Міністерство освіти і науки України
Національний технічний університет України «КПІ» імені Ігоря Сікорського Кафедра обчислювальної техніки ФІОТ

ЗВІТ
з лабораторної роботи №5
з навчальної дисципліни «Економіка ІТ індустрії та підприємництво»

Тема:

ОЦІНЮВАННЯ ВАРТОСТІ І ІНШИХ ХАРАКТЕРИСТИК ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА МЕТОДОМ АНАЛОГІЇ

Виконав:
Студент 4 курсу кафедри ФІОТ,
Навчальної групи ІП-11
Головня О. Р.

Перевірив:
Родіонов П. В.

Київ 2024

I. Мета.

Мета роботи: навчитися здійснювати оцінку проєкту за рахунок використання методу аналогій.

II. Завдання.

Використовуючи власні наробки, або веб сервіси для спільної розробки програмного забезпечення (наприклад GitHub), або проєктуючи за завданням викладача відповідні застосунки виконати наступне:

1. Вибрати проєкти та створити базу даних проєктів. Навести посилання на вибрані проєкти, для можливості викладачем перегляду проєктів і оцінювання вірності результатів що будуть отримані студентом під час виконання розрахунків.

2. Вибрати проєкт, який відкладаємо в сторону – він буде як «новий проєкт» що використовується для оцінки зусиль проєкту програмного забезпечення за аналогією.

3. Визначити необхідні для застосування методу аналогії атрибути (характеристики) «нового проєкту», не враховувати в ці атрибути size, Effort та мову програмування проєкту. Обчислити значення відповідних характеристик для всіх проєктів.

4. Внести всі дані у власний Database (табл. 18):

Таблиця 18 – Структура Database

Проєкти	githuburl	att1	att2	att3	…	Мова програмування	Size	Effort
project1								
project2								
…..								
project20								
project_new								

4. Застосовуючи одну або декілька метрик відстані, обчислити відстані усіх проєктів до «нового» проєкту, внести у свій data set. Виявити три найближчі проєкти. Навести розрахунки, найближчі проєкти позначити – наприклад іншим кольором.

5. Оцінити економічні показники нового проєкту. При оцінці враховувати мову програмування аналогічних проєктів (оскільки «новий» проєкт має розроблятися на певній мові і відповідно для застосування методу аналогій повинна бути вибрана ця мова або близька).

III. Результати виконання лабораторної роботи.

Замітки:

Економічні розрахунки

Зусилля (людина/місяць) Effort = ab * size^{bb},
де ab, bb коефіцієнти,
size – розмір продукту в KLOC.

Вартість (грн.) Cost = Effort * salary,
де salary – заробітна платня.

Час на розробку Schedule = cb * Effort^{db},
де cb, db –коефіцієнти.

Таблиця 1.

Тип проекту	***ab***	***bb***	***cb***	***db***
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Тип «Organic» являє собою відносно невеликий (до 25 тис. рядків коду) та простий проект, який виконується невеликою командою.

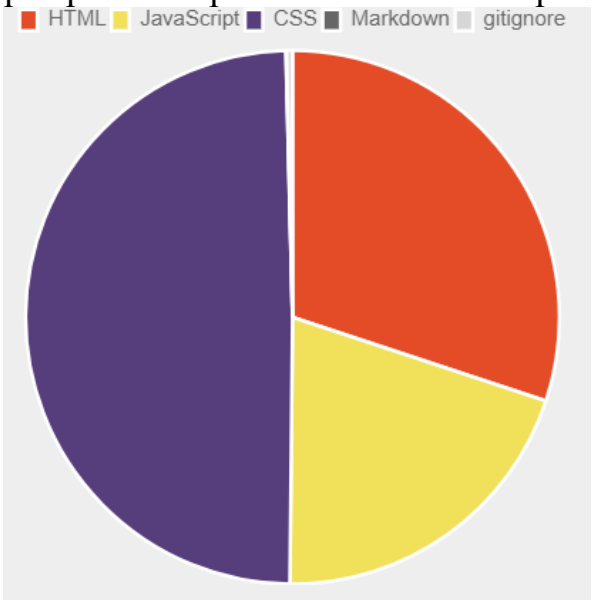
Тип «Semi-detached» являє собою середній за розміром (до 75 тис. рядків коду) та складністю проект, в якому команда має змішаний рівень досвіду і відносно жорсткі вимоги.

Тип «Embedded» являє собою проект, який виконується в умовах жорстких технічних, програмних та експлуатаційних обмежень.

Таким чином:

Sallary		ab	bb	cd	db
400	Organic	2.4	1.05	2.5	0.38
800	Semi-detached	3	1.12	2.5	0.35
2000	Embedded	3.6	1.2	2.5	0.32

Як «Новий проєкт» я обрав репозиторій свого власного проєкту



Files	Lines	Blanks	Comments	Lines of Code
39	3153	438	195	2520

Для оцінки обрав 20 випадкових проєктів, включаючи власний, з публічних репозиторіїв на GitHub і оцінив за критеріями: Lines of code, Files, Effort, Cost, Schedule

