

Maintenance Report & Impact Analysis

QuantuMoonLight

Team

Matteo Cicalese - 0522501516

Luca Contrasto – 0522500104

Repo

github.com/CicaMatt/QML-IGES

github.com/Robertales/QuantuMoonLight

Sommario

- Sommario..... 2
- 1. Change Request 1 (CR_1)..... 3
 - 1.1 Analisi d’impatto 4
 - 1.2 Implementazione modifiche 6
 - 1.3 Metriche 8
- 2 Change Request 2 (CR_2)..... 9
 - 2.1 Analisi d’impatto 10
 - 2.2 Implementazione modifiche 11
 - 2.3 Metriche 13
- 3 Change Request 3 (CR_3)..... 14
 - 3.1 Analisi d’impatto 15
 - 3.2 Implementazione modifiche 16
 - 3.3 Metriche 17
- 4 Change Request 4 (CR_4)..... 18
 - 4.1 Analisi d’impatto 20
 - 4.2 Implementazione modifiche 21
 - 4.3 Metriche 22

1. Change Request 1 (CR_1)

La presente change request è focalizzata sull'aggiornamento delle librerie di Qiskit, in modo congiunto all'upgrade della versione di Python, oltre ad introdurre la possibilità di effettuare il salvataggio dei modelli quantistici.

Situazione Attuale

Al momento, le versioni delle librerie in uso sono qiskit-terra 0.19.1 e qiskit-machine-learning 0.3.1, mentre la versione di Python è la 3.7. Tali versioni sono considerate obsolete, mancando di diverse funzionalità e ottimizzazioni presenti nelle versioni più recenti. Queste ultime, a loro volta, richiedono una versione di Python pari o superiore alla 3.9.

Proposta di Modifica

La proposta è di aggiornare le librerie di Qiskit alle versioni più recenti, in modo particolare le librerie qiskit-terra alla versione 0.25.2.1 e qiskit-machine-learning alla versione 0.6.1. Simultaneamente, si suggerisce di effettuare un upgrade della versione di Python alla 3.9, al fine di sfruttare a pieno le nuove funzionalità offerte dalle librerie e di predisporre l'ambiente per futuri aggiornamenti.

Benefici

L'implementazione di questa modifica può portare a diversi benefici:

- **Fruizione di Nuove Funzionalità:** Le nuove versioni delle librerie qiskit-terra e qiskit-machine-learning offrono nuove funzionalità, come il download dei modelli quantistici. Ciò consente notevoli risparmi di tempo durante eventuali retraining dei modelli o test sullo stesso modello ma con nuovi dati. I modelli per cui sarà possibile il download saranno: SVC, K-Neighbors, Decision Tree, Naive Bayes, Random Forest, SVR, Linear Regressor. Per la versione di Qiskit attualmente installata (0.3.1) non è possibile salvare il resto dei modelli quantistici. Installando la versione 0.6.1, è possibile salvare anche QSVC, PegasosQSVC, NeuralNetworkClassifier, QSVR, VQR.
- **Training time ridotto:** l'aggiornamento della libreria qiskit-machine-learning alla versione 0.6.1 comprende miglioramenti nel codice dei modelli ed ottimizzazioni in termini di tempo che migliora di circa il 20% il tempo necessario ad effettuare il training dei modelli quantistici.

- **Aggiornamento della Versione di Python:** L'upgrade a Python 3.9 comporta miglioramenti significativi in termini di prestazioni, risoluzione di bug, sicurezza e compatibilità. Questo assicura un ambiente più stabile e preparato per future evoluzioni.

L'attuazione di queste modifiche non solo risponde alle esigenze di mantenimento e aggiornamento del software ma fornisce anche un ambiente più efficiente e all'avanguardia per lo sviluppo e l'esecuzione delle funzionalità del sistema. In seguito, sarà specificata l'analisi d'impatto relativo a questa modifica necessaria per garantire un processo di implementazione senza intoppi.

1.1 Analisi d'impatto

Starting Impact Set – 10 Componenti

Sono state individuate le seguenti classi e metodi (sono utilizzate sigle riportate a destra del modulo):

- routes - smista
- ClassifyControl – classify_control (CC), classification_thread (CT), classify (CL)
- myQSVR – classify (QSVR)
- myQSVC – classify (QSVC)
- myQSVM – classify (QSVM)
- myPegasosQSVC – classify (PQSVC)
- myNeuralNetworkClassifier – classify (NNC)
- myNeuralNetworkRegressor – classify (NNR)

	FD	smista	CC	CT	CL	QSVR	QSVC	QSVM	PQSVC	NNC	NNR
FD	/	1	2	3							
smista		/	1	2	3						
CC			/	1	2	3	3	3	3	3	3
CT				/	1	2	2	2	2	2	2
CL					/	1	1	1	1	1	1
QSVR						/					
QSVC							/				
QSV								/			
PQSVC									/		
NNC										/	
NNR											/

Matrice di raggiungibilità

Per individuare le componenti coinvolte e stabilire il Candidate Impact Set (CIS) è stata generata la matrice di raggiungibilità considerando l'approccio distance based (fino al livello 3 di propagazione) per ridurre le occorrenze di falsi positivi.

