Санкт-Петербургский государственный университет

Кафедра информационно-аналитических систем

Группа 22.Б08-мм

Расширение набора простых статистик в “Desbordante”

Аносов Павел Игоревич

Отчёт по учебной практике
в форме «Решение»

Научный руководитель:
асс. кафедры ИАС Г. А. Чернышев

Санкт-Петербург
2024

Оглавление

Введение **3**

1. Постановка задачи **4**

2. Обзор **5**

2.1. pybind11 5

3. Реализация **6**

3.1. Статистики 6

3.2. Pybind11 7

3.3. Архитектурные улучшения 8

3.4. Тестирование 8

3.5. UML-диаграммы 9

4. Эксперимент **10**

4.1. Нагрузочное тестирование 10

4.2. Примеры использования 11

5. Итоговая таблица статистик **12**

Заключение **15**

Список литературы **16**

Введение

Профилирование данных — это процесс исследования данных, направленный на извлечение из них метаданных. Они представляют собой название файла, его тип, размер, а также различные скрытые закономерности. Профилирование данных позволяет оценить качество данных: наличие пропущенных значений, нарушения целостности и логики связей между значениями полей.

Профилирование данных бывает двух типов: наукоёмкое и простое. Наукоёмкое включает в себя поиск функциональных зависимостей и других скрытых закономерностей, тогда как простое подразумевает подсчёт простых статистик: среднее и медианное значения, процентиль, стандартное отклонение, количество строк и столбцов в таблице.

Desbordante [1] — это профилировщик данных, направленный на извлечение наукоемких паттернов. Он разрабатывается с 2019 года, написан на C++ и является open-source проектом с кодом на GitHub. Изначально Desbordante создавался как профилировщик наукоёмких данных, из-за чего он уступал аналогам, которые предоставляли больше информации о данных за счёт простого профилирования.

Данная работа является второй работой, призванной расширить нишу Desbordante и сократить разрыв между функционалом, предоставляемым простыми профилировщиками данных, в ней мы сосредоточимся на простом профилировании табличных данных и продолжим расширять набор уже существующих простых статистик в Desbordante.

1. Постановка задачи

К моменту начала работы в проекте уже были реализованы некоторые простые статистики, а также набор модульных тестов к ним. В работе Михаила Фирсова “Реализация статистик к Desbordante” [3] был реализован и доработан ряд простых статистик, но не было возможности их использования в Python.

Целью работы является расширение имеющегося набора простых статистик и добавление поддержки вызовов методов для получения статистик из Python при помощи библиотеки pybind11 [2]. Для достижения данной цели были поставлены следующие задачи:

1. Реализовать методы для получения новых статистик в C++;
2. Создать модульные тесты в C++, проверяющие их корректность;
3. Сделать возможным использование статистик в Python;
4. Создать тесты для проверки корректности работы статистик в Python.

2. Обзор

2.1. pybind11

При работе над данным проектом была использована библиотека pybind11. Это легковесная библиотека, которая делает возможным вызов C++ кода из Python. Она предоставляет средства для привязки функций, классов и объектов C++ к Python, что делает процесс интеграции более удобным, так как количество стороннего кода, который нужно написать для достижения результата, минимально. Использование этой библиотеки выглядит следующим образом:

1. Создание привязок с помощью pybind11 API между C++ и Python объектами;

2. Компиляция .cpp файла в исполняемый файл и последующий импорт в Python.

Пример использования:
example.cpp:

```
#include <pybind11/pybind11.h>
int add(int a, int b) {
    return a + b;
}
PYBIND11_MODULE(example, m) {
    m.def("add", &add, "A␣function␣which␣adds␣two␣numbers");
}
```

example.py:

```
import example
result = example.add(4, 5)
print(result)  # Output: 9
```

3. Реализация

Перед началом работы было необходимо выяснить, какие статистики уже были реализованы, а какие ещё предстоит реализовать. В работе Михаила Фирсова [3] приведена итоговая таблица, в которой представлен список ещё не реализованных статистик. Дальнейший выбор был обусловлен частотой использования тех или иных статистик: в первую очередь были реализованы наиболее используемые.

