МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАЗРАБОТКА СИСТЕМЫ ДЛЯ КЛАССИФИКАЦИИ ПРОЕКТОВ GITHUB ПО ПОПУЛЯРНОСТИ

КУРСОВАЯ РАБОТА
по дисциплине «Программная инженерия»
ЮУрГУ – 09.03.04.2024.308-570.ВКР

Нормоконтролер,
доктор физ.-мат. наук
_____ М.Л. Цымблер
“____”_____ 2024 г.

Научный руководитель
доктор физ.-мат. наук
_____ М.Л. Цымблер

Автор работы,
студент группы КЭ-303
_____ Д.И. Гольденберг

Работа защищена с оценкой
_____
“____”_____ 2024 г.

Челябинск-2024

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

УТВЕРЖДАЮ
Зав. кафедрой СП
_____ Л.Б. Соколинский
06.02.2024

ЗАДАНИЕ
на выполнение выпускной квалификационной работы бакалавра
студентке группы КЭ-303
Гольденберг Дарье Игоревне,
обучающейся по направлению 09.03.04 «Программная инженерия»

1. **Тема работы**
   Разработка системы для классификации проектов GitHub по популярности.
2. **Срок сдачи студентом законченной работы:** 05.06.2024.
3. **Исходные данные к работе**
   3.1. Milovidov A., 2020. Everything You Ever Wanted To Know About GitHub (But Were Afraid To Ask), https://ghe.clickhouse.tech/
4. **Перечень подлежащих разработке вопросов**
   4.1. Провести анализ и спецификацию предметной области.
   4.2. Изучить статистику проектов, репозитории которых расположены на GitHub.
5. **Дата выдачи задания:** 09.02.2024.

Научный руководитель
к.ф.-м.н., доцент М.Л. Цымблер

Задание принял к исполнению Д.И. Гольденберг

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Актуальность темы

Актуальность работы заключается в том, что с развитием технологий тяжело понять, насколько может быть популярен проект, такая система может быть полезна для дальнейшей работы с репозиторием.

Цель и задачи исследования

Целью данной работы является разработка системы прогноза популярности проектов GitHub. В качестве исходных данных используются датасет ClickHouse о репозиториях на GitHub. Для достижения поставленной цели необходимо было решить следующие *задачи*:

1) провести анализ предметной области и обзор научной литературы по тематике исследования;
2) разработать алгоритм обработки данных;
3)

Структура и объем работы

Курсовая работа состоит из введения, пяти разделов, заключения и списка литературы. Объем работы составляет n страниц, объем библиографического списка составляет – k наименований.

Содержание работы

Первый раздел, «Обзор работ по тематике исследования», содержит обзор на работы по тематике исследования.

Второй раздел, «Анализ предметной области», описывает постановку задачи и описание данных, которые будут использоваться для анализа.

Третий раздел, «Проектирование», определяет требования к системе, описаны модели данных и структура приложения.

Четвертый раздел, «Реализация», описывает реализацию компонентов системы.

Пятый раздел, «Тестирование», описывает функциональное тестирование работы и эксперименты с разработанными моделями данных.

В заключении приведены основные итоги проделанной работы.

1. ОБЗОР РАБОТ ПО ТЕМАТИКЕ ИССЛЕДОВАНИЯ

Исследования, связанные с анализом данных, возникающих при использовании GitHub и разработке продуктов на этой платформе, представляют собой популярную тему для научных исследований.

Так, авторы статьи [1] обсуждают проблему классификации отзывов о приложениях, которые содержат важную информацию о потребностях пользователей. Исследователи предлагают подход для создания более обобщенной модели, используя информацию из системы отслеживания задач GitHub, которая содержит ценные данные о потребностях пользователей. После проведения экспериментов, они показывают, что использование размеченных задач из GitHub может улучшить точность и полноту классификации отзывов, особенно для отчетов о багах и запросов на новые функции. В другой статье [10] поднимается важность репозиториев программного обеспечения для управления проектами, включая исходный код, документацию и отчеты об ошибках. Особое внимание уделяется платформе GitHub, которая для помощи разработчикам в поиске подходящих артефактов использует темы (topics), являющимися короткими текстами, присваиваемыми хранимым артефактам. Однако неправильное присвоение тем может негативно сказаться на популярности репозитория.

