

ПРАКТИЧНА РОБОТА №5

ОЦІНЮВАННЯ ВАРТОСТІ І ІНШИХ ХАРАКТЕРИСТИК ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА МЕТОДОМ АНАЛОГІЇ

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

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Методи і моделі оцінки вартості програмного забезпечення можна розділити на дві групи: неалгоритмічні методи і алгоритмічні моделі. До неалгоритмічних методів належать Price-to-win, оцінка по Паркінсону, експертна оцінка, оцінка по аналогії. До алгоритмічних моделей належать SLIM і COSOMO.

Сутність неалгоритмічних методів полягає в тому, що при оцінці вартості програмного забезпечення використовуються певні схеми і принципи, а не математичні формули.

Аналогією зазвичай вважають структурні зв'язки між доменом-джерелом (базою) та цільовими доменами. Щоб встановити аналогію, загальні під структури двох доменів ідентифікуються та відображаються один в одній, внаслідок чого виникає або ні аналогічне відношення.

Встановлення аналогії, як правило, регулюється певними обмеженнями, наприклад систематичність, структурна узгодженість або обмеження можливих відображень одне в одне. Загально прийнятого набору таких принципів немає. Основне поняття, яке застосовано в методі аналогії - відстань між історичними проєктами і новим проєктом. Відстань вимірює, наскільки відрізняється два проєкти (один з них історичний, а другий, новий) з точки зору значень їх атрибутів. Використовується метрика відстані на основі стандартизованих значень k атрибутів для цих проєктів. Найвідоміша така метрика відстані - евклідова або прямолінійна відстань, яка має прямолінійне геометричне значення як відстань двох точок у k -мірному евклідовому просторі.

Оцінка за аналогією здійснюється шляхом використання інформації про характеристики проєктів, які було виконано в минулому для оцінки

характеристик нового програмного забезпечення, яке буде розроблятися. Основною ідеєю оцінки вартості за методом аналогії є використання історичної інформації від завершених проєктів з відомими витратами.

Методика розв'язання типових задач, виконання типових завдань

Спочатку необхідно охарактеризувати новий проєкт, атрибутами, ідентичними тим, які притаманні виконаним проєктам, зареєстрованим в базі даних. Приклади атрибутів проєкту є довжина вихідного коду, мова програмування, досвід персоналу, інші. Потім, порівнювати з обраними виконаними проєктами.

Оцінка зусиль проєкту програмного забезпечення за аналогією зазвичай включає ряд кроків:

1. Вимірювання або оцінка значень показників для цільового проєкту.
2. Пошук у сховищі виконаних проєктів, проєктів подібних цільовому та вибір одного або декількох проєктів як вихідних аналогів.
3. Використання значення зусиль і вартості проєктів як вихідних аналогів в якості початкової оцінки для цільового проєкту.
4. Порівняння відомих метричних значень для цільового та вихідних проєктів.
5. Коригування значення зусиль і вартості з урахуванням відмінностей між ціллю та джерелами.

В таблиці 15 наведено приклад множини показників для цільового проєкту.

Таблиця 15 – Показники для цільового проєкту

	Зусилля	Атрибут 1	Атрибут 2	...	Атрибут k
Проект 1	E_1	X_{11}	X_{12}	...	X_{1k}
Проект 2	E_2	X_{21}	X_{22}	...	X_{2k}
Проект 3	E_3	X_{31}	X_{32}	...	X_{3k}
....
Проект n	E_n	X_{n1}	X_{n2}	...	X_{nk}
Новий проєкт	Невідомий	Y_1	Y_2	...	Y_k

Пошук у серед виконаних проєктів, проєктів подібних цільовому здійснюється шляхом використання поняття відстань. Відстань вимірює, наскільки відрізняється два проєкти з точки зору значень їх атрибутів. Використовується метрика відстані на основі стандартизованих значень атрибутів k для цих проєктів.

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

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

Евклідова або прямолінійна відстань в трьох мірному просторі продемонстрована на рисунку 17. Зазначимо, що можуть також використовуватися інші метрики відстані.

