

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ КАФЕДРА
МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

КОМПЛЕКСНА ДИПЛОМНА РОБОТА НА ТЕМУ:

Система автоматизованого аналізу даних при регресійному тестуванні.
Класифікація багів в регресійному тестуванні.
Кластеризація багів в регресійному тестуванні.

ВИКОНАЛИ СТУДЕНТИ ГРУПИ КА-51

РОМАНЧУК Ю.Ю.

КОРОЛЬ І.В.

НАУКОВИЙ КЕРІВНИК: К.Т.Н ДОЦ.

ДІДКОВСЬКА М.В

Регресійне тестування

Регресійне тестування — загальна назва для всіх видів тестування програмного забезпечення, спрямованих на виявлення помилок у вже протестованих ділянках початкового коду. Такі помилки — коли після внесення змін до програми перестає працювати те, що мало б працювати, — називають регресивними помилками.

Одна з головних цілей регресійного тестування - це визначити, чи впливає зміна в одній частині програмного забезпечення на його інші частини.

АКТУАЛЬНІСТЬ РОБОТИ

- Економія часу, коштів, ресурсів при розробці програмного продукту
- Зменшення мануального навантаження на розробників автоматизованих тестів
- Підвищення ефективності та покращення взаємодії учасників SCRUM команди



ІСНУЮЧІ ПІДХОДИ

- Базуються на ручному оцінюванні експертів та автоматичному зберіганні попередніх результатів.
- Переваги:
 - Висока точність результатів аналізу
- Недоліки:
 - Необхідність обробки великої кількості даних вручну
 - Великі часові затрати
 - Рутинна робота
 - Якість залежить від компетенції експерта



Regression Testing

Об'єкт, предмет і мета дослідження

- Об'єкт дослідження –
- Предмет дослідження -
- Мета дослідження – розробити програмний продукт для класифікації та кластеризації багів в регресійному тестуванні

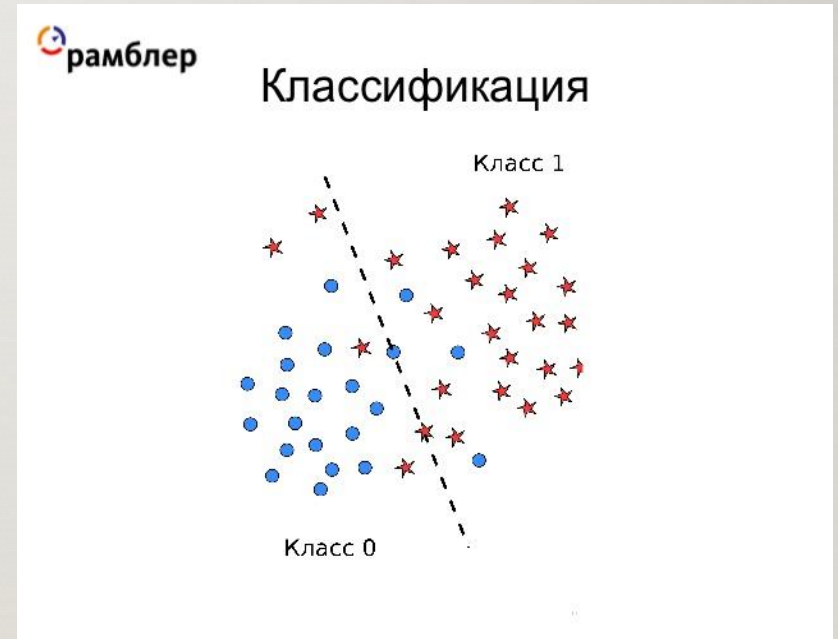
ПОСТАНОВКА ЗАДАЧІ

- Дослідження сучасних математичних моделей побудови кластеризації та класифікації
- Розробка алгоритму для кластеризації та класифікації багів в регресійному тестуванні
- Створення архітектури системи для автоматизованого аналізу даних
- Дослідження якості розробленої системи

Алгоритм класифікації для автоматичного аналізу даних в регресійному тестуванні

Процес класифікації багів пропонується провести в 5 етапів:

1. Передобробка даних
2. Індексція тестових логів
3. Визначення ваги ознак логів
4. Побудова класифікатора
5. Отримання результатів та оцінка якості класифікації



Переобробка даних та індексація тестових логів

Попередня обробка тестових логів включає в себе токенізацію(розбиття тексту на більш маленькі частини, токени), нормалізацію (видалення знаків пунктуації, чисел, артиклів тощо).

