



Laurea Magistrale in informatica  
Università degli Studi di Salerno

# Pre-CR

# Test Incident Report

# QuantuMoonLight

## Team

Matteo Cicalese - 0522501516  
Luca Contrasto – 0522501512

## Repo

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

## Sommario

1. Introduzione .....	3
2. Test di integrazione/unità aggiunti .....	3
3. Test di integrazione/unità rimossi .....	4
4. Miglioramenti adottati .....	4
5. Incident Report.....	5
5.1. Incident Report STI1 .....	5
5.2. Incident Report UTI1 .....	6
5.3. Incident Report UTI2 .....	7
5.4. Incident Report UTI3 .....	8

## 1. Introduzione

---

Nel presente report saranno dettagliati i difetti e le problematiche riscontrate durante l'esecuzione dei casi di test sia a livello di sistema che di unità ed integrazione.

L'obiettivo del documento è quindi fornire un'analisi dei fault emersi durante il test preliminare dell'applicazione e di illustrare le azioni risolutive intraprese con lo scopo di risolverli.

## 2. Test di integrazione/unità aggiunti

---

- *Test\_ClassifyControl*
  - *Test\_classify*: testa l'algoritmo di classificazione SVC
  - *Test\_classify\_QSVC*: testa l'algoritmo di classificazione QSVC
  - *Test\_classify\_NNClassifier*: testa l'algoritmo di classificazione NeuralNetworkClassifier
  - *Test\_classify\_Pegasos\_QSVC*: testa l'algoritmo di classificazione Pegasos QSVC
- *Test\_routes*
  - *Test\_routes1*: testa l'integrazione di validazione (SimpleSplit), preprocessing(data imputation, Standard Scaling, Feature Extraction, PrototypeSelection) e classificazione (SVC)
  - *Test\_routes2*: testa l'integrazione di validazione (SimpleSplit), preprocessing (MinMax Scaling, Feature Selection, PrototypeSelection, data balancing) e classificazione (SVR)
  - *Test\_routes3*: testa l'integrazione di validazione (KFold), preprocessing (Feature Extraction)
  - *Test\_routes4*: testa l'integrazione di validazione (SimpleSplit), preprocessing(Standard Scaling, Feature Extraction, PrototypeSelection) e classificazione (SVC)

### 3. Test di integrazione/unità rimossi

---

- *Test\_routes*
  - *Test\_routes*: utilizza l'algoritmo QSVM deprecato ed è stato sostituito da più test di routes che testano l'integrazione di diverse componenti
- *Test.Validazione*
  - *test\_ValidazioneControl\_KFold\_SimpleSplit*: non è più necessario effettuare questo test in quanto le due variabili booleane che identificavano i metodi di validazione SimpleSplit e KFold sono state sostituite da un'unica stringa, *validation*, che ha valore "SimpleSplit" e "KFold" a seconda dell'algoritmo desiderato

### 4. Miglioramenti adottati

---

Sono stati necessari alcuni miglioramenti di aspetto generale da adottare ai test, in quanto la loro introduzione ha indotto ad un notevole miglioramento della qualità dei test.

In particolare, le azioni compiute sono state le seguenti:

- Sostituzione di *test\_all*, che conteneva una copia di tutti i test sviluppati nelle singole suite associate ad ogni componente del sistema con un'istruzione che consente l'autodiscovery ed esecuzione dei test importati. L'applicazione di questa miglioria ha sollevato un altro problema: la risoluzione dei percorsi dei file di test generati non era corretta in quanto venivano inseriti sempre nella directory *testingFiles* contenuta nella singola componente e non nella directory *testingFiles* contenuta nella directory padre di *test\_all*. La risoluzione è stata l'aggiunta del metodo *resolve()* della libreria *Path* che riconosce correttamente il file .py da cui parte l'esecuzione dei test e di conseguenza fa riferimento alla directory *testingFiles* sia quando i test vengono eseguiti dalla singola componente che quando vengono eseguiti da *test\_all*.
- Eliminazione di dipendenze, in quanto diversi test effettuavano controlli su righe inserite nel database da altri test eseguiti in precedenza, di conseguenza sono stati creati metodi di *setup* e *teardown* sia a livello di classe che di metodo in modo da non causare mai dipendenze tra i test.

## 5. Incident Report

---

### 5.1. Incident Report STI1

---

- **Informazioni:**

- Progetto: "QuantuMoonLight"
- Ambiente: Ambiente di test
- Nome Incident: STI1 (System Testing Incident 1)
- Test Case: TC\_3\_4, TC\_3\_5, TC\_3\_6
- Priorità: Alta
- Gravità: Critica

- **Descrizione dell'Incidente:**

Durante il testing preliminare di sistema volto a testare l'effettiva correttezza delle funzionalità, abbiamo riscontrato il seguente problema:

Caricando un dataset vuoto in uno qualsiasi degli input disponibili il sistema continua l'esecuzione e non lancia un messaggio d'errore

- Passi per Riprodurre:

- Caricare un file .csv o .txt vuoto come dataset di Train, Test o Prediction;
- Eseguire l'upload.

