

Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Группа 21.Б08-мм

Реализация интерфейса командной строки в рамках профайлера Desbordante

ГОНЧАРОВ Даниил Юрьевич

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
ассистент кафедры ИАС Г. А. Чернышев

Санкт-Петербург
2024

Оглавление

Введение	3
1. Постановка задачи	4
2. Обзор	5
3. Реализация	7
3.1. Сценарий использования	7
3.2. Опция help	7
3.3. Опции алгоритмов	8
3.4. Дополнительные опции	9
3.5. Излишние опции	9
3.6. Итоги	10
4. Тестирование	11
Заключение	12
Список литературы	13

Введение

Платформы для профилирования данных — удобный инструмент, который содержит в себе набор алгоритмов, упрощающих поиск и изучение информации о различных наборах данных.

Desbordante [2] — одна из таких платформ, позволяющая произвести статистический анализ, поиск ассоциативных правил, поиск и верификацию функциональных зависимостей, поиск и верификацию метрических функциональных зависимостей и многое другое.

Платформа Desbordante предоставляет несколько интерфейсов: веб-приложение, библиотека для языка Python и командная строка. Однако на момент начала выполнения настоящей работы, интерфейс командной строки имел ряд недостатков, который мог пагубно сказаться на опыте использования платформы новым пользователем, что крайне нежелательно для развивающегося продукта.

В ходе данной работы был переосмыслен интерфейс командной строки платформы Desbordante. Основной целью интерфейса стало удобство в использовании, а также предоставление необходимой информации для новых пользователей.

1. Постановка задачи

Целью работы является реализация интерфейса командной строки, позволяющего производить поиск и верификацию функциональных зависимостей, поиск и верификацию приближённых функциональных зависимостей и верификацию метрических функциональных зависимостей, а также способного предоставить информацию об используемых алгоритмах и их возможных опциях. Для её выполнения были поставлены следующие задачи:

1. Провести обзор существующих инструментов для создания интерфейса командной строки и выбрать подходящий.
2. Выбрать сценарий использования платформы Desbordante через интерфейс командной строки.
3. Реализовать интерфейс командной строки.
4. Протестировать работоспособность реализованного интерфейса.

2. Обзор

Для нового интерфейса командной строки было решено использовать уже реализованный интерфейс платформы Desbordante: библиотеку для языка Python, поэтому обзор проводился среди инструментов, предназначенных для соответствующего языка программирования.

Были рассмотрены популярные библиотеки с открытым кодом, а также модуль стандартной библиотеки языка Python.

1. Модуль `argparse`. Плюсом является то, что данный модуль встроен в стандартную библиотеку. Однако, как следует из названия, `argparse` не предоставляет никаких инструментов кроме парсинга аргументов, поэтому все проверки и всю логику, а также тестирующий модуль придётся реализовывать самостоятельно.
2. Python Fire [3] — разрабатываемая Google библиотека, которая позволяет автоматически создавать интерфейс командной строки из любого Python объекта. Помимо этого, данный модуль имеет большой инструментарий, помогающий адаптировать существующий код в CLI.

Как видно из описания, представленный инструмент предназначен для переделывания существующего кода в интерфейс командной строки, что не подходит для выполнения данной работы.

3. Click [1] — популярная библиотека, которая предоставляет инструментарий для создания интерфейса командной строки через использование декораторов. Данный подход улучшает читаемость кода и позволяет избавиться от реализации всей логики в `main`. Данный модуль имеет встроенные инструменты для проверки правильности ввода параметров, настройки значений по умолчанию, а также тестирования реализованного интерфейса.
4. Typer [4] является развивающейся библиотекой, которая работает на основе `click`, но использует совершенно другой синтаксис.

Однако это всё ещё молодой проект, который находится в активной разработке (версия на момент выполнения данной работы — 0.8.0).

На основе проведённого обзора было решено реализовывать интерфейс командной строки с использованием библиотеки `click`, поскольку она предоставляет необходимый инструментарий, имеет стабильную версию (на момент выполнения данной работы — 8.1.6), а также подробную документацию [5].

3. Реализация

3.1. Сценарий использования

Было рассмотрено два следующих сценария использования платформы Desbordante из консоли:

1. Стандартный one-liner;
2. Интерактивное задание аргументов:
 - Выбрать задачу;
 - Выбрать алгоритм;
 - Выбрать таблицу;
 - Задать необходимые параметры;
 - Показать результат, с которым можно взаимодействовать.

Click предоставляет инструменты для реализации обоих сценариев, поэтому используемая библиотека на выбор не влияла.

После обсуждения было решено, что даже при использовании второго варианта, реализация первого всё равно необходима, чтобы пользоваться платформой Desbordante через Python скрипты. Поэтому было принято решение реализовывать первый сценарий.

3.2. Опция `help`

Далее было необходимо выбрать каким образом пользователю увидеть `help` страницу для конкретной задачи или алгоритма, поскольку хранение информации обо всех задачах, алгоритмах и соответствующих опциях в одной `help` странице может существенно затруднить поиск нужной части.

Были рассмотрены следующие варианты:

1. Задание необходимой части информации опции `help` в виде значения;

Возможное использование при таком варианте выглядит так:

```
python3 cli.py --help=fd  
python3 cli.py --help=pyro
```

2. Показывать help страницу в зависимости от уже заданных опций;

Возможное использование:

```
python3 cli.py --task=fd --help  
python3 cli.py --algo=pyro --help
```

3. Создание различных help опций для каждой страницы;

Возможное использование:

```
python3 cli.py --fd-help  
python3 cli.py --pyro-help
```

Из перечисленных вариантов был выбран второй, поскольку первый и третий варианты слишком выделяются на фоне других консольных приложений, что может пагубно сказаться на опыте новых пользователей.