Для індексації тестових логів використовувався механізм, який представляє Elasticsearch. Кожен лог представлений у JSON форматі

```
{
  "_index":    "log_1",
  "_type":    "001A",
  "_id":      "4432de34-124b-489e-99b8-fdc04cedaa2a",
  "message":  "at NUnit.Framework.Assert.Fail(String message, Object[] args) at NUnit.Framework.Assert.Fail(String message) at Pages
               .Pages.WaitForElementNotPresent(By element, Int32 timeout) in C:\src\ne\Testing\Selenium Pages\Pages\Pages.cs:line 1786"
}
```


Визначення ваги ознак логів

TF-IDF — статистичний показник, що використовується для оцінки важливості слів у контексті документа, що є частиною колекції документів чи корпусу. Вага (значимість) слова пропорційна кількості вживань цього слова у документі, і обернено пропорційна частоті вживання слова у інших документах колекції.

$$TF = \frac{n_{t,d}}{n_d}$$

$$IDF = \log \frac{|D|}{D_t}$$

$$V_{t,d} = TF \cdot IDF$$

Метод k-найближчих сусідів

$$Q_j = \sum_{i=1}^n \frac{1}{d(x, a_i)^2},$$

$$x_k = \frac{\sum_{i=1}^n k_i d(x, a_i)^2}{\sum_{i=1}^n d(x, a_i)^2},$$

- Переваги методу:
 - можливість відновити навчальну вибірку без перенавчання класифікатора
 - стійкість алгоритму до аномальних викидів у вихідних даних
 - легко зрозуміти результати роботи алгоритму
- Недоліки методу:
 - Пошук найближчого сусіда передбачає порівняння класифікуемого об'єкта з усіма об'єктами вибірки, що збільшує трудомісткість

АНАЛІЗ РЕЗУЛЬТАТІВ КЛАСИФІКАЦІЇ

Тип вибірки	Accuracy	Precision	Recall
Тестова	0,97	0,97	0,96
Реальна	0,95	0,94	0,94

Отже, як можна побачити з таблиці тестова вибірка демонструє кращі результати чим реальна.

ДЕМОНСТРАЦІЯ РОБОТИ ПРОДУКТУ КЛАСИФІКАЦІЇ

NAME	START	TTL	PS	FL	SKP	PRODUCT BUG	AUTO BUG	SYSTEM ISSUE	TO INVEST	
NUnit 3 Demo Launch #28 🕒 46m 50s 👤 superadmin 🛑 api demo stopped this is description stopped	21 days ago	2356	138	2218					1126	☐
NUnit 3 Demo Launch #27 🛑 api demo	24 days ago	2500	170	2330					1565	☐
Move to debug Force finish Analysis Delete EXPORT: PDF XLS HTML NUnit 3 Demo Launch #26 🛑 api demo	a month ago	100	18	82					82	☐
NUnit 3 Demo Launch #25 🕒 21m 22s 👤 superadmin 🛑 api demo stopped this is description stopped	a month ago	3000	555	2445					721	216 ☐
NUnit 3 Demo Launch #24 🕒 3s 👤 superadmin 🛑 api demo this is description	a month ago	500	131	369					369	☐