La ragione per cui sono stati aggiornate tutte le librerie di Qiskit, senza limitarci a qiskit-terra e qiskit-machine-learning come inizialmente proposto (quelli principalmente utilizzati sulla piattaforma) sta nel fatto che queste librerie risultano avere dipendenze fra loro, oltre al fatto che si vuole fornire una base di partenza aggiornata per eventuali estensioni della piattaforma con altre funzionalità di natura quantistica non correlate solo al Machine Learning.

É stato riscontrato che le modifiche interessate da questa Change Request hanno un impatto solo sulla componente di classificazione, dove in particolare è stato aggiunto nella classe *ClassifyControl* il codice necessario al salvataggio dei classificatori/regressori. Inoltre, è stato necessario eliminare il classificatore QSVM, definito in una classe apposita *myQSVM*, in quanto deprecato nella versione aggiornata della libreria e sostituito dal classificatore QVSC per il quale era già presente la classe *myQVSC*. Per quanto riguarda le componenti del Candidate Impact Set, andrà considerato anche la pagina *formDataset*, che avrà dei nomi di

parametri diversi e alcune funzionalità modificate in seguito agli aggiornamenti delle librerie di classificazione.

Candidate Impact Set – 11 Componenti

Sono state individuate le seguenti classi e metodi:

- `templates – formDataset (FD)`
- `routes - smista`
- `ClassifyControl – classify_control (CC), classification_thread (CT), classify (CL)`
- `myQSVR – classify (QSVR)`
- `myQSVC – classify (QSVC)`
- `myQSVM – classify (QSVM)`
- `myPegasosQSVC – classify (PQSVC)`
- `myNeuralNetworkClassifier – classify (NNC)`
- `myNeuralNetworkRegressor – classify (NNR)`

1.2 Implementazione modifiche

Prima di procedere con l'aggiornamento delle librerie, abbiamo prioritariamente eseguito un upgrade della versione di base di Python, passando dalla versione 3.7 alla versione 3.9, con miglioramenti apprezzabili in termini di sicurezza e prestazioni.

Successivamente, nell'analisi delle librerie da aggiornare, abbiamo fatto attenzione a selezionare quelle con il miglior rapporto impatto/guadagno. Le librerie scelte sono state:

- **qiskit:** da 0.34.0 a 0.44.2
- **qiskit-terra:** da 0.19.1 a 0.25.2.1
- **qiskit-aer:** da 0.10.1 a 0.12.2
- **qiskit-ignis:** da 0.7.0 a 0.6.0 (downgrade per compatibilità)
- **qiskit-machine-learning:** da 0.3.1 a 0.6.1
- **qiskit-ibmq-provider:** da 0.18.3 a 0.20.2

Va riportato che, a differenza di quanto pianificato, non è stato necessario effettuare modifiche ai metodi *classify* in *myQSVR*, *myQSVC*, *myPegasosQSVC*, *myNeuralNetworkClassifier* e *myNeuralNetworkRegressor*, in quanto, dovendo solo aggiungere il codice necessario ad

attivare il download, è stato sufficiente modificare in tal senso solo il metodo `classify` di `ClassifyControl`, che gestisce in tal senso tutti i vari classificatori/regressori.

Ovviamente, con l'aggiornamento delle librerie sopracitate, è stato effettuato anche l'aggiornamento dell'ambiente, che è installabile servendosi dell'apposito file `environment.yaml` presente nella repository.

Questi aggiornamenti hanno portato significativi benefici, tra cui:

- **Miglioramento Prestazionale e Stabilità:** Le classificazioni risultano ora del 20% più veloci, contribuendo a una maggiore efficienza operativa.
- **Nuove Funzionalità Introdotte:**
 - Possibilità di scaricare modelli quantistici addestrati, apportando un importante valore aggiunto.
 - Supporto per la misurazione del rumore e la codifica degli errori, migliorando la robustezza complessiva del sistema.
 - Introduzione di nuovi framework di simulazione e codifica di circuiti quantistici pronti all'uso, facilitando lo sviluppo.

Una nota importante riguarda l'eliminazione del modulo `QSVM`, diventato deprecato dopo l'aggiornamento. Tutte le sue referenze sono state rimosse dalle pagine HTML correlate.