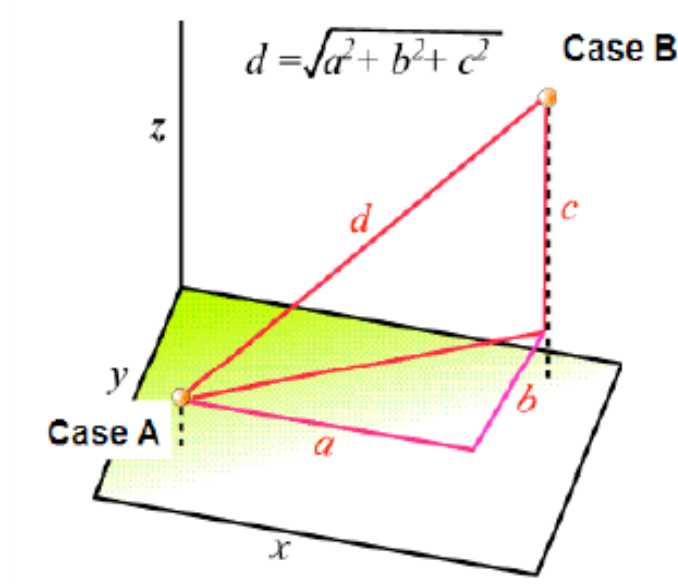


Рисунок 17 – Евклідова або прямолінійна відстань

Оцінка зусиль за допомогою аналогій базується на завершених проєктах, які є подібними до нового. Користувач методу повинен обчислити відстані нового проєкту від усіх проєктів бази даних та визначити кілька «сусідських» проєктів, тобто тих, з якими відносно невелике значення відстані. Оцінка зусиль зрештою отримують деякі поєднання зусиль сусідніх проєктів. Зазвичай використовується середня статистика (зважена або проста) або медіана цих значень зусиль. Відсутні значення потребують подальшого коригування методом аналогій. Оскільки взаємозв'язок між зусиллям і розміром є хорошим встановлену, статистику можна також коригувати, використовуючи сусідній за розміром проєкт. Наприклад, у випадку одного «сусіда» лінійне регулювання розміру виконується наступним чином:

$$EFFORT_{NEW} = \frac{SIZE_{NEW}}{SIZE_{NEIGHBOUR}} \times EFFORT_{NEIGHBOUR}$$

Калібрування методу, заснованого на аналогії, вимагає виявлення найкращих конфігурацій доступних параметрів методу. Параметри, які можуть бути скориговані, є:

(а) метрика відстані, за якою проєкти бази даних будуть відсортовані відповідно до їх подібності до оцінюваної (наприклад, евклідова відстань, відстань Манхеттена);

- (b) кількість найближчих проєктів (аналогій);
- (c) набір атрибутів для оцінювання аналогій;
- (d) статистичні дані, які будуть обчислені з зусиль найближчих проєктів і будуть виконувати функції оцінки зусиль для нового проєкту.

Крім того, статистика може бути скоригована за розміром.

Можна використовувати такі набори даних, як база даних Кемерера (табл. 16), база даних Альбрехта (табл. 17), COCOMO Dataset та інші.

Таблиця 16 – База даних Кемерера

Програмне забезпечення	Апаратне забезпечення	Місяці	Рядки коду, тис	Рядки коду, людино-місяці	Кориговані функціональні точки	Некореговані функціональні точки	Зусилля
1	1	17	253,6	884	1217,1	1010	287
1	2	7	40,5	491	507,3	457	82,5
1	4	15	450	406	2306,8	2284	1107,3
1	1	18	214,4	2467	788,5	881	86,9
1	2	13	449,9	1338	1337,6	1583	336,3
1	3	5	50	595	421,3	411	84
2	3	5	43	1853	99,9	97	23,2
1	2	11	200	1535	993	998	130,3
1	1	14	289	2491	1592,9	1554	116
3	1	5	39	542	240	250	72
1	1	13	254,2	983	1611	1603	258,7
1	2	31	128,6	557	789	724	230,7
1	5	20	161,4	1028	690,9	705	157
1	1	26	164,8	667	1374,5	1375	246,9
3	1	14	60,2	861	1044,3	976	69,9

Таблиця 17 – База даних Альбрехта

Вхід	Вихід	Файл	Запити	Функціональні точки	Рядки коду, тис	Некореговані функціональні точки	Зусилля
25	150	60	75	1750	130000	1750	102,4
193	98	36	70	1902	318000	1902	105,2
70	27	12	0	428	20000	535	11,1
40	60	12	20	759	54000	660	21,1
10	69	9	1	431	62000	478	28,8
13	19	23	0	283	28000	377,33	10
34	14	5	0	205	35000	256,25	8
17	17	5	15	289	30000	262,73	4,9
45	64	16	14	680	48000	715,79	12,9
40	60	15	20	794	93000	690,43	19
41	27	5	29	512	57000	465,45	10,8
33	17	5	8	224	22000	298,59	2,9
28	41	11	16	417	24000	490,59	7,5
43	40	35	20	682	42000	802,35	12
7	12	8	13	209	40000	220	4,1
28	38	9	24	512	96000	487,62	15,8
42	57	5	12	606	40000	550,91	18,3
27	20	6	24	400	52000	363,64	8,9
48	66	50	13	1235	94000	1073,91	38,1
69	112	39	21	1572	110000	1310	61,2