- Comportamento Attuale: l'applicazione continua a funzionare e viene effettuato un redirect alla pagina del download.

- **Risoluzione e Pianificazione:**

- Stato: chiuso

- Cause Identificate: quando vengono letti i file dal tag input non viene effettuato un controllo sulla dimensione quindi anche un file vuoto viene accettato dall'app e l'esecuzione continua

- Componente Coinvolto: *routes.py*

- Pianificazione: la modifica è minima e coinvolge una singola classe quindi non riteniamo di dover eseguire Impact analysis.

- Soluzione: viene inserito un controllo sulla dimensione dei file inseriti, se uguale a zero l'esecuzione si ferma e viene lanciato un messaggio d'errore che sarà mostrato sulla pagina del caricamento.

## 5.2. Incident Report UTI1

---

- **Informazioni:**

- Progetto: "QuantuMoonLight"
- Ambiente: Ambiente di test
- Nome Incident: UTI1 (Unit Testing Incident 1)
- Test Case: test\_getClassifiedDataset
- Priorità: Alta
- Gravità: Media

- **Descrizione dell'Incidente:**

Durante il testing di unità del sistema abbiamo riscontrato il seguente problema:

Dopo l'elaborazione del dataset e il redirecting alla pagina di download dei risultati non viene inviata la mail contenente il risultato della classificazione quando selezionata.

○ Passi per Riprodurre:

- Caricare un file .csv o .txt vuoto come dataset di Train e un dataset per la Prediction;
- Selezionare un algoritmo di classificazione;
- Eseguire l'upload.

○ Comportamento Attuale: l'applicazione continua a funzionare e viene effettuato un redirect alla pagina del download ma il risultato della classificazione non viene inviato per email

- **Risoluzione e Pianificazione:**

- Stato: chiuso
- Cause Identificate: la password della mail mittente quantummoonlight@gmail.com non è stata aggiornata dopo averla modificata
- Componente Coinvolto: *ClassifyControl*
- Pianificazione: la modifica è minima e coinvolge una singola classe quindi non riteniamo di dover eseguire Impact Analysis
- Soluzione: è stata creata una "Password per App" da Gmail in modo che l'autenticazione necessaria per l'invio della mail nell'applicazione non dipende dalla password della mail e da eventuali sue modifiche

## 5.3. Incident Report UTI2

---

- **Informazioni:**

- Progetto: "QuantuMoonLight"
- Ambiente: Ambiente di test
- Nome Incident: UTI2 (Unit Testing Incident 2)
- Test Case: test\_routes
- Priorità: Alta
- Gravità: Critica

- **Descrizione dell'Incidente:**

Durante il testing di unità del sistema abbiamo riscontrato il seguente problema:

Quando vengono salvati nel database dati relativi all'esperimento l'esecuzione si ferma poiché l'applicazione va in errore.

○ Passi per Riprodurre:

- Caricare un file .csv o .txt come dataset di Train;
- Eseguire l'upload.

○ Comportamento Attuale: l'esecuzione dell'applicazione termina con "Internal Server Error"

- **Risoluzione e Pianificazione:**

- Stato: chiuso
- Cause Identificate: l'attributo name della tabella Dataset ha il limite di 30 caratteri che sono troppo pochi per salvare un path
- Componente Coinvolto: *models*
- Pianificazione: la modifica è minima e coinvolge una singola classe quindi non riteniamo di dover eseguire Impact Analysis
- Soluzione: è stata modificata la dimensione dell'attributo impostandola a 255

## 5.4. Incident Report UTI3

---

- **Informazioni:**

- Progetto: "QuantuMoonLight"
- Ambiente: Ambiente di test
- Nome Incident: UTI1 (Unit Testing Incident 3)
- Test Case: *test\_classification\_thread*, *test\_classify*, *test\_classify\_ibmFail*, *test\_PreprocessingControl\_FE\_QSVM*, *test\_routes*
- Priorità: Alta
- Gravità: Critica

- **Descrizione dell'Incidente:**

Durante il testing di unità del sistema abbiamo riscontrato il seguente problema:

Quando viene caricato l'esperimento impostando l'algoritmo QSVM di classificazione quantistica il sistema restituisce "Internal Server Error".

○ Passi per Riprodurre:

- Caricare un file .csv come dataset di Train e un dataset per la Prediction;
- Selezionare l'algoritmo di classificazione QSVM;
- Eseguire l'upload.

○ Comportamento Attuale: l'applicazione restituisce "Internal Server Error"

- **Risoluzione e Pianificazione:**

- Stato: chiuso
- Cause Identificate: poiché QSVM appartiene ad una versione deprecata di qiskit machine learning non viene più attivata un'istanza su QuantumIBM utile alla classificazione quantistica quindi la variabile provider non viene valorizzata correttamente e l'applicazione genera un fault
- Componente Coinvolto: *ClassifyControl*
- Pianificazione: la modifica è minima e coinvolge una singola classe quindi non riteniamo di dover eseguire Impact Analysis
- Soluzione: è necessario eliminare il classificatore QSVM in quanto non più utile e perché è stato sostituito da una versione più veloce QSVC nella versione più aggiornata della libreria