A livello pratico, per rendere i modelli scaricabili, abbiamo implementato la funzionalità di download nel metodo `classify` della classe `ClassifyControl`. Per i modelli quantistici, è stato introdotto il nuovo metodo `save`, mentre per quelli classici abbiamo utilizzato il metodo `.dump` della libreria `pickle`.

Actual Impact Set – 4 Componenti

- `routes` - smista
- `templates` - formDataset
- `ClassifyControl` – `classify`
- `myQSVM` - `classify`

False Positive Impact Set – 7 Componenti

- myQSVR – classify
- myQSVC – classify
- myPegasosQSVC - classify
- myNeuralNetworkClassifier - classify
- myNeuralNetworkRegressor – classify
- ClassifyControl – classify_control, classification_thread

Discovered Impact Set – 0 Componenti

Nessuna nuova componente scoperta.

1.3 Metriche

Dopo aver eseguito le modifiche, sono stati calcolati i parametri di Recall, Precision al fine di valutare la qualità dell'analisi d'impatto condotta, i quali risultano essere:

- **Recall** = $(\text{CIS} \cap \text{AIS}) \div \text{AIS} = 4 \div 4 = 1$
- **Precision** = $(\text{CIS} \cap \text{AIS}) \div \text{CIS} = 4 \div 11 = 0.4$

In questa Change Request è stata riscontrata una diminuzione della precisione dovuta all'idea iniziale del salvataggio del modello, pianificato per essere implementato all'interno della classe corrispondente ad ogni modello. Al fine di garantire maggiore coesione, leggibilità e semplicità nel codice, durante la fase di implementazione, è stata adottata la decisione di realizzare la funzionalità direttamente all'interno della classe *ClassifyControl*, come precedentemente specificato. Questa scelta ha evitato la necessità di apportare modifiche alle classi associate ai singoli modelli, contribuendo così a mantenere la struttura del codice più chiara e facilmente gestibile, ed in generale a ridurre l'impatto della modifica.

2 Change Request 2 (CR_2)

La presente change request si propone di introdurre una funzionalità cruciale per migliorare l'esperienza degli utenti sulla piattaforma, ovvero la possibilità di scaricare i modelli precedentemente addestrati direttamente dall'area personale dell'utente, insieme a tutti i file relativi all'esperimento associato.

Situazione Attuale

Attualmente, gli utenti possono solo visualizzare le metriche risultanti dalla fase di prediction dei modelli precedentemente allenati tramite l'area personale. Questo limita fortemente la possibilità di effettuare ulteriori predizioni e test utilizzando gli stessi modelli addestrati con gli stessi dati, addestramenti che in alcuni casi possono essere molto lunghi. Inoltre, il processo di training dei modelli quantistici si rivela essere particolarmente time-consuming, rappresentando un ostacolo significativo al riuso degli stessi e all'efficienza della piattaforma.

Proposta di Modifica

La soluzione proposta per migliorare l'esperienza utente è di implementare un meccanismo che consenta il download di tutti i file associati a un determinato esperimento, incluso il file del modello addestrato. Questa funzionalità sarà integrata attraverso la sezione "Statistics and history of operations", visualizzabile nell'area riservata degli utenti. L'implementazione di questa funzionalità non solo agevolerà l'accesso ai dati e alle informazioni rilevanti, ma contribuirà anche in modo significativo ad aumentare la flessibilità complessiva della piattaforma e a renderla molto utile.

Benefici Attesi

L'implementazione di questa modifica comporterebbe diversi vantaggi:

- **Migliore fruibilità dei modelli:** Gli utenti potrebbero scaricare i modelli per condurre ulteriori esperimenti, aumentando il valore complessivo della piattaforma;

- **Riduzione dei tempi di addestramento:** Eliminando la necessità di addestrare i modelli ad ogni esperimento, si ridurrebbe notevolmente il tempo necessario per condurre nuovi test e analisi;
- **Accesso a dati e grafici:** Tutti i dati relativi ai vari step di processing e i grafici risultanti dagli esperimenti sarebbero facilmente accessibili e scaricabili successivamente, consentendo agli utenti di conservare e utilizzare i risultati in qualsiasi momento.

Questa modifica non solo migliorerebbe l'utilità pratica della piattaforma, ma potrebbe anche incentivare una maggiore partecipazione degli utenti, contribuendo a consolidare la piattaforma stessa come un ambiente flessibile ed efficiente per la gestione dei modelli addestrati.

