



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Ingegneria, Gestione ed Evoluzione del Software

Pre-Modification Test Case Specification

TEAM MEMBER

Donia Daniele - 0522501575

La Marca Antonio - 0522501557

Somma Pasquale - 0522501543

1	Introduzione	1
1.1	Approccio	1
2	Categorie e scelte	2
2.1	Test Case Specification di unità	2
2.1.1	Repo Mining	2
2.1.2	Text Mining	20
2.1.3	Rifinitura dati analisi statica	75
2.1.4	Union	92
2.2	Test Case Specification di integrazione	142
2.2.1	Repo Mining	142
2.2.2	Text Mining	150
2.2.3	Rifinitura dati analisi statica	162
2.2.4	Union	168

1.1 Approccio

Il seguente documento descrive la specifica dettagliata dei test case sviluppati prima dell'implementazione delle change request, con l'obiettivo di validare il sistema nelle sue funzionalità iniziali. I test case sono organizzati in categorie fondamentali e coprono diversi livelli di test, inclusi test di unità ed integrazione. Ogni test case fornisce una descrizione degli input, degli output attesi e delle condizioni per il corretto funzionamento del sistema, garantendo una solida base di verifica per confrontare l'efficacia e la stabilità del sistema prima delle modifiche successive.

 Categorie e scelte

Di seguito sono riportate le categorie e le scelte individuate per gli oggetti dell'ambiente citati nella sezione precedente.

2.1 Test Case Specification di unità

2.1.1 Repo Mining

Divide Dataset

Test case ID: TC_1.1_1	Test Frame: EI2
Input	
Il file initial_dataset.csv e la directory Dataset_Divided non esistono.	
Output	
FileNotFoundError	
Test Driver	
Classe: TestDivideDataset	Metodo: test_case_1
Oracolo	
FileNotFoundError	

Tabella 2.1: Metodo dello script `test_divide_dataset.py`

Test case ID: TC_1.1_2	Test Frame: EI1 N1 ED1 S1
Input	
Il file <code>initial_dataset.csv</code> contiene solo l'header, senza righe di dati.	
Output	
La directory <code>Dataset_Divided</code> contiene il file <code>1.csv</code> con solo l'header.	
Test Driver	
Classe: <code>TestDivideDataset</code>	Metodo: <code>test_case_2</code>
Oracolo	
La directory <code>Dataset_Divided</code> contiene il file <code>1.csv</code> con solo l'header.	

Tabella 2.2: Metodo dello script `test_divide_dataset.py`

Test case ID: TC_1.1_3	Test Frame: S2
Input	
Il file <code>initial_dataset.csv</code> esiste e contiene 50 record. La directory <code>Dataset_Divided</code> esiste, ma non vi sono i permessi di scrittura	
Output	
PermissionError	
Test Driver	
Classe: <code>TestDivideDataset</code>	Metodo: <code>test_case_3</code>
Oracolo	
PermissionError	

Tabella 2.3: Metodo dello script `test_divide_dataset.py`

Test case ID: TC_1.1_4	Test Frame: EI1 N2 ED1 S1
Input	
Il file <code>initial_dataset.csv</code> esiste e contiene 50 record. La directory <code>Dataset_Divided</code> esiste.	
Output	
Dataset_Divided con <code>1.csv</code> con 49 record del dataset e <code>2.csv</code> con un record e sprovvisto di header	
Test Driver	

Classe: TestDivideDataset	Metodo: test_case_4
Oracolo	
Dataset_Divided con un singolo file CSV contenente i 50 record del dataset	

Tabella 2.4: Metodo dello script test_divide_dataset.py

Test case ID: TC_1.1_5	Test Frame: EI1 N3 ED1 S1
Input	
Il file initial_dataset.csv contiene 60 record. La directory Dataset_Divided esiste	
Output	
Dataset_Divided contiene i file 1.csv e 2.csv. 1.csv contiene l'header e 49 record, 2.csv è sprovvisto di header e contiene 11 record	
Test Driver	
Classe: TestDivideDataset	Metodo: test_case_5
Oracolo	
Dataset_Divided contiene 1.csv e 2.csv, entrambi con l'header. 1.csv contiene 50 record, 2.csv e contiene 10 record	

Tabella 2.5: Metodo dello script test_divide_dataset.py

Test case ID: TC_1.1_6	Test Frame: EI1 N3 ED2 S1
Input	
Il file initial_dataset.csv contiene 60 record. La directory Dataset_Divided non esiste	
Output	
È creata la directory Dataset_Divided contenente file 1.csv e 2.csv. 1.csv contiene l'header e 49 record, 2.csv è sprovvisto di header e contiene 11 record	
Test Driver	
Classe: TestDivideDataset	Metodo: test_case_6
Oracolo	

Dataset_Divided è creata e contiene 1.csv e 2.csv, entrambi con l'header.

1.csv contiene 50 record,

2.csv e contiene 10 record

Tabella 2.6: Metodo dello script test_divide_dataset.py

Initialize

Test case ID: TC_1.2_1	Test Frame: FM2 ED1 EM1 FD1 ND3 ER1
Input	
<p>Il parametro <code>miniDatasetName</code> ha un formato invalido.</p> <p>Il dataset da minare esiste e ha un formato valido.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p>	
Output	
OSError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_1
Oracolo	
OSError	

Tabella 2.7: Metodo dello script test_initialize.py

Test case ID: TC_1.2_2	Test Frame: FM1 ED2 EM1 ER1
Input	
<p>Il parametro <code>miniDatasetName</code> ha un formato valido.</p> <p>Il dataset da minare non esiste.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p>	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_2
Oracolo	

FileNotFoundError

Tabella 2.8: Metodo dello script test_initialize.py

Test case ID: TC_1.2_3	Test Frame: FM1 ED1 EM2
Input	
Il parametro <code>miniDatasetName</code> ha un formato valido. Il dataset da minare esiste ed è valido. La directory <code>mining_results</code> non esiste.	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_3
Oracolo	
FileNotFoundError	

Tabella 2.9: Metodo dello script test_initialize.py

Test case ID: TC_1.2_4	Test Frame: FM1 ED1 EM1 FD2 ND3 ER1
Input	
Il parametro <code>miniDatasetName</code> ha un formato valido. Il dataset da minare esiste ma ha un formato invalido. La directory <code>mining_results</code> esiste. La directory relativa al mining della specifica repo esiste.	
Output	
ValueError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_4
Oracolo	
ValueError	

Tabella 2.10: Metodo dello script test_initialize.py

Test case ID: TC_1.2_5	Test Frame: FM1 ED1 FD1 ND3 EM1 ER1
------------------------	-------------------------------------

Input	
<p>Il parametro <code>miniDatasetName</code> ha un formato valido.</p> <p>Il dataset da minare esiste, ha un formato valido e contiene 3 record.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p>	
Output	
Il dizionario data contiene i record contenuti nel dataset	
Test Driver	
Classe: <code>TestInitialize</code>	Metodo: <code>test_case_5</code>
Oracolo	
Il dizionario data contiene i record contenuti nel dataset	

Tabella 2.11: Metodo dello script `test_initialize.py`

Test case ID: TC_1.2_6	Test Frame: FM1 ED1 EM1 FD1 ND1 ER2
Input	
<p>Il parametro <code>miniDatasetName</code> ha un formato valido.</p> <p>Il dataset da minare esiste, ha un formato valido e non contiene record.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p>	
Output	
Il dizionario data è vuoto	
Test Driver	
Classe: <code>TestInitialize</code>	Metodo: <code>test_case_6</code>
Oracolo	
Il dizionario data è vuoto e la repository è creata.	

Tabella 2.12: Metodo dello script `test_initialize.py`

Test case ID: TC_1.2_7	Test Frame: FM1 ED1 EM1 FD1 ND1 ER1
Input	

<p>Il parametro <code>miniDatasetName</code> ha un formato valido.</p> <p>Il dataset da minare esiste, ha un formato valido e non contiene record.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p>	
Output	
Il dizionario <code>data</code> è vuoto	
Test Driver	
Classe: <code>TestInitialize</code>	Metodo: <code>test_case_7</code>
Oracolo	
Il dizionario <code>data</code> è vuoto	

Tabella 2.13: Metodo dello script `test_initialize.py`

Test case ID: TC_1.2_8	Test Frame: FM1 ED1 FD1 ND2 EM1 ER1
Input	
<p>Il parametro <code>miniDatasetName</code> ha un formato valido.</p> <p>Il dataset da minare esiste, ha un formato valido e contiene un record.</p> <p>La directory <code>mining_results</code> esiste.</p> <p>La directory relativa al mining della specifica repo esiste.</p>	
Output	
Il dizionario <code>data</code> contiene i commit specificati nel dataset.	
Test Driver	
Classe: <code>TestInitialize</code>	Metodo: <code>test_case_5</code>
Oracolo	
Il dizionario <code>data</code> contiene i commit specificati nel dataset.	

Tabella 2.14: Metodo dello script `test_initialize.py`**Start Mining Repo**

Test case ID: TC_1.3_1	Test Frame: FD1 N1 FW1 EW1 FR1 ER1 ERI1 ERC1 EC1 SC1 EE1 SE1
Input	

<p>Il parametro <code>data</code> ha un formato corretto e non contiene alcun record.</p> <p>Il parametro <code>cwd</code> ha un formato valido e indica un percorso esistente.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
<code>CHECK.txt</code> è vuoto.	
Test Driver	
Classe: <code>TestStartMiningRepo</code>	Metodo: <code>test_case_1</code>
Oracolo	
<code>CHECK.txt</code> è vuoto.	

Tabella 2.15: Metodo dello script `test_start_mining_repo.py`

Test case ID: TC_1.3_2	Test Frame: FD2
Input	
<p>Il parametro <code>data</code> ha un formato non valido.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
KeyError	
Test Driver	
Classe: <code>TestStartMiningRepo</code>	Metodo: <code>test_case_2</code>
Oracolo	
KeyError	

Tabella 2.16: Metodo dello script `test_start_mining_repo.py`

Test case ID: TC_1.3_3	Test Frame: FD1 N3 CL3
Input	

<p>Il parametro <code>data</code> ha un formato valido, ha 2 record, ma contiene link non valido.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
MissingSchema	
Test Driver	
Classe: TestStartMiningRepo	Metodo: test_case_3
Oracolo	
MissingSchema	

Tabella 2.17: Metodo dello script test_start_mining_repo.py

Test case ID: TC_1.3_4	Test Frame: FD1 N3 CL2
Input	
<p>Il parametro <code>data</code> ha un formato valido, ha 2 record e contiene link valido ma non esistente.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
ConnectionError	
Test Driver	
Classe: TestStartMiningRepo	Metodo: test_case_4
Oracolo	
ConnectionError	

Tabella 2.18: Metodo dello script test_start_mining_repo.py

Test case ID: TC_1.3_5	Test Frame: FD1 N3 CL1 CR2 FW1 EW1 FR1 ER1 ERI1 ERC1 EC1 SC1 EE1 SE1
Input	

<p>Il parametro <code>data</code> ha un formato valido ma contiene link a una repository non valida.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
<code>CHECK.txt</code> indica che la repository non è disponibile.	
Test Driver	
Classe: <code>TestStartMiningRepo</code>	Metodo: <code>test_case_5</code>
Oracolo	
<code>CHECK.txt</code> indica che la repository non è disponibile.	