			Name	Lines of Code	KLOC	Files	Effort	Cost	Schedule		Language
5	1	Новий проєкт	YeaLow	2520	2.52	39	6.33	2533.6	5.04	Organic	HTML, JavaScript
6	2	https://github.com/freeCodeCam	freeCodeCamp	1431780	1 431.78	12030	10273.13	8218502.4	63.39	Semi-detached	Markdown, TypeScript
7	3	https://github.com/EbookFounda	free-programming-books	21320	21.32	218	59.63	23850.5	11.82	Organic	Markdown, YAML
8	4	https://github.com/jwasham/codi	coding-interview-university	49884	49.88	36	239.24	191393.8	17.00	Semi-detached	Markdown, TypeScript
9	5	https://github.com/sindresorhus/	awesome	1131	1.13	16	2.73	1092.5	3.66	Organic	Markdown
10	6	https://github.com/kamranahmed	developer-roadmap	527932	527.93	7534	6658.80	13317603.5	41.82	Embedded	Markdown, TypeScript
11	7	https://github.com/public-apis/pu	public-apis	3166	3.17	18	8.05	3219.6	5.52	Organic	Markdown, Python
12	8	https://github.com/donnemartin/s	system-design-primer	11091	11.09	59	30.02	12008.5	9.11	Organic	Markdown, Python
13	9	https://github.com/facebook/fbos	fboss	3340808	3 340.81	3751	60943.07	121886138.1	84.94	Embedded	C++
14	10	https://github.com/codecrafters-i	build-your-own-x	451	0.45	2	1.04	416.1	2.54	Organic	Markdown
15	11	https://github.com/tensorflow/ten	tensorboard	242515	242.52	1622	1406.14	1124911.8	31.60	Semi-detached	TypeScript, Python
16	12	https://github.com/getify/You-Do	You-Dont-Know-JS	12748	12.75	66	34.75	13899.1	9.63	Organic	Markdown
17	13	https://github.com/trekhleb/javas	javascript-algorithms	45815	45.82	613	217.50	173996.2	16.45	Semi-detached	JavaScript
18	14	https://github.com/twbs/bootstra	bootstrap	134348	134.35	586	725.67	580537.4	25.07	Semi-detached	JavaScript, Sass
19	15	https://github.com/vinta/awesom	awesome-python	1097	1.10	14	2.65	1058.0	3.62	Organic	Markdown, YAML
20	16	https://github.com/ohmyzsh/ohm	ohmyzsh	68369	68.37	770	340.54	272429.2	19.24	Semi-detached	Markdown, Zsh
21	17	https://github.com/cp-algorithms	cp-algorithms	27304	27.30	264	190.45	380893.9	13.41	Embedded	Markdown, C++
22	18	https://github.com/bregman-arie/	devops-exercises	27123	27.12	339	120.91	96727.4	13.39	Semi-detached	Markdown, Python
23	19	https://github.com/Chalarangelo/	portfolio	4664	4.66	25	12.09	4835.8	6.45	Organic	JavaScript
24	20	https://github.com/TheAlgorithm	C-Plus-Plus	36229	36.23	422	267.41	534810.7	14.95	Embedded	C++

Вирахуємо евклідову відстань від нашого «нового проекту» до інших за відповідною формулою:

Найвідоміша така метрика відстані - евклідова або прямолінійна відстань, яка має прямолінійне геометричне значення як відстань двох точок у k-мірному евклідовому просторі:

$$d_{new,i} = \left\{ \sum_{j=1}^{k} (Y_j - X_{ij})^2 \right\}^{1/2}, \qquad i = 1, 2, \ldots, n$$

Результати замірів:

Name	Відстань	
YeaLow	0	0
freeCodeCamp	8339375.215	0.08339
free-programming-books	28423.32296	0.00028
coding-interview-university	194708.9416	0.00195
awesome	2001.700291	0.00002
developer-roadmap	13325436	0.13325
public-apis	942.5419157	0.00001
system-design-primer	12776.4355	0.00013
fboss	121929327.6	1.21929
build-your-own-x	2960.782718	0.00003
tensorboard	1147752.126	0.01148
You-Dont-Know-JS	15290.08833	0.00015
javascript-algorithms	176845.3036	0.00177
bootstrap	592847.2026	0.00593
awesome-python	2050.125227	0.00002
ohmyzsh	277813.4949	0.00278
cp-algorithms	379171.2516	0.00379
devops-exercises	97354.39558	0.00097
portfolio	3145.943982	0.00003
C-Plus-Plus	533343.6352	0.00533

Обравши 3 проєкти з найменшою відстанню, а також наближені по мові програмування, це і є наші найбільш схожі проекти.

Порівняємо середнє значення Effort наших наближених проектів:

public-apis	3166	3.17	18	8.05
system-design-primer	11091	11.09	59	30.02
fboss	3340808	3 340.81	3751	60943.07
build-your-own-x	451	0.45	2	1.04
tensorboard	242515	242.52	1622	1406.14
You-Dont-Know-JS	12748	12.75	66	34.75
javascript-algorithms	45815	45.82	613	217.50
bootstrap	134348	134.35	586	725.67
awesome-python	1097	1.10	14	2.65
ohmyzsh	68369	68.37	770	340.54
cp-algorithms	27304	27.30	264	190.45
devops-exercises	27123	27.12	339	120.91
portfolio	4664	4.66	25	12.09
C-Plus-Plus	36229	36.23	422	267.41
			Average:	7.594534936

AVG Effort = 7.59

IV. Висновки.

У ході лабораторної роботи було проведено поверхневий аналіз проєктів на метод аналогій. Під час виконання даної лабораторної роботи я навчився виконувати оцінку проектів за характеристиками та обчислювати їх схожість.

Виконав: студент ІП-11 Олександр Головня