2.1 Analisi d'impatto

Starting Impact Set – 2 Componenti

Sono state individuate le seguenti classi e metodi:

- UtenteControl – download
- utils – encryption
- templates - compareExperiments

	download	utils	compareExperiments
download	/	1	
utils		/	
compareExperiments	1	1	/

Matrice di raggiungibilità

Per individuare le componenti coinvolte e stabilire il Candidate Impact Set (CIS) è stata generata la matrice di raggiungibilità considerando l'approccio distance based (fino al livello 3 di propagazione) per ridurre le occorrenze di falsi positivi.

Poiché la funzione da introdurre effettua operazioni su file statici, che una volta generati non vengono modificati o utilizzati da qualsiasi altra componente, non dovremmo avere altre componenti coinvolte, avendo dunque il seguente candidate impact set.

Candidate Impact Set – 3 Componenti

Sono state individuate le seguenti classi e metodi:

- UtenteControl – download
- utils – encryption
- templates – compareExperiments

2.2 Implementazione modifiche

L'integrazione della funzionalità di download è stata mirata a garantire un'esperienza utente fluida e una gestione efficiente dei file associati agli esperimenti.

Per permettere agli utenti di scaricare i modelli addestrati, è stato inserito un bottone di download per ciascun esperimento all'interno della lista presente nell'area utente. Questo bottone costituisce il punto di partenza per avviare il processo di download, la cui gestione è stata affidata ad un nuovo metodo denominato **experimentDownload** all'interno del controller dell'utente **UtenteControl**, il quale gestisce la richiesta di download proveniente dal bottone associato a un esperimento specifico.

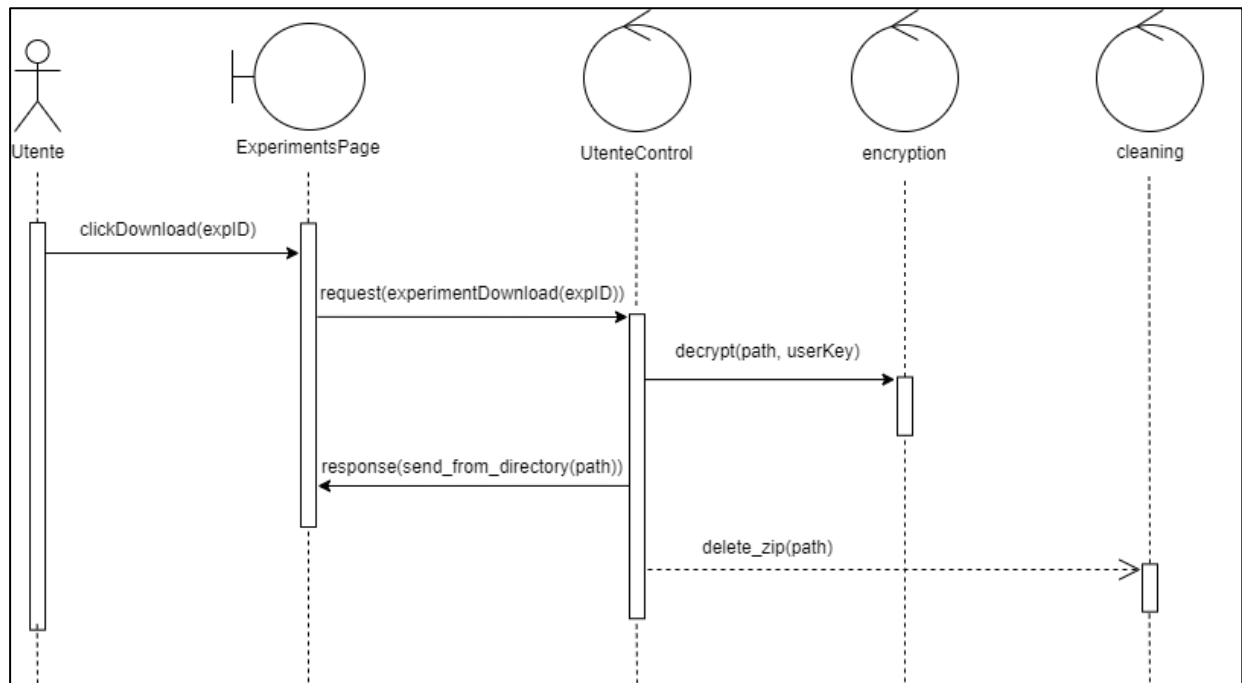
Alla pressione del bottone di download, l'ID dell'esperimento corrispondente viene acquisito e utilizzato per ottenere il riferimento ai file associati a quell'esperimento. Questi file vengono decifrati, considerando la funzionalità di cifratura/decifratura introdotta nella CR3.

Successivamente, un file zip è creato e tutti i file decifrati dell'esperimento vengono inseriti al suo interno. Tramite la response, il file zip viene scaricato direttamente dal browser dell'utente, garantendo un processo di download efficiente.