Tabella 2.19: Metodo dello script `test_start_mining_repo.py`

Test case ID: <code>TC_1.3_6</code>	Test Frame: <code>FD1 N3 CL1 CR1 CC2 FW1 EW1 FR1 ER1 ERI1 ERC1 EC1 SC1 EE1 SE1</code>
Input	
<p>Il parametro <code>data</code> ha un formato valido ma contiene link a un commit CON path eccedente la lunghezza massima.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
<code>GitCommandError</code>	
Test Driver	
Classe: <code>TestStartMiningRepo</code>	Metodo: <code>test_case_7</code>
Oracolo	
<code>GitCommandError</code>	

Tabella 2.20: Metodo dello script `test_start_mining_repo.py`

Test case ID: TC_1.3_7	Test Frame: FD1 N3 CL1 CR1 CC3 FW1 EW1 FR1 ER1 ERI1 ERC1 EC1 SC1 EE1 SE1
Input	
<p>Il parametro <code>data</code> ha un formato valido ma contiene link a un commit non esistente.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
<code>CHECK.txt</code> indica che il commit non è esistente.	
Test Driver	
Classe: <code>TestStartMiningRepo</code>	Metodo: <code>test_case_7</code>
Oracolo	
<code>CHECK.txt</code> indica che il commit non è esistente.	

Tabella 2.21: Metodo dello script `test_start_mining_repo.py`

Test case ID: TC_1.3_8	Test Frame: FD1 N3 CL1 CR1 CC4 FW1 EW1 FR1 ER1 ERI1 ERC1 EC1 SC1 EE1 SE1
Input	
<p>Il parametro <code>data</code> ha un formato valido ma contiene link a due repo con commit non definiti.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
<p><code>CHECK.txt</code> e <code>ERRORS.txt</code> indicano per ciascun commit non definito: indice, url della repo e id commit; ma l'indice non è incrementato</p>	
Test Driver	
Classe: <code>TestStartMiningRepo</code>	Metodo: <code>test_case_8</code>
Oracolo	
<p><code>CHECK.txt</code> e <code>ERRORS.txt</code> indicano per ciascun commit non definito: indice, url della repo e id commit;</p>	

Tabella 2.22: Metodo dello script test_start_mining_repo.py

Test case ID: TC_1.3_9	Test Frame: FD1 N3 CL1 CR1 CC1 CM1 FW1 EW1 FR1 ER1 ERI1 ERC1 EC1 SC1 EE1 SE1
Input	
<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, senza modifiche.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
<code>CHECK.txt</code> indica status OK.	
Test Driver	
Classe: TestStartMiningRepo	Metodo: test_case_9
Oracolo	
<code>CHECK.txt</code> indica status OK.	

Tabella 2.23: Metodo dello script test_start_mining_repo.py

Test case ID: TC_1.3_10	Test Frame: FD1 N1 CL1 CR1 CC1 CM2 FW1 EW1 FR1 ER1 ERI1 ERC1 EC1 SC1 EE1 SE1
Input	
<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, con modifiche non relative a file Java.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
<code>CHECK.txt</code> indica status OK.	
Test Driver	

Classe: TestStartMiningRepo	Metodo: test_case_10
Oracolo	
CHECK.txt indica status OK.	

Tabella 2.24: Metodo dello script test_start_mining_repo.py

Test case ID: TC_1.3_11	Test Frame: FD1 N3 CL1 CR1 CC1 CM3 FW1 EW1 FR1 ER1 ERI2 ERC2 EC1 SC1 EE2 SE1
Input	
<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, con introduzione di file Java.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit non esistono.</p>	
Output	
CHECK.txt indica status OK e la directory con l'ID del commit è creata e vuota.	
Test Driver	
Classe: TestStartMiningRepo	Metodo: test_case_11
Oracolo	
CHECK.txt indica status OK e la directory con l'ID del commit è creata e vuota.	

Tabella 2.25: Metodo dello script test_start_mining_repo.py

Test case ID: TC_1.3_12	Test Frame: FD1 N3 CL1 CR1 CC1 CM4 FW1 EW1 FR1 ER1 ERI1 ERC1 EC1 SC1 EE1 SE1
Input	
<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, con modifiche a file Java.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p>CHECK.txt e ERRORS.txt esistono</p>	
Output	

CHECK.txt indica status OK e la directory con l'ID del commit ha file modificati.	
Test Driver	
Classe: TestStartMiningRepo	Metodo: test_case_12
Oracolo	
CHECK.txt indica status OK e la directory con l'ID del commit ha file modificati.	

Tabella 2.26: Metodo dello script test_start_mining_repo.py

Test case ID: TC_1.3_13	Test Frame: FD1 N3 CL1 CR1 CC1 CM4 FW1 EW1 FR1 ER1 ERI1 ERC1 EC2 SC1 EE2 SE1
Input	
<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, con modifiche a file Java.</p> <p>Il parametro <code>cwd</code> ha un formato valido ed esiste.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p>CHECK.txt e ERRORS.txt non esistono</p>	
Output	
CHECK.txt e ERRORS.txt sono creati. CHECK.txt indica status OK e la directory con l'ID del commit ha file modificati.	
Test Driver	
Classe: TestStartMiningRepo	Metodo: test_case_13
Oracolo	
CHECK.txt e ERRORS.txt sono creati. CHECK.txt indica status OK e la directory con l'ID del commit ha file modificati.	

Tabella 2.27: Metodo dello script test_start_mining_repo.py

Test case ID: TC_1.3_14	Test Frame: FW3
Input	

<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, con modifiche a file Java.</p> <p>Il parametro <code>cwd</code> non è una stringa.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono</p>	
Output	
TypeError	
Test Driver	
Classe: TestStartMiningRepo	Metodo: test_case_14
Oracolo	
TypeError	

Tabella 2.28: Metodo dello script test_start_mining_repo.py

Test case ID: TC_1.3_15	Test Frame: FW2
Input	
<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, con modifiche a file Java.</p> <p>Il parametro <code>cwd</code> non ha un formato valido.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono</p>	
Output	
OSError	
Test Driver	
Classe: TestStartMiningRepo	Metodo: test_case_15
Oracolo	
OSError	

Tabella 2.29: Metodo dello script test_start_mining_repo.py

Test case ID: TC_1.3_16	Test Frame: FW1 EW2
-------------------------	---------------------

Input	
<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, con modifiche a file Java.</p> <p>Il parametro <code>cwd</code> è valido ma indica path non esistente.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono</p>	
Output	
FileNotFoundError	
Test Driver	
Classe: TestStartMiningRepo	Metodo: test_case_16
Oracolo	
FileNotFoundError	

Tabella 2.30: Metodo dello script test_start_mining_repo.py

Test case ID: TC_1.3_17	Test Frame: FR2
Input	
<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, con modifiche a file Java.</p> <p>Il parametro <code>cwd</code> è valido. Il parametro <code>repoName</code> non ha un formato valido. Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
OSError	
Test Driver	
Classe: TestStartMiningRepo	Metodo: test_case_17
Oracolo	
OSError	

Tabella 2.31: Metodo dello script test_start_mining_repo.py

Test case ID: TC_1.3_18	Test Frame:FR3
-------------------------	----------------

Input	
<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, con modifiche a file Java.</p> <p>Il parametro <code>cwd</code> è valido. Il parametro <code>repoName</code> non è una stringa. Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
TypeError	
Test Driver	
Classe: <code>TestStartMiningRepo</code>	Metodo: <code>test_case_18</code>
Oracolo	
TypeError	

Tabella 2.32: Metodo dello script `test_start_mining_repo.py`

Test case ID: TC_1.3_19	Test Frame: FR1 ER2
Input	
<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, con modifiche a file Java.</p> <p>Il parametro <code>cwd</code> è valido. Il parametro <code>repoName</code> ha un formato valido ma indica una repo non esistente.</p>	
Output	
FileNotFoundError	
Test Driver	
Classe: <code>TestStartMiningRepo</code>	Metodo: <code>test_case_19</code>
Oracolo	
FileNotFoundError	

Tabella 2.33: Metodo dello script `test_start_mining_repo.py`

Test case ID: TC_1.3_20	Test Frame: EC1 SC2
Input	

<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, con modifiche a file Java.</p> <p>Il parametro <code>cwd</code> è valido e indica path esistente.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono, ma <code>CHECK.txt</code> non è accessibile.</p>	
Output	
Si verifica un <code>PermissionError</code> .	
Test Driver	
Classe: <code>TestStartMiningRepo</code>	Metodo: <code>test_case_20</code>
Oracolo	
<code>PermissionError</code>	

Tabella 2.34: Metodo dello script `test_start_mining_repo.py`

Test case ID: <code>TC_1.3_21</code>	Test Frame: <code>EE1 SE2</code>
Input	
<p>Il parametro <code>data</code> ha un formato valido e contiene link repo con commit valido, con modifiche a file Java.</p> <p>Il parametro <code>cwd</code> è valido e indica path esistente.</p> <p>Il parametro <code>repoName</code> ha un formato valido e indica una repo esistente.</p> <p>Le directory corrispondenti all'id della repo e all'id del commit esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono, ma <code>ERRORS.txt</code> non è accessibile</p>	
Output	
<code>PermissionError</code>	
Test Driver	
Classe: <code>TestStartMiningRepo</code>	Metodo: <code>test_case_21</code>
Oracolo	
<code>PermissionError</code>	

Tabella 2.35: Metodo dello script `test_start_mining_repo.py`

2.1.2 Text Mining

Text Mining

Test case ID: TC_2.1_1	Test Frame: CN1
Input	
Parametro: tokens	[]
Output	
Restituisce una lista vuota	
Test Driver	
Classe: TestRemoveNotAlpha	Metodo: test_case_1
Oracolo	
Restituisce una lista vuota	

Tabella 2.36: Metodo dello script test_text_mining.py

Test case ID: TC_2.1_2	Test Frame: CN2
Input	
Parametro: tokens	["class", "public", "int", "String", "try", "catch", "main", "final", "while", "return"]
Output	
Restituisce una lista identica a quella di input	
Test Driver	
Classe: TestRemoveNotAlpha	Metodo: test_case_2
Oracolo	

Restituisce una lista identica a quella di input

Tabella 2.37: Metodo dello script test_text_mining.py

Test case ID: TC_2.1_3	Test Frame: CN3
Input	
Parametro: tokens	["main1", "while2", "return4", "if1else", "for3", "abstract4"]
Output	
Restituisce una lista vuota	
Test Driver	
Classe: TestRemoveNotAlpha	Metodo: test_case_3
Oracolo	
Restituisce una lista vuota	

Tabella 2.38: Metodo dello script test_text_mining.py

Test case ID: TC_2.1_4	Test Frame: CN4
Input	
Parametro: tokens	["class", "public", "int", "String", "try", "catch", "main1", "final", "while2", "return", "if1else", "for3", "abstract4"]
Output	
["class", "public", "int", "String", "try", "catch", "final", "return"]	
Test Driver	