3.3. Опции алгоритмов

Необходимо было решить, как задавать опции для всех алгоритмов. Вариант, при котором все опции прописываются вручную, был отброшен сразу, поскольку при таком подходе после любых изменений в библиотеке Desbordante, например изменение имени опции, необходимо переписывать код для CLI.

Поскольку в коде планировалось реализовывать словарь, содержащий объекты поддерживаемых алгоритмов, было решено добавить в библиотеку Desbordante метод, позволяющий получить название и тип всех опций для конкретного алгоритма. Используя функцию, возвращающую декоратор, добавленный метод позволил задавать опции для алгоритмов динамически. Данное решение значительно упростит поддержание кода для CLI.

3.4. Дополнительные опции

Помимо опций `help`, `task`, `algo` и опций алгоритмов, были добавлены дополнительные опции: `filename` и `verbose`, чтобы улучшить взаимодействие с платформой Desbordante через интерфейс командной строки.

- Опция `filename`

Данная опция позволяет изменить поток вывода для результата. Если опции предоставлено строковое значение, будет создан новый файл либо перезаписан существующий с соответствующим названием. Файл будет содержать результат выполнения.

```
python3 cli.py --algo=hyfd --table=table.csv --filename=res.txt
```

- Опция `verbose`

Данная опция позволяет, помимо результата, получить дополнительную информацию о выполненной задаче: все установленные опции и время выполнения.

```
python3 cli.py --algo=hyfd --table=table.csv --verbose
```

```
algo: hyfd
```

```
table: ('iris.csv', ',', True)
```

```
verbose: True
```

```
Elapsed time during execution in seconds: 0.0009400229999999982
```

```
Result:
```

```
[( 0 1 2 ) -> 4, ( 0 2 3 ) -> 4, ( 1 2 3 ) -> 4, ( 0 1 3 ) -> 4]
```

3.5. Излишние опции

Для улучшения взаимодействия с платформой Desbordante через CLI, пользователь получает информацию об излишне заданных опциях.

```
python3 cli.py --algo=hyfd --table=table.csv --q=2
```

```
Unused options: 'q'
```

```
[( 0 1 2 ) -> 4, ( 0 2 3 ) -> 4, ( 1 2 3 ) -> 4, ( 0 1 3 ) -> 4]
```

3.6. Итоги

Был реализован интерфейс командной строки, способный запускать поиск и верификацию функциональных зависимостей, поиск и верификацию приближённых функциональных зависимостей и верификацию метрических функциональных зависимостей. Также возможно получить информацию обо всех алгоритмах и задачах, изменить поток вывода и добавить к результату расширенную информацию о выполненной задаче.

Реализация данного интерфейса была принята в основной репозиторий GitHub¹.

¹<https://github.com/Desbordante/desbordante-core/pull/237>

4. Тестирование

Тестирование реализованного интерфейса производилось при помощи встроенного в Click модуля CliRunner. Данный модуль позволяет эмулировать запуск кода через консоль.

Было принято решение тестировать только правильность создания объекта алгоритма, поскольку тестирование корректности результатов работы алгоритмов уже реализовано в библиотеке Desbordante.

Для такого варианта тестирования можно было создать отдельную опцию и выделить отдельный путь выполнения кода, если он запускается для тестов. Однако был придуман другой способ. Как уже было сказано, в коде существует словарь, содержащий объекты всех алгоритмов, в тестирующем модуле объекты алгоритмов подменяются mock объектами, у которых метод execute() сравнивает опции алгоритма, заданные внутри кода, с опциями, прописанными отдельно, и завершает выполнение. Таким образом, в интерфейсе отсутствуют лишние опции, и выделять отдельный путь выполнения кода не нужно.

Тестирование показало, что объекты алгоритмов, создающиеся внутри кода, соответствуют ожидаемым.

Заключение

В рамках данной работы были достигнуты следующие результаты:

1. Проведён обзор инструментов для создания интерфейса командной строки.
2. Выбран сценарий использования платформой Desbordante при помощи интерфейса командной строки.
3. Реализован интерфейс командной строки, с помощью которого можно запускать алгоритмы для поиска и верификации различных функциональных зависимостей.
4. Проведено тестирование реализованного интерфейса.

Список литературы

- [1] Click: Python composable command line interface toolkit. — <https://github.com/pallets/click>. — 2023.
- [2] Desbordante: a Framework for Exploring Limits of Dependency Discovery Algorithms / Maxim Strutovskiy, Nikita Bobrov, Kirill Smirnov, George A. Chernishev // 29th Conference of Open Innovations Association, FRUCT 2021, Tampere, Finland, May 12-14, 2021. — IEEE, 2021. — P. 344–354. — URL: <https://doi.org/10.23919/FRUCT52173.2021.9435469>.
- [3] Python Fire: a library for automatically generating command line interfaces. — <https://github.com/google/python-fire>. — 2023.
- [4] Typer: build great CLIs. — <https://github.com/tiangolo/typer>. — 2023.
- [5] Welcome to Click. — <https://click.palletsprojects.com/en/8.1.x/>. — 2023.