Per evitare l'accumulo di file indesiderati, è stata implementata una funzione denominata **delete_zip** all'interno del modulo **cleaning** (è stato preferito introdurre un nuovo modulo con all'interno tutti i metodi deputati alla pulizia piuttosto che unirli ai metodi di generale utilità presenti in *utils*). Questa funzione, avviata all'interno di un thread separato, rimuove il file zip una volta completato il download. L'uso di un thread separato evita conflitti nel caso in cui il file .zip sia ancora aperto e utilizzato dal metodo **experimentDownload**.

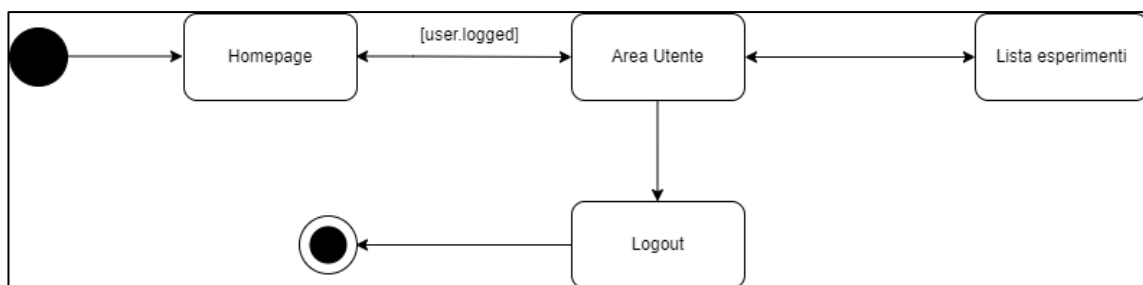
Questa implementazione garantisce un processo di download sicuro, efficiente e pulito, migliorando complessivamente la gestione dei modelli addestrati nella piattaforma.

Tramite il seguente Sequence Diagram possiamo comprendere come la modifica è stata realizzata e l'interazione tra le componenti interessate:



Sequence Diagram - Experiment Download

È possibile, inoltre, comprendere tramite il seguente navigation path l'interazione tra le pagine coinvolte nella funzionalità in questione:



Navigation Path - Experiment Download

Actual Impact Set – 3 Componenti

- cleaning– delete_zip
- templates – compareExperiments
- UtenteControl – experimentDownload

False Positive Impact Set – 2 Componenti

- utils – encryption
- UtenteControl - download

Discovered Impact Set – 2 Componenti

- UtenteControl – experimentDownload
- cleaning – delete_zip

2.3 Metriche

Dopo aver eseguito le modifiche, sono stati calcolati i parametri di Recall, Precision al fine di valutare la qualità dell'analisi d'impatto condotta, i quali risultano essere:

- **Recall** = $(\text{CIS} \cap \text{AIS}) \div \text{AIS} = 2 \div 3 = 0.7$
- **Precision** = $(\text{CIS} \cap \text{AIS}) \div \text{CIS} = 2 \div 3 = 0.7$

3 Change Request 3 (CR_3)

La presente Change Request mira a modificare la logica di memorizzazione dei file generati dagli esperimenti degli utenti, con l'obiettivo di migliorare la sicurezza e la privacy.

In particolare, si propone di spostare la cartella contenente tali file in una posizione facilmente rintracciabile sulla macchina host, implementando contemporaneamente una cifratura simmetrica rispetto ai file generati dagli esperimenti e memorizzati, per garantire la sicurezza e la riservatezza dei dati.

Situazione Attuale

Attualmente, i file vengono memorizzati in chiaro all'interno della root del progetto, nella cartella *upload_dataset*. Questa cartella contiene diverse sottocartelle corrispondenti agli utenti registrati, identificate dalla loro email, in ognuna di esse sono presenti ulteriori sottocartelle per ciascun esperimento, identificate dall'ID dell'esperimento.

Proposta di Modifica

La proposta di modifica prevede il trasferimento della cartella principale *upload_dataset* dalla root del progetto alla cartella **QMLdata** situata nella cartella dell'utente che hosta il sistema. Inoltre, si introdurrà un meccanismo di cifratura basato sull'algoritmo AES, implementato nella libreria **Fernet**, per cifrare e decifrare i file generati dagli esperimenti. Questa modifica mira a migliorare la sicurezza complessiva del sistema e a proteggere la privacy dell'utente.