Classe: TestRemoveNotAlpha	Metodo: test_case_4
Oracolo	
Restituisce una lista contenente le parole con soli caratteri alfabetici	

Tabella 2.39: Metodo dello script test_text_mining.py

Test case ID: TC_2.2_1	Test Frame: CN1
Input	
Parametro: s	""
Output	
Restituisce una lista vuota	
Test Driver	
Classe: TestStringTokenizer	Metodo: test_case_1
Oracolo	
Restituisce una lista vuota	

Tabella 2.40: Metodo dello script test_text_mining.py

Test case ID: TC_2.2_2	Test Frame: CN2
Input	
Parametro: s	"int main() return 0; "
Output	
["int", "main", "return"]	
Test Driver	

Classe: TestStringTokenizer	Metodo: test_case_2
Oracolo	
Restituisce una lista con tutte e sole le parole all'interno della stringa di input	

Tabella 2.41: Metodo dello script test_text_mining.py

Test case ID: TC_2.2_3	Test Frame: CN3
Input	
Parametro: s	<pre>// this is a comment\n /* This is another block comment \n that spans multiple lines */</pre>
Output	
['This', 'is', 'another', 'block', 'comment', 'that', 'spans', 'multiple', 'lines']	
Test Driver	
Classe: TestStringTokenizer	Metodo: test_case_3
Oracolo	
Restituisce una lista delle parole dei commenti multi-line	

Tabella 2.42: Metodo dello script test_text_mining.py

Test case ID: TC_2.2_4	Test Frame: CN4
Input	
Parametro: s	" \"Testiamo le costanti stringhe\""
Output	

[]	
Test Driver	
Classe: TestStringTokenizer	Metodo: test_case_4
Oracolo	
Restituisce una lista vuota	

Tabella 2.43: Metodo dello script test_text_mining.py

Test case ID: TC_2.2_5	Test Frame: CN5
Input	
Parametro: s	"string main() {// comment \n return Test passato} "
Output	
["string", "main", "return"]	
Test Driver	
Classe: TestStringTokenizer	Metodo: test_case_5
Oracolo	
Restituisce una lista con tutte e sole le parole all'interno della stringa di input, tranne dei commenti single-line	

Tabella 2.44: Metodo dello script test_text_mining.py

Test case ID: TC_2.2_6	Test Frame: CN6
Input	

Parametro: s	"string main() { // comment \n return " \\Prova " }
Output	
["string", "main", "return"]	
Test Driver	
Classe: TestStringTokenizer	Metodo: test_case_6
Oracolo	
Restituisce una lista con tutte e sole le parole all'interno della stringa di input, tranne dei commenti single-line	

Tabella 2.45: Metodo dello script test_text_mining.py

Test case ID: TC_2.3_1	Test Frame: AC2
Input	
Parametro: java_file	Non accessibile
Output	
PermissionError	
Test Driver	
Classe: TestRemoveComments	Metodo: test_case_1
Oracolo	
PermissionError	

Tabella 2.46: Metodo dello script test_text_mining.py

Test case ID: TC_2.3_2	Test Frame: AC1 CN1
Input	
Parametro: java_file	""
Output	
"	
Test Driver	
Classe: TestRemoveComments	Metodo: test_case_2
Oracolo	
Restituisce una stringa vuota	

Tabella 2.47: Metodo dello script test_text_mining.py

Test case ID: TC_2.3_3	Test Frame: AC1 CN2
Input	
Parametro: java_file	'public class Test { public static void main(String[] args) { System.out.println("Hello World"); } }'
Output	
'public class Test { public static void main(String[] args) { System.out.println("Hello World"); } }'	
Test Driver	
Classe: TestRemoveComments	Metodo: test_case_3
Oracolo	

Restituisce una stringa uguale al contenuto del file

Tabella 2.48: Metodo dello script test_text_mining.py

Test case ID: TC_2.3_4	Test Frame: AC1 CN3
Input	
Parametro: java_file	''' \n /* This is a comment */ \n // Another comment \n '''
Output	
' \n \n // Another comment \n '	
Test Driver	
Classe: TestRemoveComments	Metodo: test_case_4
Oracolo	
Restituisce una stringa contenente il codice e i commenti single-line	

Tabella 2.49: Metodo dello script test_text_mining.py

Test case ID: TC_2.3_5	Test Frame: AC1 CN4
Input	
Parametro: java_file	''' \n public class Test { \n /* This is a comment */ \n public static void main(String[] args) { \n // Single line comment\n System.out.println("Hello World"); /* Inline comment */\n } \n } \n '''
Output	

<pre>'public class Test {n {n public static void main(String[] args) {n // Single line comment{n System.out.println("Hello World"); {n {n '</pre>	
Test Driver	
Classe: TestRemoveComments	Metodo: test_case_5
Oracolo	
Restituisce una stringa con solo il codice e commenti single-line	

Tabella 2.50: Metodo dello script test_text_mining.py

Test case ID: TC_2.4_1	Test Frame: ST1
Input	
Parametro: java_file_name	File inesistente
Output	
FileNotFoundException	
Test Driver	
Classe: TestTakeJavaClass	Metodo: test_case_1
Oracolo	
FileNotFoundException	

Tabella 2.51: Metodo dello script test_text_mining.py

Test case ID: TC_2.4_2	Test Frame: ST2
Input	
Parametro: java_file_name	File esistente con permessi limitati

Output	
PermissionError	
Test Driver	
Classe: TestTakeJavaClass	Metodo: test_case_2
Oracolo	
PermissionError	

Tabella 2.52: Metodo dello script test_text_mining.py

Test case ID: TC_2.4_3	Test Frame: ST3 CN1
Input	
Parametro: java_file_name	Contenuto file: ""
Output	
{}	
Test Driver	
Classe: TestTakeJavaClass	Metodo: test_case_3
Oracolo	
Dizionario vuoto	

Tabella 2.53: Metodo dello script test_text_mining.py

Test case ID: TC_2.4_4	Test Frame: ST3 CN2
Input	
Parametro: java_file_name	Contenuto file: ""void public main""

Output	
"void": 1, "public": 1, "main": 1	
Test Driver	
Classe: TestTakeJavaClass	Metodo: test_case_4
Oracolo	
Restituisce un dizionario con tutte le parole e frequenza 1	

Tabella 2.54: Metodo dello script test_text_mining.py

Test case ID: TC_2.4_5	Test Frame: ST3 CN3
Input	
Parametro: java_file_name	Contenuto file: 'word word2 word int'
Output	
"word": 2, "int": 1	
Test Driver	
Classe: TestTakeJavaClass	Metodo: test_case_5
Oracolo	
Restituisce un dizionario con tutte le parole e relativa frequenza	

Tabella 2.55: Metodo dello script test_text_mining.py

Test case ID: TC_2.4_6	Test Frame: ST4
Input	
Parametro: java_file_name	Fa riferimento ad una directory

Output	
IsADirectoryError	
Test Driver	
Classe: TestTakeJavaClass	Metodo: test_case_6
Oracolo	
IsADirectoryError	

Tabella 2.56: Metodo dello script test_text_mining.py

Test case ID: TC_2.5_1	Test Frame: DF1
Input	
La directory <code>mining_results</code> non esiste.	
Output	
FileNotFoundException	
Test Driver	
Classe: TestMainMining	Metodo: test_case_1
Oracolo	
FileNotFoundException	

Tabella 2.57: Metodo dello script test_text_mining.py

Test case ID: TC_2.5_2	Test Frame: DF2 ES1
Input	
La directory <code>mining_results</code> esiste, ma la directory di lavoro non è presente.	

Output	
FileNotFoundError	
Test Driver	
Classe: TestMainMining	Metodo: test_case_2
Oracolo	
FileNotFoundError	

Tabella 2.58: Metodo dello script test_text_mining.py

Test case ID: TC_2.5_3	Test Frame: DF2 ES2 CN1
Input	
La directory <code>mining_results</code> esiste e la directory di lavoro è presente ma vuota.	
Output	
Il file <code>text_mining.txt</code> non viene creato	
Test Driver	
Classe: TestMainMining	Metodo: test_case_3
Oracolo	
Il file <code>text_mining.txt</code> non viene creato	

Tabella 2.59: Metodo dello script test_text_mining.py

Test case ID: TC_2.5_4	Test Frame: DF2 ES2 CN2 STC1
Input	

La directory <code>mining_results</code> esiste. La directory di lavoro è presente e non vuota. La variabile <code>cvd_id</code> è un file.	
Output	
NotADirectoryError	
Test Driver	
Classe: TestMainMining	Metodo: <code>test_case_4</code>
Oracolo	
NotADirectoryError	

Tabella 2.60: Metodo dello script `test_text_mining.py`

Test case ID: TC_2.5_5	Test Frame: DF2 ES2 CN2 STC2
Input	
La directory esiste e non è vuota, ma <code>cvd_id</code> è un file non ammesso di nome <code>CHECK.txt</code> .	
Output	
Il file <code>text_mining.txt</code> non viene creato.	
Test Driver	
Classe: TestMainMining	Metodo: <code>test_case_5</code>
Oracolo	
Il file <code>text_mining.txt</code> non viene creato.	

Tabella 2.61: Metodo dello script `test_text_mining.py`

Test case ID: TC_2.5_6	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF1
------------------------	--

Input	
La directory esiste e non è vuota, <code>cvd_id</code> è una directory vuota.	
Output	
Il file <code>text_mining.txt</code> non viene creato.	
Test Driver	
Classe: <code>TestMainMining</code>	Metodo: <code>test_case_6</code>
Oracolo	
Il file <code>text_mining.txt</code> non viene creato.	

Tabella 2.62: Metodo dello script `test_text_mining.py`

Test case ID: <code>TC_2.5_7</code>	Test Frame: <code>DF2 ES2 CN2 STC3 CNC2 STF1</code>
Input	
La directory esiste e non è vuota, <code>cvd_id</code> è una directory non vuota. La variabile <code>folder</code> è un file.	
Output	
<code>NotADirectoryError</code>	
Test Driver	
Classe: <code>TestMainMining</code>	Metodo: <code>test_case_7</code>
Oracolo	
<code>NotADirectoryError</code>	

Tabella 2.63: Metodo dello script `test_text_mining.py`

Test case ID: TC_2.5_8	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF2
Input	
La variabile <code>folder</code> è un file non ammesso <code>.DS_Store</code> .	
Output	
Il file <code>text_mining.txt</code> non viene creato.	
Test Driver	
Classe: <code>TestMainMining</code>	Metodo: <code>test_case_8</code>
Oracolo	
Il file <code>text_mining.txt</code> non viene creato.	

Tabella 2.64: Metodo dello script `test_text_mining.py`

Test case ID: TC_2.5_9	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF1
Input	
La variabile <code>folder</code> è una directory vuota.	
Output	
Il file <code>text_mining.txt</code> non viene creato.	
Test Driver	
Classe: <code>TestMainMining</code>	Metodo: <code>test_case_9</code>
Oracolo	
Il file <code>text_mining.txt</code> non viene creato.	