Вхід	Вихід	Файл	Запити	Функціональні точки	Рядки коду, тис	Некореговані функціональні точки	Зусилля
25	28	22	4	500	15000	476,19	3,6
61	68	11	0	694	24000	694	11,8
15	15	3	6	199	3000	189,52	0,5
12	15	15	0	260	29000	237,68	6,1

Для розрахунку відстаней можна діяти наступними шляхами, а саме, по перше, обчислювати вручну, по другу, запрограмувати калькулятор, або по третє, використати одну з відомих систем оцінювання.

Приклад виконання

1. Загальні відомості про проєкт що використовується.

Датасет у вигляді електронної таблиці, яка буде використовуватись для розрахунків (рис. 18).

Атрибути:

- довжина розробки (місяців);
- кількість транзакцій;
- кількість сутностей в моделі даних системи;
- ненормовані функціональні точки;
- нормовані функціональні точки.

Мови програмування (на яких розроблені проєкти для датасету):

Проект №1 – C.

Проект №2 – Java.

Проект №3 – C#.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Project	TeamExp	ManagerExp	YearEnd	Length	Transactions	Entities	PointsNonAdjust	Adjustment	PointsAjust	Language	Effort
2	1	2	2	88	3	126	49	175	38	180	3	651
3	2	4	4	85	16	116	170	286	27	263	1	7252
4	3	2	3	84	13	134	77	211	13	165	2	2275
5	4	1	4	86	8	89	200	289	33	283	1	3983
6	5	1	1	84	3	33	72	105	19	88	1	2282
7	6	1	3	86	17	317	119	436	34	432	2	9135
8	7	4	1	85	14	175	277	452	37	461	1	3948
9	8	2	3	87	27	173	332	505	19	424	1	14987
10	9	1	3	86	6	42	31	73	27	67	2	1267
11	10	2	1	84	17	146	112	258	40	271	1	9051
12	11	0	0	86	4	140	94	234	24	208	1	2149
13	12	3	1	85	14	148	324	472	39	491	1	4277
14	13	2	4	85	18	88	170	258	34	255	1	5180
15	14	2	-1	87	10	224	110	334	28	309	2	6783
16	15	2	3	85	8	106	39	145	6	103	1	2331
17	16	3	4	87	14	9	386	395	21	340	2	4494
18	17	1	4	88	4	158	59	217	18	180	3	847
19	18	1	3	86	12	132	89	221	5	155	2	3647
20	19	2	0	86	6	71	235	306	37	312	1	3542
21	20	4	3	86	6	79	128	207	27	190	1	3927
22	21	3	4	87	10	101	57	158	9	117	2	1155
23	22	4	5	86	26	482	227	709	26	645	2	9100
24	23	4	4	86	14	68	316	384	20	326	2	3437
25	24	1	4	85	12	253	52	305	34	302	1	5152
26	25	4	4	85	1	40	60	100	18	83	1	805
27	26	4	4	85	36	886	241	1127	34	1116	1	23940
28	27	4	4	86	12	318	269	587	34	581	2	14973
29	28	2	3	86	24	473	182	655	40	688	2	13860
30	29	3	1	85	12	172	88	260	30	247	1	7854
31	30	4	4	85	12	229	169	398	39	414	3	1400
32	31	3	2	86	6	101	45	146	15	117	2	1876
33	32	2	4	87	34	661	132	793	23	698	3	2352

Рисунок 18 – Дані для розрахунків

Проект №3 виступатиме в нас у ролі «нового» проекту і для нього ми рахуватимемо трудомісткість. А оскільки вона вже й відома, то ми спочатку порахуємо приблизну, а потім порівняємо її із реальною (рис 19):

Project	TeamExp	ManagerExp	YearEnd	Length	Transaction	Entities	ntsNonAdj	adjustmen	ointsAjust	Language	Effort
81	3	2	85	8	119	48	167	26	152	2	1617

Рисунок 19 – Дані щодо проекту №3

Відстань рахуватиметься за евклідовою відстанню. Вона слугуватиме індикатором «наскільки наш проект схожий на вже відомі». Підрахунки робитимуться у Excel, бо це значно полегшує роботу із подібного роду даними (рис. 20).

