-kafka-connect, которое наряду с приложением kafka-connect-jdbc (приложение с открытым исходным кодом) служит для учёта количества товаров на складе. Информация о поступлении новых товаров на склады в других локациях в режиме реального времени поступает в центральный склад платформы электронной коммерции. Также в рамках магистерской работы была разработана платформа электронной коммерции MyCart. Разработка данного приложения была разделена на две части; на фронтенд часть приложения mycart-fe и бэкенд часть mycart-be. Приложения Liya и mycart-kafka-connect наряду с kafka-connect-jdbc были интегрированы в платформу электронной коммерции MyCart. Разработка и внедрение данных

приложений в платформу электронной коммерции уменьшает затраты бизнеса за счёт автоматизации процесса общения, и увеличивает в несколько раз увеличивает продажи за счёт поступления информации о новых товарах в режиме реального времени. Данные приложения могут быть взяты за основу при проектировании платформ электронной коммерции, а также внедрении в уже существующие для увеличения продаж и уменьшения затрат бизнеса.

Структура и объём работы. Магистерская работа состоит из введения, 3-х разделов, заключения, списка использованных источников и исходного кода приложений. Общий объём работы – 133 страниц, из них 64 страниц – основное содержание, 7 страниц - список использованных источников, 62 страниц – исходный код приложений.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «LIYA-разработка интерактивного чатбота» посвящен рассмотрению областей электронной коммерции в которых наиболее востребовано применение машинного обучения, а также разработке чатбота. Разработка чатбота начинается с исследования набора данных. Были исследованы около 50 наборов данных. 10 из них были выбраны как наиболее подходящие для электронной коммерции. Учитывая специфику основного приложения, которая служит платформой для продажи товаров разного характера, набор данных службы поддержки Twitter был выбран в качестве набора для дальнейшего исследования. Следующий этап создания чатбота включает в себя обзор нейронных сетей и выбор наиболее подходящей для данного чатбота. Были рассмотрены несколько нейронных сетей. Рекуррентная нейронная сеть (RNN) была выбрана в качестве нейронной сети для тренировки модели.

В качестве реализации Рекуррентной нейронной сети (RNN) была выбрана реализация фреймворка Tensorflow, которая используется для перевода с разных языков. Данная реализация при переводе, например, с Русского на Английский, переводит входной текст в нейтральное состояние, а затем на основании этого состояния формирует выходное значение. С учетом этого факта данная реализация подходит идеально для тренировки чатботов, где в данном случае входным параметром служит запрос, а выходным – ответ на данный запрос. Перед тем как применить выбранную нейронную сеть для тренировки, данные нужно подготовить для нее. Для этой цели проводится очистка данных и сохранение в MySQL базу данных локально. Затем данные переносятся на Cloud SQL базу данных на платформе Google Cloud. Из очищенных данных формируются 2 файла request.csv и response.csv, которые используются для тренировки модели. В качестве среды для обучения выбрана облачная машина, продукт Google Cloud Platform, Cloud ML с предустановленным фреймворком Tensorflow.

Второй раздел «MYCART-платформа электронной коммерции» посвящен разработке платформы электронной коммерции. Разработка приложения разделена на три основные части. Первая часть включает в себя разработку схемы базы данных для приложения. Следующим этапом служит разработка бэкенд-части приложения MyCart-BE, которая написана на языке программирования Java с использованием фреймворка Spring Boot.

Основная задача MyCart-BE – создание REST API, которая служит для осуществления CRUD операций в базе данных. REST API разработан с использованием внутренней возможности фреймворка, никаких сторонних библиотек при разработке для данной цели не использовалось. Ресурсы REST API созданы с учетом требований фронтенд-части приложения, а также общих требования для построения REST API. Уровень зрелости REST API в данном случае равен трем. В качестве ОРП используется Hibernate. При разработке данного модуля соблюдалась классическая трехслойная архитектура веб-приложений. Классы подразделены на Контроллеры, Сервисы, Репозитории и классы сущности, которые используются ОРП Hibernate.

Фронтенд-модуль MyCart-FE разработан с использованием фреймворка React и библиотеки Redux. React имеет компонентную архитектуру. Каждая страница состоит из одного или нескольких компонентов. Разработка MYCART-FE начинается с создания Store, которая хранит в себе состояния всех компонентов, затем рендерятся основные компоненты приложения в «root» тэге index.html файла.

Для создания каждой страницы в MyCart-BE использовалось несколько компонентов, состояния которых управляются с использованием библиотеки Redux.

