

Relazione Ingegneria del Software 2

Machine Learning for Software Engineering

Deliverable 3

Camilli Michela 0286047

Introduzione

Descrizione del problema

Applicazione

Passo 1

Passo 2

Passo 3

Passo 4

Passo 5

Passo 6

Passo 7

Risultati ottenuti

Passo 8

Questa Deliverable ha lo scopo di fornire uno scenario di utilizzo del CBAM, che rappresenta un esempio di Decision Making Technique. Questa tecnica consta di più passi, la cui realizzazione è stata spiegata nelle slide successive.

Il problema considerato, per quest'applicazione, è stato quello relativo alla scelta del linguaggio di programmazione da utilizzare per il progetto di Ingegneria del Software 1, nel corso della triennale. Le alternative per questa decisione sono state:

- Java
- Python
- C
- C++

PASSO 2

Definizione degli attributi di qualità



Sono stati, quindi, definiti dei criteri di decisione, assegnando loro dei punteggi di importanza. Questi ultimi sono stati selezionati in modo tale che la loro somma desse 100.

Tra i possibili attributi, quelli selezionati sono stati:

- Performance: 20
- Conoscenza: 25
- Espressività: 15
- Robustezza: 15
- Portabilità: 5
- Debugging: 10
- Modularità: 10

PASSO 3

Rating delle alternative



A ciascuna alternativa è stato, quindi, assegnato un punteggio tra [-1, 1] per ciascun attributo di qualità. In seguito sono mostrati i valori assegnati per ogni linguaggio.

PASSO 3

Rating delle alternative: **JAVA**



6

Per quanto riguarda **Java**, i punteggi assegnati sono stati:

- Performance: 0.6
- Conoscenza: 0.9
- Espressività: 0.8
- Robustezza: 0.4
- Portabilità: 0.9
- Debugging: 0.5
- Modularità: 0.8

PASSO 3

Rating delle alternative: **Python**



Per quanto riguarda **Python**, i punteggi assegnati sono stati:

- Performance: 0.7
- Conoscenza: 0.7
- Espressività: 0.8
- Robustezza: 0.5
- Portabilità: 0.8
- Debugging: 0.7
- Modularità: 0.7

PASSO 3

Rating delle alternative: **C**



Per quanto riguarda **C**, i punteggi assegnati sono stati:

- Performance: 0.9
- Conoscenza: 0.6
- Espressività: 0.6
- Robustezza: 0.8
- Portabilità: 0.5
- Debugging: 0.3
- Modularità: 0.8

PASSO 3

Rating delle alternative: **C++**



Per quanto riguarda **C++**, i punteggi assegnati sono stati:

- Performance: 0.6
- Conoscenza: 0.2
- Espressività: 0.7
- Robustezza: 0.6
- Portabilità: 0.5
- Debugging: 0.4
- Modularità: 0.8

Per ciascuna alternativa è stato, poi, assegnato un valore di *rischio* in un intervallo $[0,1]$, tenendo conto di diversi fattori, tra cui: il grado di conoscenza del linguaggio e la difficoltà nell'utilizzo.

Sono stati, quindi, associati alle alternative i valori:

- Java: 0.5
- Python: 0.6
- C: 0.8
- C++: 0.8

Per ciascuna alternativa è stato, quindi, calcolato il punteggio di *benefit*, in base ai valori ottenuti in precedenza. La formula utilizzata è stata:

$$Benefit(AS_i) = (\sum_j (AS_{ij} * QAscore_j)) * |Risk_i - 1|$$

I risultati ottenuti sono stati:

- **Java:** 35
- **Python:** 31.6
- **C:** 13.5
- **C++:** 10.2

A ciascuna alternativa è stato assegnato un *costo* in \$, tenendo in conto il dispendio, in termini di tempo, necessario alla programmazione (in un'ora) con le varie alternative considerate. I valori assegnati sono stati:

- **Java:** 11\$
- **Python:** 12\$
- **C:** 17\$
- **C++:** 15\$

PASSO 7

Calcolo della desirability



13

Per ciascuna alternativa è stata, infine, calcolata la *Desirability*, in base ai valori ottenuti in precedenza. La formula utilizzata è stata:

$$Desirability(AS_i) = Benefit(AS_i) / Cost(AS_i)$$

I risultati ottenuti, ponendo la *Desirability* in ordine decrescente sono stati:

- **Java:** 3.2
- **Python:** 2.6
- **C:** 0.8
- **C++:** 0.7