Tabella 2.65: Metodo dello script `test_text_mining.py`

Test case ID: TC_2.5_10	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 ES2 SV1
Input	
La variabile <code>folder</code> è una directory non vuota e la variabile <code>file</code> è un file Java accessibile.	
Output	
Il file <code>text_mining.txt</code> viene creato con successo.	
Test Driver	
Classe: TestMainMining	Metodo: <code>test_case_10</code>
Oracolo	
Il file <code>text_mining.txt</code> viene creato con successo.	

Tabella 2.66: Metodo dello script `test_text_mining.py`

Test case ID: TC_2.5_11	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 ES1 SV1
Input	
La directory <code>folder</code> non è vuota e contiene sia un file java che il relativo file di text mining	
Output	
Il file <code>text_mining.txt</code> viene sovrascritto e viene creato un nuovo file.	
Test Driver	
Classe: TestMainMining	Metodo: <code>test_case_11</code>
Oracolo	
Il file <code>text_mining.txt</code> viene sovrascritto.	

Tabella 2.67: Metodo dello script test_text_mining.py

Test case ID: TC_2.5_12	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 ES2 SV2
Input	
La variabile <code>file</code> è un file Java accessibile. I permessi impediscono il salvataggio del file <code>text_mining.txt</code> .	
Output	
Il file <code>text_mining.txt</code> non viene creato per motivi di permesso.	
Test Driver	
Classe: TestMainMining	Metodo: test_case_12
Oracolo	
Il file <code>text_mining.txt</code> non viene creato per motivi di permesso.	

Tabella 2.68: Metodo dello script test_text_mining.py

Test case ID: TC_2.5_13	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 ES2 SV3
Input	
La variabile <code>file</code> è un file Java accessibile. Si cerca però di scrivere un tipo di valore errato nel nuovo file.	
Output	
TypeError	
Test Driver	

Classe: TestMainMining	Metodo: test_case_13
Oracolo	
TypeError	

Tabella 2.69: Metodo dello script test_text_mining.py

Test case ID: TC_2.5_14	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC2
Input	
La variabile <code>file</code> è un file Java non accessibile.	
Output	
PermissionError	
Test Driver	
Classe: TestMainMining	Metodo: test_case_14
Oracolo	
PermissionError	

Tabella 2.70: Metodo dello script test_text_mining.py

Test case ID: TC_2.5_15	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI2
Input	
La variabili <code>file</code> è un file <code>.DS_Store</code> che non è ammesso.	
Output	

Il file <code>text_mining.txt</code> non viene creato.	
Test Driver	
Classe: <code>TestMainMining</code>	Metodo: <code>test_case_15</code>
Oracolo	
Il file <code>text_mining.txt</code> non viene creato.	

Tabella 2.71: Metodo dello script `test_text_mining.py`

Test case ID: <code>TC_2.5_16</code>	Test Frame: <code>DF2 ES2 CN2 STC3 CNC2 STF3</code> <code>CNF2 CNFI3</code>
Input	
La variabile <code>file</code> è una cartella.	
Output	
<code>IsADirectoryError</code>	
Test Driver	
Classe: <code>TestMainMining</code>	Metodo: <code>test_case_16</code>
Oracolo	
<code>IsADirectoryError</code>	

Tabella 2.72: Metodo dello script `test_text_mining.py`

Test case ID: <code>TC_2.5_17</code>	Test Frame: <code>DF2 ES2 CN2 STC3 CNC2 STF3</code> <code>CNF2 CNFI4</code>
Input	

La variabile <code>file</code> è un file non java.	
Output	
Il file <code>text_mining.txt</code> viene creato con successo.	
Test Driver	
Classe: <code>TestMainMining</code>	Metodo: <code>test_case_17</code>
Oracolo	
Il file <code>text_mining.txt</code> viene creato con successo.	

Tabella 2.73: Metodo dello script `test_text_mining.py`**Dictionary generator**

Test case ID: <code>TC_2.6_1</code>	Test Frame: <code>DF1</code>
Input	
La directory <code>mining_results</code> non esiste.	
Output	
<code>FileNotFoundError</code>	
Test Driver	
Classe: <code>TestMainDict</code>	Metodo: <code>test_case_1</code>
Oracolo	
L'eccezione <code>FileNotFoundError</code> viene sollevata.	

Tabella 2.74: Metodo dello script `test_dict_generator.py`

Test case ID: TC_2.6_2	Test Frame: DF2 ES2
Input	
La directory <code>mining_results</code> esiste, ma la directory di lavoro <code>repo</code> non esiste.	
Output	
FileNotFoundError	
Test Driver	
Classe: TestMainDict	Metodo: <code>test_case_2</code>
Oracolo	
L'eccezione <code>FileNotFoundError</code> viene sollevata.	

Tabella 2.75: Metodo dello script `test_dict_generator.py`

Test case ID: TC_2.6_3	Test Frame: DF2 ES1 CN1
Input	
La directory <code>mining_results</code> esiste. La directory di lavoro <code>repo</code> esiste ed è vuota.	
Output	
Il file viene creato con un dizionario vuoto.	
Test Driver	
Classe: TestMainDict	Metodo: <code>test_case_3</code>
Oracolo	
Il file viene creato con un dizionario vuoto.	

Tabella 2.76: Metodo dello script `test_dict_generator.py`

Test case ID: TC_2.6_4	Test Frame: DF2 ES1 CN2 STC1
Input	
La directory <code>mining_results</code> esiste. La directory di lavoro <code>repo</code> esiste e contiene file. La variabile <code>cvd_id</code> è un file.	
Output	
NotADirectoryError.	
Test Driver	
Classe: TestMainDict	Metodo: test_case_4
Oracolo	
Viene sollevata un'eccezione NotADirectoryError.	

Tabella 2.77: Metodo dello script test_dict_generator.py

Test case ID: TC_2.6_5	Test Frame: DF2 ES1 CN2 STC2
Input	
La directory <code>mining_results</code> esiste. La directory di lavoro <code>repo</code> esiste e contiene file. La variabile <code>cvd_id</code> è il file "CHECK.txt"	
Output	
Il file viene creato con un dizionario vuoto.	
Test Driver	
Classe: TestMainDict	Metodo: test_case_5
Oracolo	
Il file viene creato con un dizionario vuoto.	

Tabella 2.78: Metodo dello script test_dict_generator.py

Test case ID: TC_2.6_6	Test Frame: DF2 ES1 CN2 STC3 SC1
Input	
La directory <code>mining_results</code> esiste. La directory di lavoro <code>repo</code> esiste e contiene file. La variabile <code>cvd_id</code> è una cartella non vuota. La sottocartella è vuota.	
Output	
Il file viene creato con un dizionario vuoto.	
Test Driver	
Classe: TestMainDict	Metodo: test_case_6
Oracolo	
Il file viene creato con un dizionario vuoto.	

Tabella 2.79: Metodo dello script test_dict_generator.py

Test case ID: TC_2.6_7	Test Frame: DF2 ES1 CN2 STC3 SC2
Input	
La variabile <code>cvd_id</code> è una cartella non vuota e contiene un file qualsiasi invece di sottocartelle.	
Output	
NotADirectoryError.	
Test Driver	
Classe: TestMainDict	Metodo: test_case_7

Oracolo
Il file viene creato con un dizionario contenente le informazioni corrette sui repository.

Tabella 2.80: Metodo dello script test_dict_generator.py

Test case ID: TC_2.6_8	Test Frame: DF2 ES1 CN2 STC3 SC2 ECC1
Input	
La variabile <code>cvd_id</code> è una cartella non vuota e contiene il file <code>.DS_Store</code> .	
Output	
Il file viene creato con un dizionario vuoto.	
Test Driver	
Classe: TestMainDict	Metodo: test_case_8
Oracolo	
L'eccezione viene sollevata e il file non viene creato.	

Tabella 2.81: Metodo dello script test_dict_generator.py

Test case ID: TC_2.6_9	Test Frame: DF2 ES1 CN2 STC3 SC2 ECC2
Input	
La sottocartella <code>folder</code> è una cartella vuota.	
Output	
Il file viene creato con un dizionario vuoto.	
Test Driver	
Classe: TestMainDict	Metodo: test_case_9

Oracolo
L'eccezione viene sollevata e il file non viene creato.

Tabella 2.82: Metodo dello script test_dict_generator.py

Test case ID: TC_2.6_10	Test Frame: DF2 ES1 CN2 STC4
Input	
Input	
La sottocartella <code>folder</code> contiene un unico file <code>.DS_Store</code> .	
Output	
Il file viene creato con un dizionario vuoto.	
Test Driver	
Classe: TestMainDict	Metodo: test_case_10
Oracolo	
Il file viene creato con il dizionario corretto, elaborato in base ai parametri in input.	

Tabella 2.83: Metodo dello script test_dict_generator.py

Test case ID: TC_2.6_11	Test Frame: DF2 ES1 CN2 STC3 CNC2 STF3 CNF2 CNFI3
Input	
La sottocartella <code>folder</code> contiene un file che non contiene "text_mining.txt" nel nome	
Output	
Il file viene creato con un dizionario vuoto.	

Test Driver	
Classe: TestMainDict	Metodo: test_case_11
Oracolo	
Il file viene creato con un dizionario vuoto.	

Tabella 2.84: Metodo dello script test_dict_generator.py

Test case ID: TC_2.6_12	Test Frame: DF2 ES1 CN2 STC3 CNC2 STF3 CNF2 CNFI4
Input	
La sottocartella <code>folder</code> contiene un una cartella <code>file</code> che ha "text_mining.txt" nel nome	
Output	
Viene sollevata un'eccezione <code>IsADirectoryError</code> .	
Test Driver	
Classe: TestMainDict	Metodo: test_case_12
Oracolo	
Viene sollevata un'eccezione <code>IsADirectoryError</code> .	

Tabella 2.85: Metodo dello script test_dict_generator.py

Test case ID: TC_2.6_13	Test Frame: DF2 ES1 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 CF1
Input	
La sottocartella <code>folder</code> non ha "text_mining.txt" nel nome e contiene un file di text mining vuoto.	

Output	
Viene sollevata un'eccezione <code>SyntaxError</code> .	
Test Driver	
Classe: <code>TestMainDict</code>	Metodo: <code>test_case_13</code>
Oracolo	
Viene sollevata un'eccezione <code>SyntaxError</code> .	

Tabella 2.86: Metodo dello script `test_dict_generator.py`

Test case ID: <code>TC_2.6_14</code>	Test Frame: <code>DF2 ES1 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 CF2</code>
Input	
La sottocartella <code>folder</code> non ha "text_mining.txt" nel nome e contiene un file di text mining con un dizionario malformato.	
Output	
Viene sollevata un'eccezione <code>SyntaxError</code> .	
Test Driver	
Classe: <code>TestMainDict</code>	Metodo: <code>test_case_14</code>
Oracolo	
Viene sollevata un'eccezione <code>SyntaxError</code> .	