Преимуществом использования фреймворков React для фронтенд-части, и Spring Boot для бэкенд-части приложения является их относительная легковесность и простота в использовании. Еще одним важным преимуществом Spring Boot фреймворка является масштабируемость с использованием внутренних возможностей фреймворка.

Третий раздел «MYCART-KAFKA-CONNECT-приложения для учета количества товаров». Одной из основных проблем в электронной коммерции является обработка больших данных, поступающих в режиме реального времени. В данной главе рассматриваются масштабируемые брокеры сообщений, которые могут быть использованы для этой цели. В качестве брокера выбирается технология kafka. Также в данной главе представлены детальные шаги разработки приложения mycart-kafka-connect, которая наряду с приложением kafka-connect-jdbc используется для учета количества товаров на складе.

Существуют два типа коннекторов Source и Sink. Source используется для получения сообщения из внешних систем и отправке в Kafka. Sink в свою очередь используется для получения из Kafka сообщений и сохранения в определенное хранилище. В данном случае разрабатывается Source коннектор который по REST запросу получает данные из внешних складов и отправляет в Kafka. Данное приложение написано на языке программирования Java.

Разработка начинается с создания коннектора MyCartSourceConnector который наследуется от SourceConnector, и добавлением поля SourceConfig config который используется для конфигурации коннектора.

Помимо разработки коннектора еще необходимо разработать трансформер для каждого URL. В данном случае был разработан MyCartJsonTransformer, который трансформирует данные в формат avro, а также добавляет время получения сообщения.

Для трансформации данных в формат avro необходимо в файле формата .avsc прописать все необходимые поля. При сборке приложения из этого файла создается класс со всеми необходимыми полями, который далее используется для трансформирования данных в формат avro.

Для тестирования приложения были написаны юнит тесты. Была проверена логика коннектора, получения и трансформации сообщения.

Для создания коннекторов отправляются POST запросы на создания 2-х коннекторов mycart-kafka-connect и kafka-jdbc-connect. После создания

коннекторов Kafka Connect данные из заданных складов копируются в базу данных MySQL центрального склада.

ЗАКЛЮЧЕНИЕ

Результатом магистерской работы является разработка четырех отдельных приложений MYCART-FE, MYCART-BE, MYCART-KAFKA-CONNECT, LIYA. В ходе выполнения магистерской работы были рассмотрены области электронной коммерции, в которой уже применяется, или может быть применено машинное обучение. После обзора было выбрано создание чатбота в качестве искусственного интеллекта, который будет отвечать запросам клиентов. Также было решено интегрировать данный чатбот с платформой электронной коммерции. Разработка чатбота включила в себя такие этапы, как выбор набора данных и их структурирование и сохранение. Данные были сохранены локально, затем резервная копия была мигрирована в облако (Google Cloud Platform). Также были рассмотрены нейронные сети, и выбрана подходящая для разработки чатбота. Модель была тренирована с использованием облачной машины Cloud ML (Google Cloud Platform). Также были разработаны фронтенд и бэкэнд части приложения MyCart с использованием фреймворков React и Spring. Также было разработано отдельное приложение mycart-kafka-connect, который служит для загрузки данных из любого API с помощью REST запросов в Кафку. Данное приложение совместно с приложением kafka-connect-jdbc (готовое приложение с открытым исходным кодом) служит для ведения учетов остатков в электронном магазине. Магистерская работа была представлена на студенческой научной конференции факультета КНиИТ.

Основная часть магистерской работы была представлена на студенческой научной конференции факультета КниИТ.

Основные источники информации:

- 1) Алпайдин Э. Машинное обучение: новый искусственный интеллект : Пер. с англ. — М. : Издательская группа «Точка», Альпина Паблишер, 2017. — 208 с.
- 2) Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных/ пер. с англ. А. А. Слинкина. - М.: ДМК Пресс, 2015. - 400 с.
- 3) Narkhede N., Shapira G., Palino T. Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale – O'Reilly Media, 2017. – p.340.
- 4) Kleppmann M, Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems – O'Reilly Media, 2017. – p.142.
- 5) Kumar M., Singh C. Building Data Streaming Applications with Apache Kafka – Packt Publishing, 2017. – p.230.
- 6) Wieruch R., Learning React: Functional Web Development with React and Redux – O'Reilly Media, 2017. – p.125.
- 7) Bertoli M. React Design Patterns and Best Practices – Packt Publishing, 2017. – p.93.
- 8) Arora H, Kashyap P. Putting Apache Kafka to Use – O'Reilly Media, 2019. – p.140.