Закария Альшара и другие авторы в своей статье [2] рассматривают проблему управления задачами (issues) на платформе GitHub, особенно в случае быстрого роста числа создаваемых задач. Для помощи разработчикам в обработке задач существуют внешние участники, которые исправляют задачи, создавая pull-запросы (Pull Requests, они же PR). Однако часто такие PR не связываются с соответствующими задачами (issues), что затрудняет управление проектом. В статье предлагается использование моделей машинного обучения (ML) для автоматического восстановления связей между PR и задачами на GitHub. Установление связей между PR и задачами ценно, так как это помогает улучшить управление разработкой и обслуживанием проектов, что влияет на популярность и развитие проекта в дальнейшем.

В исследовании [9] исследует, как проводится кодирование в области науки о данных на GitHub. Авторы анализируют, как данные ученые пере-

ходят между разными этапами работы с данными. Результаты исследования показывают, что кодирование имеет определенные паттерны. Кроме того, авторы попытались обучить модели машинного обучения для предсказания этапов работы с данными и достигли точности примерно 71%.

О популярных проектах говорят в статье Джесси Айала и ее коллеги [3], а именно о важности использования непрерывной интеграции и поставки (CI/CD) и политики безопасности в известных и пользующихся интересом проектах с открытым исходным кодом, особенно на GitHub. Исследование показало, что многие проекты не активно используют эти возможности, и призывает управляющих таких проектов уделить им больше внимания для предотвращения уязвимостей. А в другом исследовании [8] фокусируются на том, как документируется информация о функциональных возможностях программного обеспечения в проектах на GitHub и связана ли она с исходным кодом. Авторы провели анализ 25 популярных репозиториев на GitHub и обнаружили, что хотя документация о функциональности часто присутствует в различных текстовых файлах, она часто неструктурированна, и связь с исходным кодом редко устанавливается, что может привести к затруднениям в его поддержке на долгосрочной перспективе.

Кроме того, существует статья [5], рассматривающая инструмент GitcProc, который предназначен для анализа проектов на GitHub. Этот инструмент позволяет извлекать информацию о разработке, включая исходный код и историю исправления ошибок. GitcProc может отслеживать изменения в исходном коде и связывать их с функциями с минимальными настройками. Он успешно работает с проектами на разных языках программирования, обнаруживая исправления ошибок и контекст изменений в коде. Помимо этого, стоит также рассмотреть следующую работу[11], где авторы обсуждают задачу классификации репозиториев на GitHub, которая представляет собой сложную задачу. Они представляют алгоритм ClassifyHub, основанный на методах ансамблирования, разработанный для соревнования InformatiCup 2017. Этот алгоритм успешно решает задачу классификации с высокой точностью и полнотой, что может быть полезно для различных приложений, таких как рекомендательные системы.

Самой приближенной к поставленной задаче статьей является «A Cross-Repository Model for Predicting Popularity in GitHub» [4], в которой рассматривается создание модели для прогнозирования популярности репозиториев на GitHub, используя данные из разных репозиториев. Модель, основанная на рекуррентной нейронной сети LSTM, позволяет более точно предсказывать популярность, чем стандартные методы прогнозирования временных рядов на основе данных из одного репозитория. Кроме неё, есть также исследование [6], в котором предлагают метод для прогнозирования популярности проектов на GitHub. Он использует 35 признаков, извлеченных из GitHub и Stack Overflow, чтобы классифицировать проекты как популярные или нет. Модель, основанная на случайном лесе, достигает высокой точности значительно превосходит существующие методы. Основными признаками для определения популярности оказались количество веток, количество открытых задач и количество участников проекта.

2. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

3. ПРОЕКТИРОВАНИЕ

В разделе 3.1. представлены функциональные и нефункциональные требования к системе. В разделе 3.2. описывается обработка данных. В разделе 3.3. представлено описание прогнозных моделей. В разделе 3.4. представлен описан модуль прогнозирования.

3.1. Требования к системе

Перечисление требований в зависимости от дальнейших выбранных библиотек и интеграций.

3.2. Обработка данных

Для создания классификации проектов на GitHub необходимо понять, по каким критериям мы сможем выстроить и определить «рейтинг» популярных репозиториев. Для работы с информацией о проектах на GitHub будет использоваться существующий датасет, загруженный в ClickHouse, кроме того, он содержит достаточно много данных, которые могут быть полезны и иметь достаточно большой вес в возможности классифицировать репозиторий. Выделим основные параметры:

- Звезды (Stars). Количество звезд репозитория указывает на его популярность. Звезды представляют интерес и поддержку сообщества разработчиков и пользователей. Большое количество звезд может привлечь внимание новых разработчиков и повысить репутацию репозитория.