Tabella 2.87: Metodo dello script `test_dict_generator.py`

Test case ID: <code>TC_2.6_15</code>	Test Frame: <code>DF2 ES1 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 CF3 SV1</code>
--------------------------------------	--

Input	
La sottocartella <code>folder</code> non ha "text_mining.txt" nel nome e contiene un file di text mining con un dizionario ben formato. Vi sono i permessi per salvare correttamente il file di text mining.	
Output	
Il file viene creato con un dizionario basato sui dati di text mining.	
Test Driver	
Classe: <code>TestMainDict</code>	Metodo: <code>test_case_15</code>
Oracolo	
Il file viene creato con un dizionario basato sui dati di text mining.	

Tabella 2.88: Metodo dello script `test_dict_generator.py`

Test case ID: <code>TC_2.6_16</code>	Test Frame: <code>DF2 ES1 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 CF3 SV2</code>
Input	
La sottocartella <code>folder</code> non ha "text_mining.txt" nel nome e contiene un file di text mining con un dizionario ben formato. Non vi sono, però, i permessi per salvare correttamente il file di text mining.	
Output	
Viene sollevata un'eccezione <code>PermissionError</code> .	
Test Driver	
Classe: <code>TestMainDict</code>	Metodo: <code>test_case_16</code>
Oracolo	

Viene sollevata un'eccezione `PermissionError`.

Tabella 2.89: Metodo dello script `test_dict_generator.py`

Test case ID: TC_2.6_17	Test Frame: DF2 ES1 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC1 CF4
Input	
La sottocartella <code>folder</code> non ha "text_mining.txt" nel nome e contiene un file di text mining, ma non contiene un dizionario.	
Output	
Viene sollevata un'eccezione <code>SyntaxError</code> .	
Test Driver	
Classe: <code>TestMainDict</code>	Metodo: <code>test_case_17</code>
Oracolo	
Viene sollevata un'eccezione <code>SyntaxError</code> .	

Tabella 2.90: Metodo dello script `test_dict_generator.py`

Test case ID: TC_2.6_18	Test Frame: DF2 ES1 CN2 STC3 CNC2 STF3 CNF2 CNFI1 AC2
Input	
La sottocartella <code>folder</code> non ha "text_mining.txt" nel nome e contiene un file di text mining, ma non è accessibile.	
Output	
Viene sollevata un'eccezione <code>PermissionError</code> .	

Test Driver	
Classe: TestMainDict	Metodo: test_case_18
Oracolo	
Viene sollevata un'eccezione <code>PermissionError</code> .	

Tabella 2.91: Metodo dello script `test_dict_generator.py`**Less element text mining**

Test case ID: TC_2.7_1	Test Frame: CN1 CD1
Input	
Parametro: fake_dic	{ }
Output	
{ }	
Test Driver	
Classe: TestSplitCamelCase	Metodo: test_case_1
Oracolo	
Restituisce un dizionario vuoto	

Tabella 2.92: Metodo dello script `test_less_element_text_mining.py`

Test case ID: TC_2.7_2	Test Frame: CN1 CD2
Input	
Parametro: fake_dic	{ 'CamelCase': 1, 'SplitThis': 2, 'AnotherTest': 3 }

Output	
{ 'camel': 1, 'case': 1, 'split': 2, 'this': 2, 'another': 3, 'test': 3 }	
Test Driver	
Classe: TestSplitCamelCase	Metodo: test_case_2
Oracolo	
Restituisce un dizionario con chiavi le parole divise in minuscolo e la frequenza come valore	

Tabella 2.93: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.7_3	Test Frame: CN1 CD3
Input	
Parametro: fake_dic	{ 'lowercase': 1, 'uppercase': 2, 'allcaps': 3 }
Output	
{ 'lowercase': 1, 'uppercase': 2, 'allcaps': 3 }	
Test Driver	
Classe: TestSplitCamelCase	Metodo: test_case_3
Oracolo	
Restituisce un dizionario uguale a quello in input	

Tabella 2.94: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.7_4	Test Frame: CN1 CD4
------------------------	---------------------

Input	
Parametro: fake_dic	{'CamelCaseKey': 4, 'lowercasekey': 2, 'AnotherCamelCase': 3}
Output	
{'camel': 7, 'case': 7, 'key': 4, 'lowercasekey': 2, 'another': 3}	
Test Driver	
Classe: TestSplitCamelCase	Metodo: test_case_4
Oracolo	
Restituisce un dizionario con chiavi le parole divise in minuscolo e la frequenza come valore	

Tabella 2.95: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.7_5	Test Frame: CN2 CD2
Input	
Parametro: fake_dic	{'CamelCase': None, 'AnotherTest': 'string_value', 'validKey': 5}
Output	
{'camel': None, 'case': None, 'another': 'string_value', 'test': 'string_value', 'valid': 5, 'key': 5}	
Test Driver	
Classe: TestSplitCamelCase	Metodo: test_case_5
Oracolo	
Restituisce un dizionario con chiavi le parole divise in minuscolo e la frequenza come valore	

Tabella 2.96: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.7_6	Test Frame: CN2 CD3
Input	
Parametro: fake_dic	{'lowercase': None, 'uppercase': 2, '2': 'ciao'}
Output	
{'lowercase': None, 'uppercase': 2, '2': 'ciao'}	
Test Driver	
Classe: TestSplitCamelCase	Metodo: test_case_6
Oracolo	
Restituisce un dizionario uguale a quello in input	

Tabella 2.97: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.7_7	Test Frame: CN2 CD4
Input	
Parametro: fake_dic	{'CamelCase': None, 'AnotherCaseTest': 'string_value', 'validKey': 5}
Output	
TypeError	
Test Driver	
Classe: TestSplitCamelCase	Metodo: test_case_7

Oracolo
TypeError

Tabella 2.98: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.7_8	Test Frame: CN3
Input	
Parametro: fake_dic	{1: 2, 'AnotherCaseTest': 3, 'validKey': 5}
Output	
TypeError	
Test Driver	
Classe: TestSplitCamelCase	Metodo: test_case_8
Oracolo	
TypeError	

Tabella 2.99: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.7_9	Test Frame: CN4
Input	
Parametro: fake_dic	['ad', 1]
Output	
TypeError	
Test Driver	

Classe: TestSplitCamelCase	Metodo: test_case_9
Oracolo	
TypeError	

Tabella 2.100: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.8_1	Test Frame: DF1
Input	
La directory mining_results non esiste.	
Output	
FileNotFoundError.	
Test Driver	
Classe: TestInitialize	Metodo: test_case_1
Oracolo	
FileNotFoundError.	

Tabella 2.101: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.8_2	Test Frame: DF2 CN1
Input	
La directory mining_results è vuota.	
Output	
None.	
Test Driver	

Classe: TestInitialize	Metodo: test_case_2
Oracolo	
None.	

Tabella 2.102: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.8_3	Test Frame: DF2 CN2 ST1 CND1
Input	
<p>La directory <code>mining_results</code> contiene il file "text_mining_dict.txt" con all'interno un dizionario valido:</p> <pre>'''{'package': 1, 'org': 4, 'apache': 4, 'import': 7, 'AccessControlFilter': 2, 'Logger': 2, 'LoggerFactory': 2, 'HttpServletResponse': 2}'''</pre>	
Output	
<pre>{'package': 1, 'org': 4, 'apache': 4, 'import': 7, 'access': 2, 'control': 2, 'factory': 2, 'filter': 2, 'logger': 4, 'http': 2, 'servlet': 2, 'response': 2}</pre>	
Test Driver	
Classe: TestInitialize	Metodo: test_case_3
Oracolo	
Restituisce un dizionario con chiavi le parole divise in minuscolo e la frequenza come valore.	

Tabella 2.103: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.8_4	Test Frame: DF2 CN2 ST1 CND2
Input	

La directory <code>mining_results</code> contiene il file "text_mining_dict.txt" con all'interno un dizionario con chiavi non valide: <pre>'''{ 'CamelCase': None, '1': 'string_value', 2: 5}'''</pre>	
Output	
TypeError.	
Test Driver	
Classe: <code>TestInitialize</code>	Metodo: <code>test_case_4</code>
Oracolo	
TypeError.	

Tabella 2.104: Metodo dello script `test_less_element_text_mining.py`

Test case ID: TC_2.8_5	Test Frame: DF2 CN2 ST1 CND3
Input	
La directory <code>mining_results</code> contiene il file "text_mining_dict.txt" però vuoto	
Output	
SyntaxError.	
Test Driver	
Classe: <code>TestInitialize</code>	Metodo: <code>test_case_5</code>
Oracolo	
SyntaxError.	

Tabella 2.105: Metodo dello script `test_less_element_text_mining.py`

Test case ID: TC_2.8_6	Test Frame: DF2 CN2 ST1 CND4
Input	
La directory <code>mining_results</code> contiene il file "text_mining_dict.txt" che non contiene un dizionario: <code>'[2, 3, 4]'</code>	
Output	
TypeError.	
Test Driver	
Classe: <code>TestInitialize</code>	Metodo: <code>test_case_6</code>
Oracolo	
TypeError.	

Tabella 2.106: Metodo dello script `test_less_element_text_mining.py`

Test case ID: TC_2.8_7	Test Frame: DF2 CN2 ST2
Input	
La directory <code>mining_results</code> contiene il file "text_mining_dict.txt" però non è accessibile.	
Output	
PermissionError.	
Test Driver	
Classe: <code>TestInitialize</code>	Metodo: <code>test_case_7</code>
Oracolo	
PermissionError.	

Tabella 2.107: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.8_8	Test Frame: DF2 CN2 ST3
Input	
La directory <code>mining_results</code> non contiene il file "text_mining_dict.txt".	
Output	
None.	
Test Driver	
Classe: <code>TestInitialize</code>	Metodo: <code>test_case_8</code>
Oracolo	
None.	