Benefici

L'implementazione di questa modifica potrebbe apportare diversi benefici:

- **Aumento del grado di sicurezza e privacy:** La cifratura dei file generati dagli esperimenti contribuirà a garantire un livello superiore di sicurezza e a preservare la privacy dell'utente.
- **Scalabilità e isolamento dei dati personali:** Il trasferimento della cartella principale alla directory dell'utente favorirà la scalabilità e isolerà in modo più efficace i dati degli utenti dai dati del progetto, migliorando la gestione complessiva dei dati.

3.1 Analisi d'impatto

Starting Impact Set – 3 Componenti

Sono state individuate le seguenti classi e metodi:

- routes - smista
- utils - encryption
- __init__

	routes	encryption	__init__	models
routes	/	1		1
encryption		/		
__init__	1		/	1
models				/

Matrice di raggiungibilità

Per individuare le componenti coinvolte e stabilire il Candidate Impact Set (CIS) è stata generata la matrice di raggiungibilità considerando l'approccio distance based (fino al livello 3 di propagazione) per ridurre le occorrenze di falsi positivi.

Per l'aggiunta della funzionalità di crittografia sarà necessario fare uso di una chiave simmetrica, che dovrà necessariamente essere memorizzata sul database, ragion per cui sarà necessaria anche una modifica a *models*, che rappresenta il nostro DDL.

Candidate Impact Set – 4 Componenti

Sono state individuate le seguenti classi e metodi:

- routes - smista
- utils - encryption
- __init__
- model - models

3.2 Implementazione modifiche

Al fine di garantire una gestione sicura dei file generati durante gli esperimenti degli utenti, sono state implementate significative modifiche durante la fase di inizializzazione della piattaforma. In particolare, è stata introdotta la creazione della cartella **QMLdata** nella root della cartella dell'utente host, assumendo la stessa struttura e logica di salvataggio precedentemente presente in *upload_dataset*, che è stata ora rimossa.

Successivamente, è stato sviluppato il modulo *encryption* all'interno del package *utils* (è stato preferito creare un nuovo modulo dedicato alle operazioni di crittografia piuttosto che unirli ai metodi di generale utilità presenti in *utils*, come definito precedentemente nel SIS e CIS). Questo modulo ospita due metodi chiave, **encrypt** e **decrypt**. Il metodo *encrypt* è responsabile della cifratura dei file all'interno di un percorso specifico, utilizzando una chiave predefinita. Analogamente, il metodo *decrypt* è dedicato alla decifratura dei file, garantendo una gestione sicura dei dati all'interno del sistema.

Il processo di cifratura viene attivato al termine di ciascun esperimento nel metodo *smista* della classe *routes*, in particolare dopo la fase di preprocessing se non è stata selezionata alcuna opzione di classificazione/regressione, o alla fine delle operazioni di classificazione/regressione in caso contrario. Inoltre, per quanto riguarda la cifratura, essa prevede sempre la cancellazione dei file in chiaro, per una questione di sicurezza, mentre la decifratura prevede la creazione dei corrispondenti file in chiaro senza cancellare i cifrati già presenti, per evitare di dover ripetere un altro passo di cifratura. Subito dopo la cifratura, viene eseguita la pulizia della directory dell'utente eliminando opportunamente i file non cifrati mediante l'utilizzo del metodo **delete_unencrypted** (tramite thread, per evitare una situazione di interleaving che si verificava sporadicamente) presente nel modulo *cleaning*. È importante sottolineare che la chiave di cifratura, elemento cruciale per garantire la sicurezza del processo, viene gestita in maniera persistente sul database, mediante associazione univoca ad ogni utente al momento della registrazione dello stesso; dunque, è stato anche aggiunto il relativo attributo nella tabella User in *models*.

Actual Impact Set – 5 componenti

- routes - smista
- __init__
- model - models
- cleaning – delete_unencrypted
- encryption – encrypt, decrypt

False Positive Impact Set – 1 componente

- utils - encryption

Discovered Impact Set – 3 componenti

- cleaning _ delete_unencrypted
- encryption – encrypt, decrypt

3.3 Metriche

Dopo aver eseguito le modifiche, sono stati calcolati i parametri di Recall e Precision al fine di valutare la qualità dell'analisi d'impatto condotta, i quali risultano essere:

Recall = $(\text{CIS} \cap \text{AIS}) \div \text{AIS} = 4 \div 5 = 0.8$

Precision = $(\text{CIS} \cap \text{AIS}) \div \text{CIS} = 4 \div 4 = 1$

4 Change Request 4 (CR_4)

La presente Change Request è focalizzata sull'aggiunta di una nuova funzionalità al software, mirata a consentire agli utenti di recuperare la propria password durante la fase di login. Attualmente, nel caso in cui un utente dimentichi la propria password, non esiste alcuna procedura automatizzata per il recupero, obbligandolo a contattare gli amministratori del sistema.