- Форки (Forks). Количество форков показывает, сколько раз репозиторий был скопирован другими разработчиками для дальнейшей работы. Большое количество форков может означать активное участие сообщества, что не редко приводит к активному развитию.

- Проблемы (Issues). Количество задач отражает активность сообщества в обнаружении и решении проблем и задач. Активные задачи могут привлечь новых участников и улучшить качество проекта.

- Коммиты (Commits). Количество коммитов указывает на активность разработчиков в репозитории. Активность и последовательность коммитов важны для развития и поддержания проекта.

- Пулл-реквесты (Pull Requests). Количество запросов на слияние (pull requests) отражает вклад и сотрудничество участников проекта. Пулл-реквесты представляют собой важный инструмент совместной разработки и улучшения кода.

Текущий датасет содержит множество полей, где каждая запись соответствует определенной действия, выполненной пользователем в одном из репозиториев. В частности, существует столбец «event_type,» который описывает различные операции. Для данной работы нас интересуют следующие операции: «CommitCommentEvent» (добавление коммита с комментарием), «ForkEvent» (создание форка репозитория), «IssuesEvent» (добавление обсуждения), «PullRequestEvent» (создание запроса на включение изменений), и «WatchEvent» (добавление звезды к проекту). В контексте работы с проектами, необходимо сгруппировать выполненные операции для каждого проекта. Следовательно, с использованием запросов можно получить информацию о названии проекта (репозитория) и количестве звезд, форков и других характеристик, связанных с данным репозиторием.

После установления ключевых параметров, необходимо определить цель использования этих данных и методы их получения. В рамках данного исследования, нацеленного на разработку системы классификации проектов на платформе GitHub в зависимости от их популярности, имеется неотъемлемая потребность в доступе к данным из базы данных. Эти данные из датасета GitHub будут использованы для обучения модели классификации, её оценки и настройки. Оценка модели позволит установить, насколько успешно она способна разделять проекты на популярные и непопулярные.

Исходными данными для этой работы служит статья о наборе данных GitHub [7], содержащем информацию о всех событиях на этой платформе с 2011 года и насчитывающем более трех миллиардов записей. Данный набор данных был загружен в ClickHouse, который представляет собой открытую систему управления базами данных, специально разработанную для эффективного анализа и хранения больших объемов информации. Одной из важных особенностей этой технологии является её акцент на аналитических задачах, что обеспечивает возможность проведения слож-

ных анализов данных.

Текущий набор данных GH Archive представлен как в формате ClickHouse Native с объемом более 70 ГБ, так и в альтернативном формате, разделенном табуляцией, с объемом 85 ГБ. Учитывая, что данные из этого набора нужны лишь для однократного обучения модели без необходимости долгосрочного доступа ко всей базе данных, было принято решение извлекать такие объемы данных из облачных хранилищ, а не с физических устройств.

ClickHouse позволяет сохранять и получать данные из облачного хранилища ClickHouse.Cloud, однако на данный момент существуют ограничения, которые мешают получить доступ к этому ресурсу из-за территориальной основы. Следовательно, для продолжения работы отсутствует возможность воспользоваться данным сервисом.

Помимо вышеописанных методов извлечения данных, для демонстрации работы с общедоступными данными ClickHouse существует веб-страница, способная обрабатывать SELECT-запросы и предоставлять обширный объем информации. Все данные, полученные таким образом, являются полными и, следовательно, было принято решение осуществлять извлечение результатов запросов из HTML-страницы.

Для этой цели был разработан метод «parsingHTML», который позволяет извлекать данные из разметочных файлов. Этот метод принимает параметр «limit», который определяет желаемый объем данных для извлечения. Данный метод имитирует взаимодействие с веб-страницей, используя библиотеку Selenium WebDriver для управления браузером. Он создает виртуальное окружение браузера, отправляет запрос. После ожидания получения результата в течение 10 секунд с использованием библиотеки Beautiful Soup, предназначенной для анализа HTML-файлов, данные извлекаются из таблицы на веб-странице. Полученные данные могут быть представлены в формате CSV для дальнейшей обработки.

3.3. Прогнозные модели

Для более точного и достоверного результата необходимо правильно сформировать модели для дальнейшего анализа.

3.4. Модуль прогнозирования

Выбор наиболее подходящего алгоритма для прогнозирования в зависимости от приоритетной потребности.