3.1. Статистики

В ходе выполнения поставленной задачи были реализованы следующие статистики:

1. Для колонок числового типа:
    - mean_absolute_deviation — среднее абсолютное отклонение
    - median_absolute_deviation — среднее медианное отклонение
    - num_zeros — число нулей
    - num_negatives — число отрицательных значений
    - sum_of_squares — сумма квадратов
    - median — медиана
    - geometric_mean — среднее геометрическое
2. Для колонок строчного типа:
    - total_char_count — общее число символов во всех строках
    - digit_chars — число цифр
    - lowercase_chars — число символов в нижнем регистре
    - uppercase_chars — число символов в верхнем регистре
    - non_letter_chars — число не буквенных символов
    - vocab — символы во всех строчках

3. Для всей таблицы:

    - row_has_null_ratio — число колонок, в которых есть значение NULL
    - row_is_null_ratio — число колонок, в которых есть только значение NULL
    - unique_row_ratio — число колонок с уникальными занчениями во всех строчках

3.2. Pybind11

С помощью библиотеки pybind11 была добавлена возможность использования уже существующих и добавленных методов для получения статистик в Python. Ниже представлены имевшиеся на момент начала учебной практики статистики в Desbordante, которые были портированы в Python:

- number_of_values — количестсво непустых и не NULL значений в колонке.
- number_of_columns — количество колонок в таблице.
- distinct — число уникальных значений в колонке.
- is_categorical — проверяет, является ли колонка категориальной. Вычисляется по формуле: $distinct \leq min($ $number\_of\_values - 1, 10 + number\_of\_values/1000)$.
- show_sample — возвращает срез таблицы в некотором диапазоне.
- average — среднее значение в колонке.
- corrected_STD — скоректированное стандартное отклонение колонки.
- skewness — коэффициент ассиметрии колонки.

- kurtosis — коэффициент эксцесса колонки.
- central_moment_of_dist — центральный момент колонки.
- corrected_central_moment_of_dist — нормированный момент колонки.
- min — минимальное значение в колонке.
- max — максимальное значениие в колонке.
- sum — сумма в колонке.
- quantile — процентиль (доступны 25, 50, 75).

3.3. Архитектурные улучшения

Во время написания новых статистик были обнаружены участки кода, где было возможно улучшение производительности. Был доработан класс model::DoubleType: добавлен новый метод, с помощью которого можно записывать данные по указанному месту в памяти.

Использование данного метода помогло снизить количество лишних выделений памяти: в методе DataStats::CalculateCentralMoment на каждой итерации цикла выделялась пямять, тогда как это можно было сделать лишь однажды.

Исправлена реализация метода ShowSample: удалена возможность вывода фиксированного количества символов для строковых и численных типов данных.

Также были исправлены ошибки в описании некоторых методов.

3.4. Тестирование

Для тестирования корректности подсчёта новых добавленных в C++ статистик был выбран уже использующийся для этих целей датасет TestDataStats.csv. Также, помимо тестов на языке C++, для подтверждения корректности работы добавленных Python-привязок для них

были написаны тесты на языке Python, которые тестировались на том же датасете.

