



Master Test Plan

Progetto ReFair

Riferimento	4_Master_Test_Plan
Versione	1.0
Data	03/07/2024
Presentato da	Daniele Galloppo, Luca Morelli, Mario Peluso



Team Members

Nome	Acronimo	Informazioni di contatto
Daniele Galloppo	DG	d.galloppo@studenti.unisa.it
Luca Morelli	LM	l.morelli6@studenti.unisa.it
Mario Peluso	MP	m.peluso37@studenti.unisa.it



Summary

1. Obiettivi e contesto	4
2. Strategie di Testing	5
2.1. Testing di unità	5
2.2. Testing di integrazione	5
2.3. Testing di sistema	5
3. Attività e classi di test	7
3.1. Funzionalità e classi testate	7
3.2. Funzionalità e classi escluse dal testing	7
3.3. Obiettivi di coverage	7



1. Obiettivi e contesto

Nome del progetto: ReFair.

Descrizione del progetto: modello di classificazione che mira ad identificare e suggerire all'utente sensitive features che se non trattate correttamente potrebbero portare a modelli biased.

Obiettivo del testing: garantire la qualità del prodotto e verificare la correttezza delle funzionalità offerte.



2. Strategie di Testing

2.1. Testing di unità

La strategia applicata per tale attività sarà la tecnica white box, utilizzando la coverage come indicatore. Obiettivo sarà quello di coprire branch scoperti selezionando input adeguati che possano raggiungerli.

Le funzionalità coperte sono:

- loadStories
- analysis
- reportStories
- reportStory
- intersection
- featureExtraction

2.2. Testing di integrazione

Tale attività fa riferimento a test che mirano alla combinazione di componenti, verificandone la corretta interazione.

Le funzionalità coperte sono:

- analysis
- reportStories
- reportStory
- getDomain
- getMLTask
- refair

2.3. Testing di sistema

La strategia applicata per tale attività sarà la tecnica black box, nello specifico category partitioning. Il campo dei dati sarà quindi suddiviso in classi di test e verrà poi applicata la tecnica WECT al fine di minimizzare il numero di possibili combinazioni. Saranno inoltre escluse le combinazioni impossibili.



Essi verranno accompagnati da test ad hoc che mireranno a studiare specifici comportamenti del sistema che non rientrano nell'attività di category partition. Esempio sono i test che mirano a controllare il comportamento di buttons e test che mirano a confrontare i file .json ottenuti mediante le analisi del modello.



3. Attività e classi di test

3.1. Funzionalità e classi testate

Le funzionalità contenute lato back-end saranno sottoposte a test di unità ed a testing di integrazione. Non verrà effettuato testing di unità lato front-end in quanto le funzionalità vengono già ampiamente coperte durante il testing di sistema.

3.2. Funzionalità e classi escluse dal testing

Saranno escluse dal testing le funzionalità che richiamano i modelli di machine learning.

3.3. Obiettivi di coverage

Tenute in considerazione il fatto che il tool si basa principalmente su modelli di machine learning e che la percentuale di coverage è attualmente pari a 0%, ci poniamo di raggiungere una percentuale di coverage $\geq 60\%$.