4. РЕАЛИЗАЦИЯ

Для разработки системы были использованы N-ые средства и язык программирования Python версии k. Также для работы были выбраны такие библиотеки, как L и как M. Перечисление и описание методов в системе.

5. ТЕСТИРОВАНИЕ

Тестирование разработанных методов и пр, а также “эксперименты” для проверки прогнозирования.

ЗАКЛЮЧЕНИЕ

Перечисление выполненных задач и дальнейших работ в качестве улучшений.

ЛИТЕРАТУРА

1. Abedini Y., Heydarnoori A. Can GitHub Issues Help in the App Review Classifications?. // CoRR. – 2023. – Vol. abs/2308.14211. – arXiv : 2308.14211.

2. Alshara Z. ML-Augmented Automation for Recovering Links Between Pull-Requests and Issues on GitHub. / Z. Alshara, H.E. Salman, A. Shatnawi, A. Seriai. // IEEE Access. – 2023. – Vol. 11. – P. 5596–5608. – URL: https://doi.org/10.1109/ACCESS.2023.3236392.

3. Ayala J., Garcia J. An Empirical Study on Workflows and Security Policies in Popular GitHub Repositories. // 1st IEEE/ACM International Workshop on Software Vulnerability, SVM@ICSE 2023, Melbourne, Australia, May 20, 2023. – IEEE, 2023. – P. 6–9. – URL: https://doi.org/10.1109/SVM59160.2023.00006.

4. Bidoki N.H. A Cross-Repository Model for Predicting Popularity in GitHub. / N.H. Bidoki, G. Sukthankar, H. Keathley, I. Garibay. // CoRR. – 2019. – Vol. abs/1902.05216. – arXiv : 1902.05216.

5. Casalnuovo C. GitcProc: a tool for processing and classifying GitHub commits. / C. Casalnuovo, Y. Suchak, B. Ray, C. Rubio-González. // Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis, Santa Barbara, CA, USA, July 10 - 14, 2017 / Ed. by T. Bultan, K. Sen. – ACM, 2017. – P. 396–399. – URL: https://doi.org/10.1145/3092703.3098230.

6. Han J. Characterization and Prediction of Popular Projects on GitHub. / J. Han, S. Deng, X. Xia, D. Wang, J. Yin. // 43rd IEEE Annual Computer Software and Applications Conference, COMPSAC 2019, Milwaukee, WI, USA, July 15-19, 2019, Volume 1 / Ed. by V. Getov, J. Gaudiot, N. Yamai, et al. – IEEE, 2019. – P. 21–26. – URL: https://doi.org/10.1109/COMPSAC.2019.00013.

7. Milovidov A. Everything You Ever Wanted To Know About GitHub (But Were Afraid To Ask). – 2020. – URL: https://ghe.clickhouse.tech/.

8. Puhlfürß T., Montgomery L., Maalej W. An Exploratory Study of Documentation Strategies for Product Features in Popular GitHub Projects. // IEEE International Conference on Software Maintenance and Evolution,

ICSME 2022, Limassol, Cyprus, October 3-7, 2022. – IEEE, 2022. – P. 379–383. – URL: https://doi.org/10.1109/ICSME55016.2022.00043.

9. Ramasamy D. Workflow analysis of data science code in public GitHub repositories. / D. Ramasamy, C. Sarasua, A. Bacchelli, A. Bernstein. // Empir. Softw. Eng. – 2023. – Vol. 28. – No. 1. – P. 7. – URL: https://doi.org/10.1007/s10664-022-10229-z.

10. Rocco J.D. HybridRec: A recommender system for tagging GitHub repositories. / J.D. Rocco, D.D. Ruscio, C.D. Sipio, P.T. Nguyen, R. Rubei. // Appl. Intell. – 2023. – Vol. 53. – No. 8. – P. 9708–9730. – URL: https://doi.org/10.1007/s10489-022-03864-y.

11. Soll M., Vosgerau M. ClassifyHub: An Algorithm to Classify GitHub Repositories. // KI 2017: Advances in Artificial Intelligence - 40th Annual German Conference on AI, Dortmund, Germany, September 25-29, 2017, Proceedings / Ed. by G. Kern-Isberner, J. Fürnkranz, M. Thimm. – Vol. 10505 of Lecture Notes in Computer Science. – Springer, 2017. – P. 373–379. – URL: https://doi.org/10.1007/978-3-319-67190-1_34.