3.5. UML-диаграммы

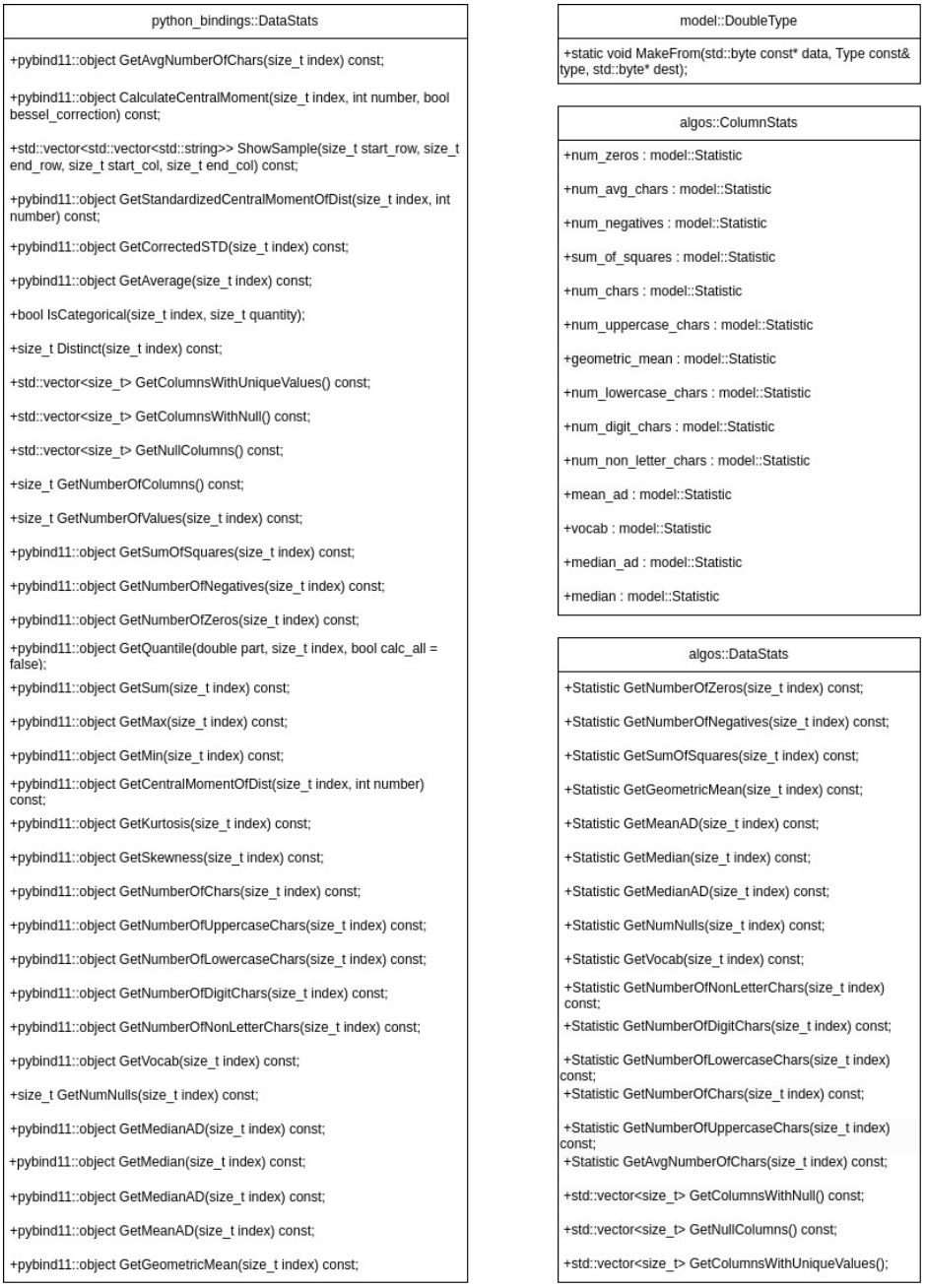

Рис. 1: Добавленные методы

4. Эксперимент

4.1. Нагрузочное тестирование

В данном разделе будет проведено нагрузочное тестирование для сбора времени выполнения статистик в Python. Для этого был выбраны датасеты iowa1kk.csv (один миллион строк, 24 столбца) и adult.csv (33 тыс. строк, 15 столбцов), так как в них присутсвуют данные и числовых типов, и строчных. Результаты приведены в Таб. 1

Таблица 1: Время подсчёта статистик

Статистика	Время подсчета (мс)	
	Датасет	
	iowa1kk.csv	adults.csv
sum_of_squares	463	13
geometric_mean	472	13
mean_ad	848	26
median_ad	1114	25
vocab	9770	130
number_of_digit_chars	1360	30
number_of_chars	958	27
avg_number_of_chars	956	28

В итоговое время входит только подсчёт соответствующей статистики, не включая время на загрузку данных из датасета.

4.2. Примеры использования

example.py:

```
import desbordante as db


# Parameters for data_stats.load_data(...)
DATASET_PATH = "examples/datasets/Workshop.csv"
SEPARATOR = ','
HAS_HEADER = True



def main() -> None:
    data_stats = db.DataStats()
    data_stats.load_data(DATASET_PATH, SEPARATOR, HAS_HEADER)
    data_stats.execute()

    num_cols = data_stats.get_number_of_columns()
    for i in range(num_cols):
        print(f"Avg: {data_stats.get_average(i)}")
        print(f"Distinct: {data_stats.get_number_of_distinct(i)})
        print(f"Median: {data_stats.get_median(i)})
        print(f"Min: {data_stats.get_min(i)})
        print(f"Max: {data_stats.get_max(i)})


    # Print all statistics at once
    print(data_stats.get_all_statistics_as_string())



if __name__ == "__main__":
    main()
```

5. Итоговая таблица статистик

Приведенная ниже таблица является анализом поддерживаемых статистик в Desbordante на момент окончания учебной практики. Более подробное описание каждой статистики можно найти в работе Михаила Фирсова “Реализация статистик к Desbordante” [3].