Tabella 2.108: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.9_1	Test Frame: AC1
Input	
Parametro: <code>dizionario_finale</code>	<code>{'key': 2, 'import': 4}</code>
FilteredTextMining.txt non accessibile	
Output	
<code>PermissionError</code>	
Test Driver	
Classe: <code>TestWriteToFile</code>	Metodo: <code>test_case_1</code>
Oracolo	

PermissionError

Tabella 2.109: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.9_2	Test Frame: AC2 CN1 SC1
Input	
Parametro: dizionario_finale	{'key': 2, 'import': 4}
Output	
Crea un file che contiene il dizionario passato in input	
Test Driver	
Classe: TestWriteToFile	Metodo: test_case_2
Oracolo	
Crea un file che contiene il dizionario passato in input	

Tabella 2.110: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.9_3	Test Frame: AC2 CN1 SC2
Input	
Parametro: dizionario_finale	{'key': 2, 'import': 4}
FilteredTextMining.txt non salvabile per permessi insufficienti	
Output	
PermissionError	
Test Driver	
Classe: TestWriteToFile	Metodo: test_case_3

Oracolo
PermissionError

Tabella 2.111: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.9_4	Test Frame: AC2 CN2 SC1
Input	
Parametro: dizionario_finale	{ }
Output	
Crea un file che contiene un dizionario vuoto	
Test Driver	
Classe: TestWriteToFile	Metodo: test_case_4
Oracolo	
Crea un file che contiene un dizionario vuoto	

Tabella 2.112: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.9_5	Test Frame: AC2 CN2 SC2
Input	
Parametro: dizionario_finale	{ }
FilteredTextMining.txt non salvabile per permessi insufficienti	
Output	
PermissionError	
Test Driver	

Classe: TestWriteToFile	Metodo: test_case_5
Oracolo	
PermissionError	

Tabella 2.113: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.9_6	Test Frame: AC2 CN3 SC1
Input	
Parametro: dizionario_finale	{1: None, 'uppercase': 2, '2': 'ciao' }
Output	
Crea un file che contiene il dizionario non valido passato in input	
Test Driver	
Classe: TestWriteToFile	Metodo: test_case_6
Oracolo	
Crea un file che contiene il dizionario non valido passato in input	

Tabella 2.114: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.9_7	Test Frame: AC2 CN3 SC2
Input	
Parametro: dizionario_finale	{1: None, 'uppercase': 2, '2': 'ciao' }
FilteredTextMining.txt non salvabile per permessi insufficienti	

Output	
PermissionError	
Test Driver	
Classe: TestWriteToFile	Metodo: test_case_7
Oracolo	
PermissionError	

Tabella 2.115: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.9_8	Test Frame: AC2 CN4 SC1
Input	
Parametro: dizionario_finale	['prova', 'tipologia', 'diversa']
Output	
Crea un file che contiene l'input dato	
Test Driver	
Classe: TestWriteToFile	Metodo: test_case_8
Oracolo	
Crea un file che contiene l'input dato	

Tabella 2.116: Metodo dello script test_less_element_text_mining.py

Test case ID: TC_2.9_9	Test Frame: AC2 CN4 SC2
Input	
Parametro: dizionario_finale	['prova', 'tipologia', 'diversa']

FilteredTextMining.txt non salvabile per permessi insufficienti	
Output	
PermissionError	
Test Driver	
Classe: TestWriteToFile	Metodo: test_case_9
Oracolo	
PermissionError	

Tabella 2.117: Metodo dello script test_less_element_text_mining.py**CSV creator**

Test case ID: TC_2.10_1	Test Frame: DF1
Input	
La directory mining_results non esiste.	
Output	
FileNotFoundError	
Test Driver	
Classe: TestMainCSV	Metodo: test_case_1
Oracolo	
FileNotFoundError	

Tabella 2.118: Metodo dello script test_creator_csv_for_TextMining.py

Test case ID: TC_2.10_2	Test Frame: DF2 ES2
-------------------------	---------------------

Input	
La directory <code>mining_results</code> esiste. Il file <code>FilteredTextMining.txt</code> non esiste.	
Output	
FileNotFoundError	
Test Driver	
Classe: <code>TestMainCSV</code>	Metodo: <code>test_case_2</code>
Oracolo	
FileNotFoundError	

Tabella 2.119: Metodo dello script `test_creator_csv_for_TextMining.py`

Test case ID: TC_2.10_3	Test Frame: DF2 ES1 CN1
Input	
Il file <code>FilteredTextMining.txt</code> esiste ma è vuoto.	
Output	
SyntaxError	
Test Driver	
Classe: <code>TestMainCSV</code>	Metodo: <code>test_case_3</code>
Oracolo	
SyntaxError	

Tabella 2.120: Metodo dello script `test_creator_csv_for_TextMining.py`

Test case ID: TC_2.10_4	Test Frame: DF2 ES1 CN3
-------------------------	-------------------------

Input	
Il file <code>FilteredTextMining.txt</code> esiste ma contiene dati non in formato dizionario.	
Output	
AttributeError	
Test Driver	
Classe: TestMainCSV	Metodo: test_case_4
Oracolo	
AttributeError	

Tabella 2.121: Metodo dello script test_creator_csv_for_TextMining.py

Test case ID: TC_2.10_5	Test Frame: DF2 ES1 CN2 ESDR2
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La directory <code>repo</code> non esiste.	
Output	
FileNotFoundError	
Test Driver	
Classe: TestMainCSV	Metodo: test_case_5
Oracolo	
FileNotFoundError	

Tabella 2.122: Metodo dello script test_creator_csv_for_TextMining.py

Test case ID: TC_2.10_6	Test Frame: DF2 ES1 CN2 ESDR1 CND1
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La directory <code>repo</code> esiste ma è vuota.	
Output	
Viene creato un file csv con solo le parole chiavi come colonne.	
Test Driver	
Classe: TestMainCSV	Metodo: test_case_6
Oracolo	
Viene creato un file csv con solo le parole chiavi come colonne.	

Tabella 2.123: Metodo dello script `test_creator_csv_for_TextMining.py`

Test case ID: TC_2.10_7	Test Frame: DF2 ES1 CN2 ESDR1 CND2 STC1
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La directory <code>repo</code> non è vuota e contiene un file ammesso. La variabile <code>cvd_id</code> è un file.	
Output	
NotADirectoryError	
Test Driver	
Classe: TestMainCSV	Metodo: test_case_7
Oracolo	

NotADirectoryError

Tabella 2.124: Metodo dello script test_creator_csv_for_TextMining.py

Test case ID: TC_2.10_8	Test Frame: DF2 ES1 CN2 ESDR1 CND2 STC2
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La directory <code>RepositoryMining</code> non è vuota e contiene un file non ammesso. La variabile <code>cvd_id</code> è un file di nome <code>ERROR.txt</code> .	
Output	
Viene creato un file csv con solo le parole chiavi come colonne.	
Test Driver	
Classe: TestMainCSV	Metodo: test_case_8
Oracolo	
Viene creato un file csv con solo le parole chiavi come colonne.	

Tabella 2.125: Metodo dello script test_creator_csv_for_TextMining.py

Test case ID: TC_2.10_9	Test Frame: DF2 ES1 CN2 ESDR1 CND2 STC3 CNC1
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La variabile <code>cvd_id</code> è una directory vuota.	
Output	

Viene creato un file csv con solo le parole chiavi come colonne.	
Test Driver	
Classe: TestMainCSV	Metodo: test_case_9
Oracolo	
Viene creato un file csv con solo le parole chiavi come colonne.	

Tabella 2.126: Metodo dello script test_creator_csv_for_TextMining.py

Test case ID: TC_2.10_10	Test Frame: DF2 ES1 CN2 ESDR1 CND2 STC3 CNC2 STF1
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La variabile <code>cvd_id</code> è una directory non vuota e contiene un file.	
Output	
NotADirectoryError	
Test Driver	
Classe: TestMainCSV	Metodo: test_case_10
Oracolo	
NotADirectoryError	

Tabella 2.127: Metodo dello script test_creator_csv_for_TextMining.py

Test case ID: TC_2.10_11	Test Frame: DF2 ES1 CN2 ESDR1 CND2 STC3 STF2
Input	

Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La variabile <code>cvd_id</code> è una directory non vuota e contiene un file non ammesso. La variabile <code>folder</code> è un file di nome <code>.DS_Store</code>	
Output	
Viene creato un file csv con solo le parole chiavi come colonne.	
Test Driver	
Classe: <code>TestMainCSV</code>	Metodo: <code>test_case_11</code>
Oracolo	
Viene creato un file csv con solo le parole chiavi come colonne.	

Tabella 2.128: Metodo dello script `test_creator_csv_for_TextMining.py`

Test case ID: <code>TC_2.10_12</code>	Test Frame: <code>DF2 ES1 CN2 ESDR1 CND2 STC3 STF3 CNF1</code>
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La variabile <code>folder</code> è una directory vuota.	
Output	
Viene creato un file csv con solo le parole chiavi come colonne.	
Test Driver	
Classe: <code>TestMainCSV</code>	Metodo: <code>test_case_12</code>
Oracolo	
Viene creato un file csv con solo le parole chiavi come colonne.	

Tabella 2.129: Metodo dello script `test_creator_csv_for_TextMining.py`

Test case ID: TC_2.10_13	Test Frame: DF2 ES1 CN2 ESDR1 CND2 STC3 STF1 CNF2 CNFI1 AC1 CNTM1 SC1
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La variabile <code>file</code> è un file di <code>text_mining.txt</code> accessibile e con un dizionario valido.	
Output	
Viene creato un file csv con le parole chiavi come colonne e una riga per ogni file con le frequenze.	
Test Driver	
Classe: <code>TestMainCSV</code>	Metodo: <code>test_case_13</code>
Oracolo	
Viene creato un file csv con le parole chiavi come colonne e una riga per ogni file con le frequenze.	

Tabella 2.130: Metodo dello script `test_creator_csv_for_TextMining.py`

Test case ID: TC_2.10_14	Test Frame: DF2 ES1 CN2 ESDR1 CND2 STC3 STF1 CNF2 CNFI1 AC1 CNTM1 SC2
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La variabile <code>file</code> è un file di nome <code>text_mining.txt</code> accessibile e con un dizionario valido. Il file csv però non può essere salvato correttamente.	
Output	
PermissionError	
Test Driver	

Classe: TestMainCSV	Metodo: test_case_14
Oracolo	
PermissionError	

Tabella 2.131: Metodo dello script test_creator_csv_for_TextMining.py

Test case ID: TC_2.10_15	Test Frame: DF2 ES1 CN2 ESDR1 CND2 STC3 STF1 CNF2 CNFI1 AC1 CNTM2 SC1
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La variabile <code>file</code> è un file di nome <code>text_mining.txt</code> accessibile e vuoto.	
Output	
Viene creato un file csv con le parole chiavi come colonne e una riga per ogni file con le frequenze a 0.	
Test Driver	
Classe: TestMainCSV	Metodo: test_case_15
Oracolo	
Viene creato un file csv con le parole chiavi come colonne e una riga per ogni file con le frequenze a 0.	

Tabella 2.132: Metodo dello script test_creator_csv_for_TextMining.py

Test case ID: TC_2.10_16	Test Frame: DF2 ES1 CN2 ESDR1 CND2 STC3 STF1 CNF2 CNFI1 AC1 CNTM3
Input	

Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La variabile <code>file</code> è un file di nome <code>text_mining.txt</code> accessibile ma con contenuto non valido.	
Output	
TypeError	
Test Driver	
Classe: <code>TestMainCSV</code>	Metodo: <code>test_case_16</code>
Oracolo	
TypeError	

Tabella 2.133: Metodo dello script `test_creator_csv_for_TextMining.py`

Test case ID: <code>TC_2.10_17</code>	Test Frame: <code>DF2 ES1 CN2 ESDR1 CND2 STC3 STF1 CNF2 CNFI1 AC2</code>
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La variabile <code>file</code> è un file di nome <code>text_mining.txt</code> , contenente un dizionario valido, ma il file non è accessibile.	
Output	
PermissionError	
Test Driver	
Classe: <code>TestMainCSV</code>	Metodo: <code>test_case_17</code>
Oracolo	
PermissionError	

Tabella 2.134: Metodo dello script `test_creator_csv_for_TextMining.py`

Test case ID: TC_2.10_18	Test Frame: DF2 ES1 CN2 ESDR1 CND2 STC3 STF1 CNF2 CNFI2
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La variabile <code>file</code> è un file non ammesso di nome <code>.DS_Store</code> .	
Output	
Viene creato un file csv con solo le parole chiavi come colonne.	
Test Driver	
Classe: TestMainCSV	Metodo: <code>test_case_18</code>
Oracolo	
Viene creato un file csv con solo le parole chiavi come colonne.	