Situazione attuale

Nella configurazione attuale del software, la perdita della password implica l'unico ricorso al contatto diretto con gli amministratori di sistema per ottenere assistenza e il ripristino delle credenziali di accesso.

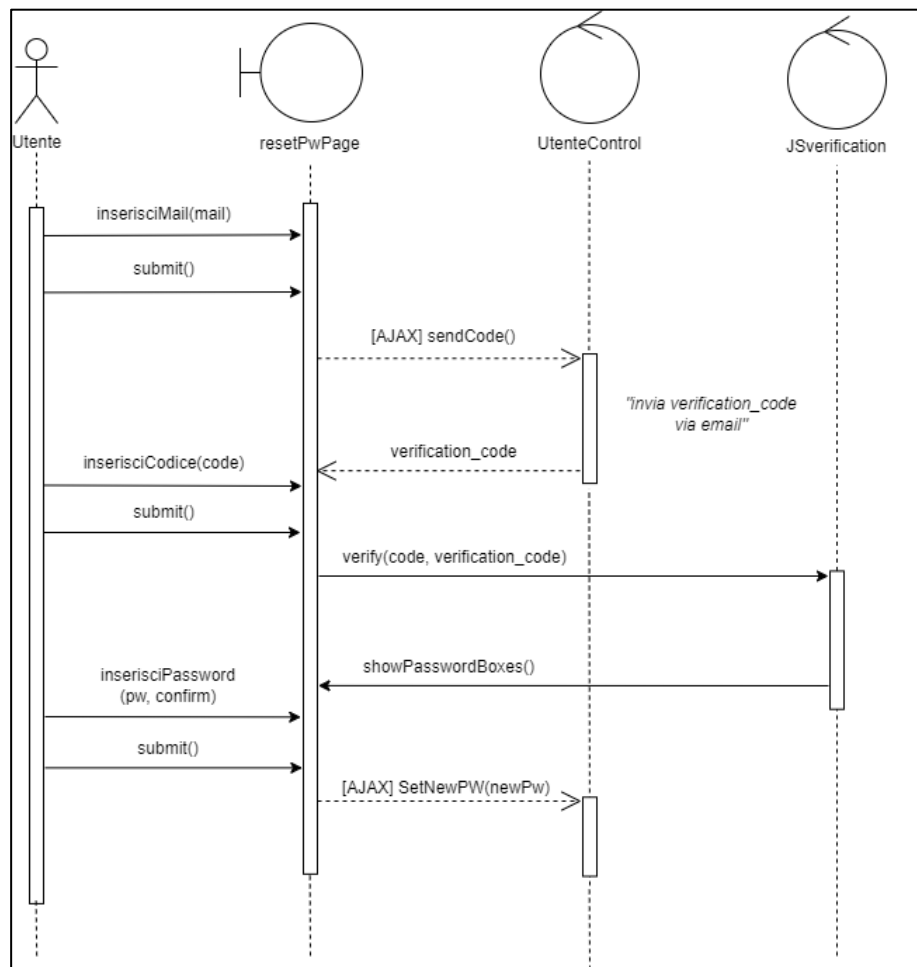
Proposta di modifica

La modifica proposta consiste nell'implementare una nuova funzionalità di recupero password. Questa funzionalità consentirà agli utenti registrati di recuperare la propria password attraverso un processo automatizzato. In particolare, fornirà agli utenti la possibilità di inserire l'indirizzo email associato al proprio account. Successivamente, il sistema invierà un codice di verifica all'indirizzo email specificato. Dopo aver inserito e verificato correttamente il codice, l'utente sarà autorizzato a definire una nuova password per il proprio account. In seguito, sarà mostrato in diagramma di sequenza per la funzionalità specificata.

Benefici

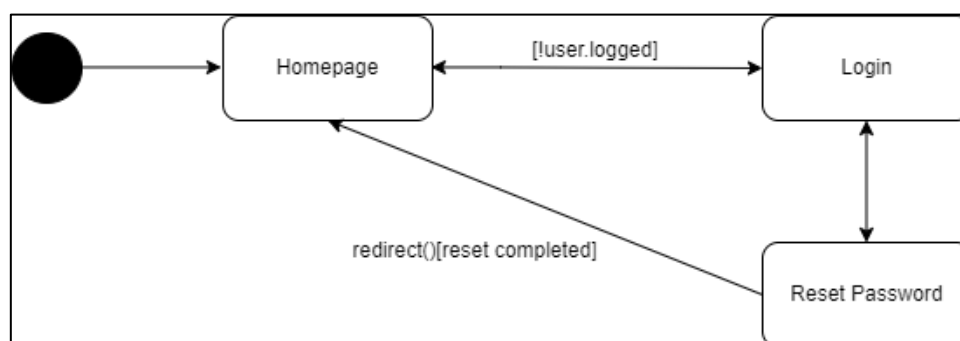
L'implementazione di questa modifica offre diversi vantaggi, tra cui:

- **Miglioramento dell'usabilità:** La nuova funzionalità semplifica e accelera il processo di recupero password, migliorando l'esperienza complessiva dell'utente durante la gestione delle credenziali di accesso;
- **Aumento della sicurezza:** L'implementazione di un sistema automatizzato per il recupero password può essere progettata per garantire la sicurezza del processo, ad esempio attraverso l'uso di codici di verifica a tempo.