Таблица 2: Поддерживаемые статистики

Тип данных	Статистика	Есть в C++	Есть в Python
Строковые	vocab	+	+
	words	−	−
	top_k_chars	−	−
	top_k_words	−	−
	min_words	−	−
	max_k_words	−	−
	word_count	−	−
	non_letter_chars	+	+
	diacritic_chars	−	−
	digit_chars	+	+
	lowercase_chars	+	+
	uppercase_chars	+	+
	excl_first_letters	−	−
	avg_white_spaces	−	−
	min_white_spaces	−	−
	max_white_spaces	−	−
	avg_chars	+	+
	min_chars	−	−
	max_chars	−	−
	total_char_count	+	+
	entirely_lowercase_count	−	−
	entirely_uppercase_count	−	−

Общие	data_type	+	+
	column_name	−	−
	categorical	+	+
	samples	+	+
	min	+	+
	max	+	+
	quantiles	+	+
	null_count	+	+
	unique_count	+	+
	sample_size	+	+
	categorical_count	−	−
	unique_ratio	−	−
	categories	−	−
Float	precision	−	−
	sample_ratio	−	−
DateTime	highest_time	−	−
	lowest_time	−	−
Числовые	sum	+	+
	mean	+	+
	median	+	+
	geometric_mean	+	+
	variance	+	+
	std_dev	+	+
	central_moment	+	+
	standardized_central_moment	+	+
	skewness	+	+
	kurtosis	+	+
	median_absolute_deviation	+	+
	mean_absolute_deviation	+	+
	num_zeros	+	+
	num_negatives	+	+
	bias_correction	−	−

	histogram	−	−
	histogram_and_quantiles	−	−
	sum_of_squares	+	+
Bool	true_count	−	−
	false_count	−	−
Вся таблица	column_count	+	+
	row_has_null_ratio	+	+
	row_is_null_ratio	+	+
	unique_row_ratio	+	+
	duplicate_row_count	−	−
	file_type	−	−
	encoding	−	−
	correction_matrix	−	−
	chi2_matrix	−	−
	profile_schema	−	−

Заключение

По итогам учебной практики стало возможным использование методов для получения всех реализованных и новых добавленных статистик в Python. Также для них были добавлены тесты и примеры использования в Python. По результатам выполнения работы:

1. Реализованы методы для получения новых статистик;
2. Созданы модульные тесты, проверяющие их корректность;
3. Стало возможным использование статистик в Python;
4. Созданы тесты для проверки корректности работы статистик в Python.

Исходный код доступен на GitHub. Изменения приняты:

1. https://github.com/Mstrutov/Desbordante/pull/315
2. https://github.com/Mstrutov/Desbordante/pull/279
3. https://github.com/Mstrutov/Desbordante/pull/224
4. https://github.com/Mstrutov/Desbordante/pull/219
5. https://github.com/Mstrutov/Desbordante/pull/208
6. https://github.com/Mstrutov/Desbordante/pull/207
7. https://github.com/Mstrutov/Desbordante/pull/193

Изменения на стадии рассмотрения:

1. https://github.com/Mstrutov/Desbordante/pull/336

Список литературы

[1] Desbordante: a Framework for Exploring Limits of Dependency Discovery Algorithms / Maxim Strutovskiy, Nikita Bobrov, Kirill Smirnov, George Chernishev // 2021 29th Conference of Open Innovations Association (FRUCT). — 2021. — P. 344–354.

[2] Jakob Wenzel, Rhinelander Jason, Moldovan Dean. pybind11 – Seamless operability between C++11 and Python. — 2017. — URL: https://github.com/pybind/pybind11.

[3] Фирсов Михаил. Реализация статистик к Desbordante. — URL: https://github.com/Mstrutov/Desbordante/blob/main/docs/papers/Statistics%20-%20Mikhail%20Firsov%20-%202022%20autumn.pdf (дата обращения: 25 ноября 2023 г.).