Project	TeamExp	ManagerExp	YearEnd	Length	Transactions	Entities	PointsNonAdjust	Adjustment	PointsAjust	Language	Effort	Distance
1	2	2	88	3	126	49	175	38	180	3	651	32,68027
2	4	4	85	16	116	170	286	27	263	1	7252	203,5706
3	2	3	84	13	134	77	211	13	165	2	2275	58,00862
4	1	4	86	8	89	200	289	33	283	1	3983	236,8523
5	1	1	84	3	33	72	105	19	88	1	2282	126,4397
6	1	3	86	17	317	119	436	34	432	2	9135	441,759
7	4	1	85	14	175	277	452	37	461	1	3948	482,1214
8	2	3	87	27	173	332	505	19	424	1	14987	521,7384
9	1	3	86	6	42	31	73	27	67	2	1267	149,2783
10	2	1	84	17	146	112	258	40	271	1	9051	165,9669
11	0	0	86	4	140	94	234	24	208	1	2149	101,0099
12	3	1	85	14	148	324	472	39	491	1	4277	534,0122
13	2	4	85	18	88	170	258	34	255	1	5180	186,8154
14	2	-1	87	10	224	110	334	28	309	2	6783	259,644
15	2	3	85	8	106	39	145	6	103	1	2331	59,46427
16	3	4	87	14	9	386	395	21	340	2	4494	462,3127
17	1	4	88	4	158	59	217	18	180	3	847	70,76016
18	1	3	86	12	132	89	221	5	155	2	3647	72,33257
19	2	0	86	6	71	235	306	37	312	1	3542	286,9146
20	4	3	86	6	79	128	207	27	190	1	3927	105,119
21	3	4	87	10	101	57	158	9	117	2	1155	44,76606
22	4	5	86	26	482	227	709	26	645	2	9100	837,2258
23	4	4	86	14	68	316	384	20	326	2	3437	389,6948

Рисунок 20 – Розрахунок евклідової відстані

Наступним кроком буде вибір 3 проєктів із найменшою дистанцією до «нового» проєкту (рис. 21).

Project	TeamExp	ManagerExp	YearEnd	Length	Transactions	Entities	PointsNonAdjust	Adjustment	PointsAjust	Language	Effort	Distance
1	2	2	88	3	126	49	175	38	180	3	651	32,68027
54	2	1	85	9	119	42	161	25	145	2	2569	11,09054
59	1	4	87	8	124	52	176	14	139	2	2723	20,85665
81	3	2	85	8	119	48	167	26	152	2	1617	

Рисунок 21 – Вибір 3 проєктів із найменшою дистанцією до «нового» проєкту

Останнім кроком буде безпосередньо розрахунок трудомісткості проєкту №3. Воно буде дорівнювати середньому значенню трудомісткості вищезазначених проєктів (рис. 22).

Project	TeamExp	ManagerExp	YearEnd	Length	Transactions	Entities	PointsNonAdjust	Adjustment	PointsAjust	Language	Effort	Distance
1	2	2	88	3	126	49	175	38	180	3	651	32,68027
54	2	1	85	9	119	42	161	25	145	2	2569	11,09054
59	1	4	87	8	124	52	176	14	139	2	2723	20,85665
81	3	2	85	8	119	48	167	26	152	2	1617	1981

Рисунок 22 – Розрахунок трудомісткості проєкту №3

Як можна побачити, різниця незначна.

Але в завданні є умова про мову програмування. Один з наведених проєктів (під номером 1) не відповідає їй і менеджер вирішив не враховувати показники для оцінки «нового» проєкту.

Тобто для розробки нового проєкту можна враховувати оцінки тільки двох аналогій.

Далі навести порівняння по атрибутам проєктів аналогій та «нового» проєкту.

Зауваження: чи достатньо двох проєктів для розробки на їх основі «нового» проєкту – вирішувати розробникам.

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

Проекти	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).

КОНТРОЛЬНІ ЗАПИТАННЯ

1. З яких груп складаються методи оцінки вартості програмного забезпечення.

2. В чому сутність неалгоритмічних методів?

3. В чому сутність методу Price-to-win?

4. В чому сутність методу оцінки за Паркінсоном?

5. В чому сутність методу експертної оцінки?

6. В чому сутність методу оцінки за аналогією?

7. В чому сутність методу Learning-Oriented Techniques?

8. В чому сутність методу Neural Networks?

9. В чому сутність методу Dynamics-Based Techniques?

10. Які існують метрики відстані?

11. Переваги та недоліки методу Price-to-win?

12. Які існують переваги та недоліки методу експертної оцінки?

13. Які існують переваги та недоліки методу оцінки за аналогією?

14. В чому сутність Дельфійської методики?