Tabella 2.135: Metodo dello script `test_creator_csv_for_TextMining.py`

Test case ID: TC_2.10_19	Test Frame: DF2 ES1 CN2 ESDR1 CND2 STC3 STF1 CNF2 CNFI3
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La variabile <code>file</code> è una directory.	
Output	
Viene creato un file csv con solo le parole chiavi come colonne.	
Test Driver	
Classe: TestMainCSV	Metodo: <code>test_case_19</code>
Oracolo	

Viene creato un file csv con solo le parole chiavi come colonne.

Tabella 2.136: Metodo dello script test_creator_csv_for_TextMining.py

Test case ID: TC_2.10_20	Test Frame: DF2 ES1 CN2 ESDR1 CND2 STC3 STF1 CNF2 CNFI4
Input	
Il file <code>FilteredTextMining.txt</code> contiene un dizionario valido. La variabile <code>file</code> è una directory con la sottostringa "text_mining.txt" nel nome.	
Output	
IsADirectoryError	
Test Driver	
Classe: TestMainCSV	Metodo: test_case_20
Oracolo	
IsADirectoryError	

Tabella 2.137: Metodo dello script test_creator_csv_for_TextMining.py

2.1.3 Rifinitura dati analisi statica

ASA vulnerability dictionary generator

Test case ID: TC_3.1_1	Test Frame: EN2 EP1 NP3 FCP1 TCP1 S1
Input	
<p>Il file <code>RepositoryMining_ASAResults_neg.csv</code> non è presente.</p> <p>Il file <code>RepositoryMining_ASAResults_pos.csv</code> è presente, è valido e contiene vulnerabilità</p>	

Output	
ASA_dict.csv con solo elementi con classe 'pos' e tipo 'vulnerability'.	
Test Driver	
Classe: TestMainASAVulnDictGen	Metodo: test_case_1_vuln
Oracolo	
ASA_dict.csv con solo elementi con classe 'pos' e tipo 'vulnerability'.	

Tabella 2.138: Metodo dello script test_asa_vuln_dict_gen.py

Test case ID: TC_3.1_2	Test Frame: EN1 NN3 FCN1 TCN1 EP2 S1
Input	
RepositoryMining_ASAResults_neg.csv è presente, è valido e contiene vulnerabilità RepositoryMining_ASAResults_pos.csv non è presente.	
Output	
ASA_dict.csv con solo elementi con classe 'neg' e tipo 'vulnerability'.	
Test Driver	
Classe: TestMainASAVulnDictGen	Metodo: test_case_2_vuln
Oracolo	
ASA_dict.csv con solo elementi con classe 'neg' e tipo 'vulnerability'	

Tabella 2.139: Metodo dello script test_asa_vuln_dict_gen.py

Test case ID: TC_3.1_3	Test Frame: S2
Input	

RepositoryMining_ASAResults_neg.csv e RepositoryMining_ASAResults_pos.csv sono presenti, validi e contengono vulnerabilità ASA_dict.csv non è accessibile	
Output	
PermissionError	
Test Driver	
Classe: TestMainASAVulnDictGen	Metodo: test_case_3_vuln
Oracolo	
PermissionError.	

Tabella 2.140: Metodo dello script test_asa_vuln_dict_gen.py

Test case ID: TC_3.1_4	Test Frame: EN1 FCN2
Input	
RepositoryMining_ASAResults_neg.csv è presente ma ha un formato invalido. RepositoryMining_ASAResults_pos.csv è presente, è valido e contiene vulnerabilità	
Output	
IndexError	
Test Driver	
Classe: TestMainASAVulnDictGen	Metodo: test_case_4_vuln
Oracolo	
IndexError	

Tabella 2.141: Metodo dello script test_asa_vuln_dict_gen.py

Test case ID: TC_3.1_5	Test Frame: EP1 FCP2
Input	
RepositoryMining_ASAResults_neg.csv è presente, è valido e contiene vulnerabilità. RepositoryMining_ASAResults_pos.csv è presente ma ha un formato invalido.	
Output	
IndexError	
Test Driver	
Classe: TestMainASAVulnDictGen	Metodo: test_case_5_vuln
Oracolo	
IndexError	

Tabella 2.142: Metodo dello script test_asa_vuln_dict_gen.py

Test case ID: TC_3.1_6	Test Frame: EN1 NN2 FCN1 TCN2 EP1 FCP1 NP2 TCP2 S1
Input	
RepositoryMining_ASAResults_neg.csv e RepositoryMining_ASAResults_pos.csv sono presenti e validi, hanno 1 record e non contengono vulnerabilità	
Output	
ASA_dict.csv è vuoto.	
Test Driver	
Classe: TestMainASAVulnDictGen	Metodo: test_case_6_vuln
Oracolo	

ASA_dict.csv è vuoto.

Tabella 2.143: Metodo dello script test_asa_vuln_dict_gen.py

Test case ID: TC_3.1_7	Test Frame: EN1 NN2 FCN1 TCN1 EP1 FCP1 NP2 TCP1 S1
Input	
RepositoryMining_ASAResults_neg.csv e RepositoryMining_ASAResults_pos.csv sono presenti e validi, hanno 1 record e contengono vulnerabilità	
Output	
ASA_dict.csv contiene elementi di classe 'pos' e 'neg'.	
Test Driver	
Classe: TestMainASAVulnDictGen	Metodo: test_case_7_vuln
Oracolo	
ASA_dict.csv contiene elementi di classe 'pos' e 'neg' e tipo 'vulnerability.	

Tabella 2.144: Metodo dello script test_asa_vuln_dict_gen.py

Test case ID: TC_3.1_8	Test Frame: EN1 NN1 FCN1 EP1 NP1 FNP FCP1 S1
Input	
RepositoryMining_ASAResults_neg.csv e RepositoryMining_ASAResults_pos.csv sono presenti, validi e non contengono record.	
Output	

ASA_dict.csv vuoto	
Test Driver	
Classe: TestMainASAVulnDictGen	Metodo: test_case_8_vuln
Oracolo	
ASA_dict.csv vuoto	

Tabella 2.145: Metodo dello script test_asa_vuln_dict_gen.py**ASA rules dictionary generation**

Test case ID: TC_3.2_1	Test Frame: EN2 EP1 NP3 FCP1 TCP1 S1
Input	
RepositoryMining_ASAResults_neg.csv non è presente. RepositoryMining_ASAResults_pos.csv è presente, è valido e contiene vulnerabilità.	
Output	
ASA_rules_dict.csv contiene solo regole relative a vulnerabilità di classe 'pos'.	
Test Driver	
Classe: TestMainRulesDictGenASA	Metodo: test_case_1
Oracolo	
ASA_rules_dict.csv contiene solo regole relative a vulnerabilità di classe 'pos'.	

Tabella 2.146: Metodo dello script test_rules_dict_gen_asa.py

Test case ID: TC_3.2_2	Test Frame: EN1 NN3 FCN1 TCN1 EP2 S1
Input	

RepositoryMining_ASAResults_neg.csv è presente, è valido e contiene vulnerabilità RepositoryMining_ASAResults_pos.csv non è presente.	
Output	
ASA_rules_dict.csv contiene solo regole relative a vulnerabilità di classe 'neg'.	
Test Driver	
Classe: TestMainRulesDictGenASA	Metodo: test_case_2
Oracolo	
ASA_rules_dict.csv contiene solo regole relative a vulnerabilità di classe 'neg'.	

Tabella 2.147: Metodo dello script test_rules_dict_gen_asa.py

Test case ID: TC_3.2_3	Test Frame: S2
Input	
RepositoryMining_ASAResults_neg.csv e RepositoryMining_ASAResults_pos.csv sono presenti, validi e contengono vulnerabilità ASA_rules_dict.csv non è accessibile	
Output	
PermissionError.	
Test Driver	
Classe: TestMainRulesDictGenASA	Metodo: test_case_3
Oracolo	
PermissionError.	

Tabella 2.148: Metodo dello script test_rules_dict_gen_asa.py

Test case ID: TC_3.2_4	Test Frame: EN1 FCN2
Input	
RepositoryMining_ASAResults_neg.csv è presente ma ha un formato non valido. RepositoryMining_ASAResults_pos.csv è presente, è valido e contiene vulnerabilità.	
Output	
IndexError	
Test Driver	
Classe: TestMainRulesDictGenASA	Metodo: test_case_4
Oracolo	
IndexError	

Tabella 2.149: Metodo dello script test_rules_dict_gen_asa.py

Test case ID: TC_3.2_5	Test Frame: EP1 FCP2
Input	
RepositoryMining_ASAResults_neg.csv è presente, è valido e contiene vulnerabilità. RepositoryMining_ASAResults_pos.csv è presente ma ha un formato non valido.	
Output	
IndexError	
Test Driver	
Classe: TestMainRulesDictGenASA	Metodo: test_case_5
Oracolo	
IndexError	

Tabella 2.150: Metodo dello script test_rules_dict_gen_asa.py

Test case ID: TC_3.2_6	Test Frame: EN1 NN2 FCN1 TCN2 EP1 NP2 FCP1 TCP2 S1
Input	
I file RepositoryMining_ASAResults_pos.csv e RepositoryMining_ASAResults_neg.csv sono presenti, hanno 1 record e non contengono vulnerabilità.	
Output	
ASA_rules_dict.csv è vuoto.	
Test Driver	
Classe: TestMainRulesDictGenASA	Metodo: test_case_6
Oracolo	
ASA_rules_dict.csv è vuoto.	

Tabella 2.151: Metodo dello script test_rules_dict_gen_asa.py

Test case ID: TC_3.2_7	Test Frame: EN1 NN2 FCN1 TCN1 EP1 NP2 FCP1 TCP2 S1
Input	
RepositoryMining_ASAResults_pos.csv e RepositoryMining_ASAResults_neg.csv sono presenti, validi, hanno 1 record e contengono vulnerabilità.	
Output	

ASA_rules_dict.csv contiene regole relative a vulnerabilità di classe 'pos' e 'neg' e tipo 'vulnerability'.	
Test Driver	
Classe: TestMainASARulesDictGen	Metodo: test_case_7
Oracolo	
ASA_rules_dict.csv contiene regole relative a vulnerabilità di classe 'pos' e 'neg' e tipo 'vulnerability'.	

Tabella 2.152: Metodo dello script test_asa_rules_dict_gen.py

Test case ID: TC_3.2_8	Test Frame: EN1 NN1 FCN1 EP1 NP1 FCP1 S1
Input	
I file RepositoryMining_ASAResults_pos.csv e RepositoryMining_ASAResults_neg.csv sono presenti, validi e non contengono record	
Output	
ASA_rules_dict.csv è vuoto.	
Test Driver	
Classe: TestMainASARulesDictGen	Metodo: test_case_8
Oracolo	
ASA_rules_dict.csv è vuoto.	

