



Laurea Magistrale in informatica  
Università degli Studi di Salerno

# Unit Test Plan

## QuantuMoonLight

### Team

Matteo Cicalese - 0522501516  
Luca Contrasto – 0522500104

### Repo

[github.com/CicaMatt/QML-IGES](https://github.com/CicaMatt/QML-IGES)  
[github.com/Robertales/QuantuMoonLight](https://github.com/Robertales/QuantuMoonLight)

## Sommario

1. Approccio di test .....	3
1.1. Change Request 1 .....	3
1.2. Change Request 2 .....	4
1.3. Change Request 3 .....	5
1.4. Change Request 4 .....	8
2. Coverage raggiunta .....	10

## 1. Approccio di test

---

Il testing di unità sarà condotto sulla base delle funzionalità sviluppate nel corso delle change request, dunque cercheremo, nel modo più esaustivo possibile, di testare in maniera unitaria tutte le sfaccettature delle funzionalità aggiunte e modificate, valutando che il risultato restituito dall'esecuzione corrisponda con quello atteso, e cercando di coprire tutte le possibili situazioni che si possono verificare.

### 1.1. Change Request 1

---

Nel caso di questa change request non è stato possibile realizzare degli unit test ad-hoc, in quanto l'aggiornamento della libreria coinvolge il progetto nella sua totalità, ed è stata realizzata soprattutto per fornire una funzionalità fondamentale nel contesto dell'applicazione, quale il salvataggio dei modelli (sia quantistici che classici) pre-addestrati. Questa modifica viene testata nei test di unità relativi a tutti i modelli di classificazione, in quanto, in ognuno di essi, è presente un assert riguardo la presenza del file .model risultante dal salvataggio del modello pre-addestrato quantistico e il file .sav nel caso in cui il modello non sia quantistico.

Un'ulteriore motivazione che ha incentivato l'implementazione di questa change request è che, come riportato anche nel Test Incident Report, il modello di classificazione quantistica QSVM presente nell'applicazione è stato deprecato, e non risulta più utilizzabile, per cui le librerie precedenti alla change request, così come le funzionalità correlate a tale modello quantistico, risultano in parte inconsistenti.

Tramite Regression Testing sarà verificato che l'aggiornamento delle librerie non abbia introdotto stati di errore e quindi malfunzionamenti all'applicazione, runnando i relativi test già presenti e opportunamente modificati. Inoltre, sarà verificato che negli unit test associati ad ogni classificatore sarà presente il file del modello addestrato.

## 1.2. Change Request 2

**Metodo testato:** *experimentDownload (UtenteControl)*

**Nomi metodo di test:** *test\_Download (test\_UtenteControl)*

In *test\_download* inviamo la richiesta POST */experimentDownload* al server, con ID esperimento uguale a 1, e viene controllato che:

- lo status code restituito corrisponde a 200
- il content type della risposta corrisponde a “application/x-zip-compressed”
- il file .zip generato è valido e integro

Richiesta	Oracolo
<i>/experimentDownload</i> - <i>explD = Integer</i>	<i>response.status_code = 200</i> <i>response.content_type = “application/x-zip-compressed”</i>
Risposta	<i>zip_ref = False</i>
File zip contenente tutti i file relativi all’esperimento	

**Metodo testato:** *experimentDownload (UtenteControl)*

**Nomi metodo di test:** *test\_Download\_invalidPath (test\_UtenteControl)*

In *test\_Download\_invalidPath* inviamo la richiesta POST */experimentDownload* al server, con ID esperimento uguale a 0 (che non può esistere, dunque non valido), e viene controllato che:

- lo status code restituito corrisponde a 500, che corrisponde ad uno status di errore

Richiesta	Oracolo
<i>/experimentDownload</i> - <i>explD = Integer</i>	<i>response.status_code = 500</i>
Risposta	
Error 500	

### 1.3. Change Request 3

---

**Metodo testato:** *encrypt (encryption)*

**Nomi metodo di test:** *test\_encrypt (test\_utils)*

In *test\_encrypt* viene preso il percorso in cui sono presenti i file da cifrare (che hanno rispettivamente estensione .csv, .txt, .xlsx, .png e .sav) e viene letta una chiave prestabilita. Successivamente, viene effettuato un tentativo di cifratura, e si verifica sia che i rispettivi file esistano nel formato cifrato, sia che il contenuto dei file decifrati corrisponda a quello originale.