Sequence Diagram - Reset Password

Inoltre, tramite il seguente navigation path, si può osservare l'interazione tra le pagine coinvolte nella funzionalità in questione:



Navigation Path - Reset Password

4.1 Analisi d'impatto

Starting Impact Set – 3 Componenti

Sono state individuate le seguenti classi e metodi:

- Templates – login, resetPwPage
- UtenteControl – setNewPW

	login	resetPwPage	setNewPW
login	/	1	2
resetPwPage		/	1
setNewPW			/

Matrice di raggiungibilità

Per individuare le componenti coinvolte e stabilire il Candidate Impact Set (CIS) è stata generata la matrice di raggiungibilità considerando l'approccio distance based (fino al livello 3 di propagazione) per ridurre le occorrenze di falsi positivi.

L'approccio che vogliamo adottare per il recupero password è di creare una nuova pagina, accessibile dalla pagina di login tramite il link "Forgot your password", e da quest'ultima iniziare il processo di recupero che parte facendo inserire all'utente l'email associata all'account da recuperare. Una volta inserita la mail e verificato che questa sia effettivamente presente nel database, viene inviata una mail con un codice di verifica randomico di 6 cifre. L'utente inserisce il codice e, se corretto, vengono visualizzati i campi "Nuova Password" e "Conferma Nuova Password"; se questi ultimi rispettano le condizioni di sicurezza il processo di recupero termina, la password viene modificata con successo e l'utente viene reindirizzato alla Home.

Considerando dunque le modifiche da effettuare, non dovremmo avere altre componenti interessate, avendo dunque un CIS come segue.

Candidate Impact Set – 3 Componenti

Sono state individuate le seguenti classi e metodi:

- UtenteControl – setNewPW

- Templates – login, resetPwPage

4.2 Implementazione modifiche

Per implementare questa nuova funzionalità, abbiamo introdotto una sezione dedicata denominata "Forgot your password" nella pagina di login. Al momento della selezione di questa sezione, viene aperta la pagina *resetPwPage*, dove l'utente è tenuto a inserire l'indirizzo email associato al proprio account per avviare il processo di recupero della password. Una volta inserita l'email, la richiesta viene inviata attraverso un'apposita chiamata AJAX al metodo denominato *sendCode* premendo il pulsante "Send".

Successivamente, nel caso in cui l'utente sia registrato nel nostro database, riceverà un codice di recupero casuale composto da 6 cifre all'indirizzo email fornito. Questo codice dovrà essere inserito in un nuovo campo di testo che apparirà nella stessa pagina di reset. La verifica del codice avviene quando l'utente preme il pulsante "Verify". Se il codice inserito risulta corretto, nella stessa pagina verranno visualizzati i campi necessari per definire la nuova password. Inoltre, al momento della conferma della nuova password, viene inviata un'altra richiesta AJAX per modificare la password nel database.

È importante sottolineare che, contrariamente a quanto previsto nel CIS, abbiamo implementato due metodi distinti, ciascuno con la propria route, all'interno del controller *UtenteControl* per gestire efficacemente l'operazione di recupero della password. Questa scelta è stata necessaria per garantire un'implementazione fluida e sicura della funzionalità, tenendo conto delle specifiche esigenze e delle eventuali complessità derivanti dalla gestione del recupero password.

Actual Impact Set – 4 Componenti

- UtenteControl – setNewPW, sendCode
- Templates – login, resetPwPage

False Positive Impact Set – 0 Componenti

Nessun falso positivo riscontrato.

Discovered Impact Set – 1 Componente

- UtenteControl – sendCode

4.3 Metriche

Dopo aver eseguito le modifiche, sono stati calcolati i parametri di Recall, Precision al fine di valutare la qualità dell'analisi d'impatto condotta, i quali risultano essere:

- **Recall** = $(\text{CIS} \cap \text{AIS}) \div \text{AIS} = 3 \div 4 = 0.8$
- **Precision** = $(\text{CIS} \cap \text{AIS}) \div \text{CIS} = 3 \div 3 = 1$