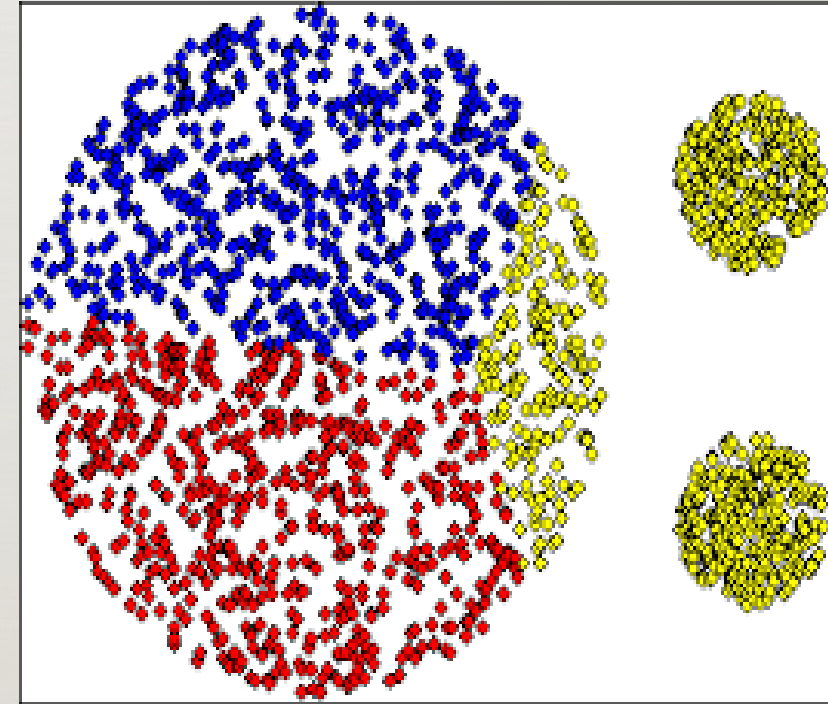
AttributesSwitchNumbering					
Test	SwitchNumbering	FAILED	a month ago	To Investigate	☐
ClosingSetIndentOutdent / IndentOutdentTest					
Test	IndentOutdentTest(Indent,"a")	FAILED	a month ago	To Investigate	☐
Test	IndentOutdentTest(Outdent,"2.")	FAILED	a month ago	To Investigate	☐
DownloadClosingSet					
Test	DownloadClosingSetTest	FAILED	a month ago	To Investigate	☐
DuplicateTest					
Test	DuplicatePlaceholderTest	FAILED	a month ago	To Investigate	☐
NumberingClosingSet / NumberingUpperrome					
Test	NumberingUpperrome(upperrome,"I.", "AddDocTest",document)	FAILED	a month ago	To Investigate	☐
Test	NumberingUpperrome(lowerrome,"ii.", "header",header)	FAILED	a month ago	To Investigate	☐
Test	NumberingUpperrome(onedigit,"3.", "[placeholder]",header)	FAILED	a month ago	To Investigate	☐
Test	NumberingUpperrome(twodigits,"02.", "header",header)	FAILED	a month ago	To Investigate	☐

AddHeaderInWorkArea							
Test	AddHeaderByClickTick 🕒 0.49s	FAILED	a month ago	AA	System Issue	☐	
Test	AddHeaderByDragAndDrop 🕒 20s	FAILED	a month ago	AA	System Issue	☐	
AttributesSwitchNumbering							
Test	SwitchNumbering 🕒 49s	FAILED	a month ago	AA	System Issue	☐	
CloneClosingSet							
Test	CloneClosingSetTest 🕒 58s	FAILED	a month ago	AA	Automation Bug	☐	
ClosingSetIndentOutdent / IndentOutdentTest							
Test	IndentOutdentTest(Indent,"(a)") 🕒 0.31s	FAILED	a month ago	AA	Product Bug	☐	
Test	IndentOutdentTest(Outdent,"2.") 🕒 0.23s	FAILED	a month ago	AA	Product Bug	☐	
DownloadClosingSet							
Test	DownloadClosingSetTest 🕒 52s	FAILED	a month ago	AA	System Issue	☐	
DuplicateTest							
Test	DuplicatePlaceholderTest 🕒 21s	FAILED	a month ago	AA	Product Bug	☐	

Алгоритм кластеризації для автоматичного аналізу даних в регресійному тестуванні

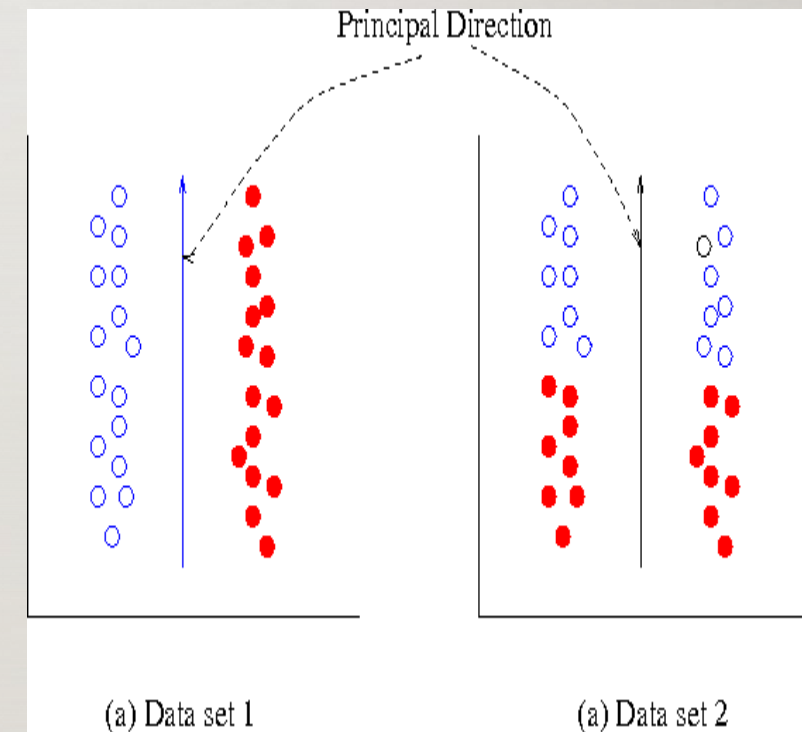
Процес кластеризації багів пропонується провести в 5 етапів:

1. Підготовка даних
2. Індексація тестових логів
3. Створення матриці індексів
4. Вибір метрики
5. Розбиття на кластери



МАТЕМАТИЧНІ ОСНОВИ РІШЕНЬ ДЛЯ КЛАСТЕРИЗАЦІЇ

- **Concept Indexing.** ФОРМУЛИ



АРХІТЕКТУРА ПРОГРАМНОГО ПРОДУКТУ



ДЕМОНСТРАЦІЯ РОБОТИ ПРОДУКТУ

 METHOD TYPE	NAME	STATUS	START TIME ^	DEFECT TYPE	☐
AddDocumentTest					
Test	AddDocumentToPlaceHolder  41s	FAILED	a month ago	 To Investigate	☐
ClosingSetIndentOutdent / IndentOutdentTest					
Test	IndentOutdentTest(Indent,"(a)")  0s	FAILED	a month ago	 To Investigate	☐
Test	IndentOutdentTest(Outdent,"2.")  0s	FAILED	a month ago	 To Investigate	☐
CommentCheck					
Test	CommentCheckTest  1m	FAILED	a month ago	 To Investigate	☐
DuplicateTest					
Test	DuplicatePlaceholderTest  22s	FAILED	a month ago	 To Investigate	☐
SelectorCheck / SelectorCheckTest					
Test	SelectorCheckTest(all,9)  0s	FAILED	a month ago	 To Investigate	☐
Test	SelectorCheckTest(headers,3)  0s	FAILED	a month ago	 To Investigate	☐
Test	SelectorCheckTest(placeholders,4)  0s	FAILED	a month ago	 To Investigate	☐

Base for Auto Analysis

☐ Launches with the same name

The test items are analyzed on base of previously investigated data in launches with the same name

☒ All launches

The test items are analyzed on base of previously investigated data in all launches

Mode of Auto-Analyse Accuracy

Classic

Moderate

Light

Minimum should match

95

%

Percent of words equality between analyzed log and particular log from the ElasticSearch. If a log from ElasticSearch has the value less then set, this log will be ignored for AA.

Minimum document frequency

1

Set the minimum frequency of the saved logs in ElasticSearch (index) in which word from analyzed log should be used. If the log count is below the specified value, that word will be ignored for AA in the analyzed log. The more often the word appears in index, the lower it weights.

Minimum term frequency

1

Set the minimum frequency of the word in the analyzed log. If the word count is below the specified value, this word will be ignored for AA. The more often the word appears in the analyzed log, the higher it weights.

Number of log lines

All

The number of first lines of log message that should be considered in ElasticSearch.

Submit



АНАЛІЗ РЕЗУЛЬТАТІВ КЛАСТЕРИЗАЦІЇ

Тип вибірки	Accuracy	Precision	Recall
Тестова	0,97	0,97	0,96
Реальна	0,95	0,94	0,94

ПОКРАЩЕННЯ РОБОТИ ПРОГРАМИ

Кількість запитів	Час відправки запитів з кешуванням	Час відправки запитів без кешування
1000	45 секунд	51 секунда
2500	110 секунд	123 секунд
5000	205 секунд	224 секунди

ВИСНОВКИ

- Запропоновано алгоритм класифікації багів в регресійному тестуванні
- Запропоновано алгоритм кластеризації багів в регресійному тестуванні
- Запропонована архітектура системи для автоматизованого аналізу даних
- Реалізовано програмний продукт, який працює для задачі кластеризації з такою то точністю та класифікації з такою то точністю
- Дозволяє зекономити час в n разів

ШЛЯХИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

- Розпізнавання дуже схожих за семантичною структурою багів
- Підвищення точності класифікації та кластеризації
- Додати навчання на основі попередніх прогонів

ДЯКУЮ ЗА УВАГУ!