Input	Oracolo
Path	Path_csv_file.is_file() = True
Key	Path_txt_file.is_file() = True
Risultato	Path_xlsx_file.is_file() = True
File cifrati all'interno del path dato in input	Path_png_file.is_file() = True Path_sav_file.is_file() = True

**Metodo testato:** *encrypt (encryption)*

**Nomi metodo di test:** *test\_encrypt\_fail (test\_utils)*

In *test\_encrypt\_fail* viene preso il percorso in cui sono presenti i file da cifrare (che hanno rispettivamente estensione .csv, .txt, .xlsx, .png e .sav) e viene letta una chiave prestabilita, la quale viene troncata appositamente per renderla non valida. Successivamente, viene effettuato un tentativo di cifratura, e si verifica che esso abbia lanciato un'eccezione e che i rispettivi file nel formato cifrato non siano presenti.

Input	Oracolo
Path	Exception raised
Key	Path_csv_file.is_file() = False
Risultato	Path_txt_file.is_file() = False
File cifrati all'interno del path dato in input assenti	Path_xlsx_file.is_file() = False Path_png_file.is_file() = False Path_sav_file.is_file() = False

**Metodo testato:** *decrypt (encryption)*

**Nomi metodo di test:** *test\_decrypt (test\_utils)*

In *test\_decrypt* viene preso il percorso in cui sono presenti i file da decifrare (che hanno rispettivamente estensione .csv, .txt, .xlsx, .png e .sav) e viene letta la chiave prestabilita (la stessa utilizzata per la fase di cifratura). Successivamente, viene effettuato un tentativo di decifratura, e si verifica che i rispettivi file nel formato decifrato originale siano presenti.

Input	Oracolo
Path	Path_csv_file.is_file() = True
Key	Path_txt_file.is_file() = True
Risultato	Path_xlsx_file.is_file() = True
File decifrati all'interno del path dato in input	Path_png_file.is_file() = True Path_sav_file.is_file() = True

**Metodo testato:** *decrypt (encryption)*

**Nome metodo di test:** *test\_decrypt\_fail (test\_utils)*

In *test\_decrypt\_fail* viene preso il percorso in cui sono presenti i file da decifrare (che hanno rispettivamente estensione .csv, .txt, .xlsx, .png e .sav) e viene letta una nuova chiave (la stessa utilizzata per la fase di cifratura). Successivamente, viene effettuato un tentativo di decifratura, e si verifica che il tentativo di decifratura lanci un'eccezione e che i rispettivi file nel formato decifrato originale siano presenti.

Input	Oracolo
Path	Exception raised
Key	Path_csv_file.is_file() = True
Risultato	Path_txt_file.is_file() = True
File decifrati all'interno del path dato in input	Path_xlsx_file.is_file() = True Path_png_file.is_file() = True Path_sav_file.is_file() = True

## 1.4. Change Request 4

**Metodo testato:** *SetNewPW(UtenteControl)*

**Nome metodo di test:** *test\_SetNewPW((test\_UtenteControl)*

In *test\_setNewPW* viene inviata la richiesta verso la route */SetNewPW*, con parametri l'email dell'utente a cui verrà modificata la password dell'account e la nuova password da impostare. Si verifica che la richiesta ha successo controllando lo status code della risposta.

Input	Oracolo
Email, newPassword	Response.status_code == 200
Risultato	
Password dell'utente aggiornata	

**Metodo testato:** *resetPW (UtenteControl)*

**Nomi metodo di test:** *test\_sendCode (test\_UtenteControl)*

In *test\_sendCode* viene inviata la richiesta verso la route */sendCode*, con parametro l'email dell'utente a cui verrà inviata la mail con il codice per il reset della password che è la stessa associata all'account. Si verifica che la richiesta ha successo controllando lo status code della risposta.

Input	Oracolo
Email	Response.status_code == 200
Risultato	
Email inviata all'utente	



**Metodo testato:** *resetPW (UtenteControl)*

**Nome metodo di test:** *test\_sendCode\_emailNotFoundError (test\_UtenteControl)*

In *test\_sendCode* viene inviata la richiesta verso la route */sendCode*, con parametro l'email di un utente non registrato. La verifica del fallimento della richiesta avviene controllando lo status code della risposta.

Input	Oracolo
Email non presente nel database	Response.status_code == 400
Risultato	
Error message	

**Metodo testato:** *resetPW (UtenteControl)*

**Nome metodo di test:** *test\_email\_sending\_failure (test\_UtenteControl)*

In *test\_email\_sending\_failure* viene simulato un errore generale, non dipendente dai parametri della richiesta, che causa il mancato invio dell'email. La verifica avviene tramite la simulazione del lancio di un'eccezione dal metodo testato.

Input	Oracolo
/	Exception is raised
Risultato	
Error message	

## 2. Coverage raggiunta

---

La percentuale di coverage raggiunta successivamente all'implementazione dei test delle change request risulta del 78% per la branch coverage e 85% line coverage. Il risultato è da considerarsi soddisfacente in quanto rispetta il vincolo definito nel **Master Test Plan**, oltre al fatto che i test ed il codice stesso hanno subito delle modifiche volte al miglioramento della coerenza e aggiornamento tra test e codice implementato.