Tabella 2.153: Metodo dello script test_asa_rules_dict_gen.py**Creator CSV for ASA**

Test case ID: TC_3.3_1	Test Frame: EA2
Input	
ASA_dict.csv non è presente. ASA_rules_dict.csv è presente, valido e contiene regole.	
Output	
FileNotFoundError	
Test Driver	
Classe: TestMainCreatorCSVForASA	Metodo: test_case_1
Oracolo	
FileNotFoundError	

Tabella 2.154: Metodo dello script test_creator_csv_for_asa.py

Test case ID: TC_3.3_2	Test Frame: ER2
Input	
ASA_dict.csv è presente, valido e contiene al più un componente per regola. ASA_rules_dict.csv non è presente.	
Output	
FileNotFoundError	
Test Driver	
Classe: TestMainCreatorCSVForASA	Metodo: test_case_2
Oracolo	
FileNotFoundError	

Tabella 2.155: Metodo dello script test_creator_csv_for_asa.py

Test case ID: TC_3.3_3	Test Frame: S2
Input	
ASA_dict.csv è presente, valido e contiene al più un componente per regola. ASA_rules_dict.csv è presente e contiene regole. csv_ASA_final.csv non è accessibile	
Output	
PermissionError	
Test Driver	
Classe: TestMainCreatorCSVForASA	Metodo: test_case_3
Oracolo	
PermissionError	

Tabella 2.156: Metodo dello script test_creator_csv_for_asa.py

Test case ID: TC_3.3_4	Test Frame: EA1 NA2 FA2
Input	
ASA_dict.csv è presente ma ha un formato invalido. ASA_rules_dict.csv è presente e contiene regole.	
Output	
JSONDecodeError	
Test Driver	
Classe: TestMainCreatorCSVForASA	Metodo: test_case_4

Oracolo
JSONDecodeError

Tabella 2.157: Metodo dello script test_creator_csv_for_asa.py

Test case ID: TC_3.3_5	Test Frame: ER1 NR2 FR2
Input	
ASA_dict.csv è presente, valido e contiene al più un componente per regola. ASA_rules_dict.csv è presente ma ha un formato non valido.	
Output	
JSONDecodeError	
Test Driver	
Classe: TestMainCreatorCSVForASA	Metodo: test_case_5
Oracolo	
JSONDecodeError	

Tabella 2.158: Metodo dello script test_creator_csv_for_asa.py

Test case ID: TC_3.3_6	Test Frame: EA1 NA3 FA1 CA1 ER1 NR3 FR1 CR1 S1
Input	
ASA_dict.csv è presente e valido e contiene al più un componente per regola. ASA_rules_dict.csv è presente e valido e contiene regole con corrispondenza.	
Output	

csv_ASA_final.csv contiene ciascun componente e la frequenza per la rispettiva regola, pari a 1	
Test Driver	
Classe: TestMainCreatorCSVForASA	Metodo: test_case_6
Oracolo	
csv_ASA_final.csv contiene ciascun componente in ASA_dict.csv e la frequenza per la rispettiva regola, pari a 1	

Tabella 2.159: Metodo dello script test_creator_csv_for_asa.py

Test case ID: TC_3.3_7	Test Frame: EA1 NA2 FA1 CA2 ER1 NR2 FR1 CR1 S1
Input	
ASA_dict.csv è presente, valido e contiene più di un componente per regola. ASA_rules_dict.csv è presente e valido e contiene regole con con corrispondenza.	
Output	
csv_ASA_final.csv contiene ciascun componente e la frequenza per la rispettiva regola, maggiore di 1	
Test Driver	
Classe: TestMainCreatorCSVForASA	Metodo: test_case_7
Oracolo	
csv_ASA_final.csv contiene ciascun componente in ASA_dict.csv e la frequenza per la rispettiva regola, maggiore di 1	

Tabella 2.160: Metodo dello script test_creator_csv_for_asa.py

Test case ID: TC_3.3_8	Test Frame: EA1 NA2 FA1 CA1 ER1 NR2 FR1 CR1 S1
Input	
<p>ASA_dict.csv è presente, valido e contiene al più un componente per regola. ASA_rules_dict.csv è presente e valido, e contiene regole con con corrispondenza.</p>	
Output	
<p>csv_ASA_final.csv contiene ciascun componente in ASA_dict.csv e la frequenza per la rispettiva regola, pari a 1.</p>	
Test Driver	
Classe: TestMainCreatorCSVForASA	Metodo: test_case_8
Oracolo	
<p>csv_ASA_final.csv contiene ciascun componente in ASA_dict.csv e la frequenza per la rispettiva regola, pari a 1.</p>	

Tabella 2.161: Metodo dello script test_creator_csv_for_asa.py

Test case ID: TC_3.3_9	Test Frame: EA1 NA2 FA1 CA1 ER1 NR2 FR1 CR2 S1
Input	
<p>ASA_dict.csv è presente, valido e contiene al più un componente per regola. ASA_rules_dict.csv è presente e valido, e contiene almeno una regola senza corrispondenza.</p>	
Output	
<p>csv_ASA_final.csv contiene il componente e frequenza pari a 0 per la regola non corrispondente.</p>	

Test Driver	
Classe: TestMainCreatorCSVForASA	Metodo: test_case_9
Oracolo	
csv_ASA_final.csv contiene ciascun componente in ASA_dict.csv e frequenza pari a 0 per la regola non corrispondente.	

Tabella 2.162: Metodo dello script test_creator_csv_for_asa.py

Test case ID: TC_3.3_10	Test Frame: EA1 NA1 FA1 ER1 NR2 FR1 CR2 S1
Input	
ASA_dict.csv è presente ma non contiene record. ASA_rules_dict.csv è presente e valido e contiene regole senza corrispondenza.	
Output	
csv_ASA_final.csv contiene solo l'header con la lista di regole contenute in ASA_rules_dict.csv.	
Test Driver	
Classe: TestMainCreatorCSVForASA	Metodo: test_case_10
Oracolo	
csv_ASA_final.csv contiene solo l'header con la lista di regole contenute in ASA_rules_dict.csv.	

Tabella 2.163: Metodo dello script test_creator_csv_for_asa.py

Test case ID: TC_3.3_11	Test Frame: EA1 NA2 FA1 CA1 ER1 NR1 FR1 S1
-------------------------	---

Input	
ASA_dict.csv è presente, valido e contiene un solo record. ASA_rules_dict.csv è presente e valido, con una sola regola.	
Output	
csv_ASA_final.csv contiene l'elenco di componenti presenti in ASA_dict.csv e le relative classificazioni.	
Test Driver	
Classe: TestMainCreatorCSVForASA	Metodo: test_case_11
Oracolo	
csv_ASA_final.csv contiene l'elenco di componenti presenti in ASA_dict.csv e le relative classificazioni.	

Tabella 2.164: Metodo dello script test_creator_csv_for_asa.py

Test case ID: TC_3.3_12	Test Frame: EA1 NA1 FA1 ER1 NR1 FR1 S1
Input	
ASA_dict.csv è presente ma vuoto. ASA_rules_dict.csv è presente ma vuoto.	
Output	
csv_ASA_final.csv è vuoto.	
Test Driver	
Classe: TestMainCreatorCSVForASA	Metodo: test_case_12
Oracolo	
csv_ASA_final.csv è vuoto.	

Tabella 2.165: Metodo dello script test_creator_csv_for_asa.py**2.1.4 Union****Union**

Test case ID: TCI_4.1_1	Test Frame: L1
Input	
La stringa line è vuota	
Output	
La funzione restituisce una stringa vuota	
Test Driver	
Classe: TestGetClass	Metodo: test_case_1
Oracolo	
La funzione restituisce stringa vuota	

Tabella 2.166: Metodo dello script union.py

Test case ID: TCI_4.1_2	Test Frame: L2 C1
Input	
La stringa line non è vuota e termina con "neg" o "pos"	
Output	
La funzione restituisce l'ultimo elemento della stringa	
Test Driver	
Classe: TestGetClass	Metodo: test_case_2

Oracolo
La funzione restituisce l'ultimo elemento della stringa

Tabella 2.167: Metodo dello script union.py

Test case ID: TCI_4.1_3	Test Frame: L2 C2
Input	
La stringa line non è vuota e non termina con "neg" o "pos"	
Output	
La funzione restituisce l'ultimo elemento della stringa	
Test Driver	
Classe: TestGetClass	Metodo: test_case_3
Oracolo	
La funzione restituisce l'ultimo elemento della stringa	

Tabella 2.168: Metodo dello script union.py

Test case ID: TCI_4.2_1	Test Frame: SM1 TM2 CE1
Input	
line_sm è None line_tm è vuota class_element è None	
Output	
ValueError	

Test Driver	
Classe: TestAnotherOption	Metodo: test_case_1
Oracolo	
ValueError	

Tabella 2.169: Metodo dello script union.py

Test case ID: TCI_4.2_2	Test Frame: SM1 TM3 CE1
Input	
line_sm è None line_tm non è vuota class_element è None	
Output	
ValueError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_2
Oracolo	
ValueError	

Tabella 2.170: Metodo dello script union.py

Test case ID: TCI_4.2_3	Test Frame: SM2 TM1
Input	

line_sm non è vuota e contiene meno di 3 elementi line_tm è vuota class_element è None	
Output	
La funzione restituisce stringa vuota	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_3
Oracolo	
La funzione restituisce stringa vuota	

Tabella 2.171: Metodo dello script union.py

Test case ID: TCI_4.2_4	Test Frame: SM3 TM1
Input	
line_sm non è vuota e contiene più di 3 elementi line_tm è vuota class_element è None	
Output	
La funzione restituisce line_sm con la rimozione dei primi due elementi e delle virgole che li seguono, aggiungendo una virgola finale	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_4
Oracolo	

La funzione restituisce line_sm con la rimozione dei primi due elementi e delle virgole che li seguono, aggiungendo una virgola finale

Tabella 2.172: Metodo dello script union.py

Test case ID: TCI_4.2_5	Test Frame: SM1 TM3 CE2
Input	
line_sm è None line_tm non è vuota class_element è vuoto	
Output	
ValueError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_5
Oracolo	
ValueError	

Tabella 2.173: Metodo dello script union.py

Test case ID: TCI_4.2_6	Test Frame: SM1 TM3 CE3 TMC1
Input	
line_sm è None line_tm non è vuota class_element è non vuoto e non conenuto in line_tm	
Output	

ValueError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_6
Oracolo	
ValueError	

Tabella 2.174: Metodo dello script union.py

Test case ID: TCI_4.2_7	Test Frame: SM1 TM3 CE3 TMC2
Input	
line_sm è None line_tm non è vuota class_element è non vuoto e contenuto in line_tm	
Output	
La funzione restituisce line_tm con la rimozione della stringa class_element	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_7
Oracolo	
La funzione restituisce line_tm con la rimozione della stringa class_element	

Tabella 2.175: Metodo dello script union.py

Test case ID: TCI_4.3_1	Test Frame: CM1
Input	

name_csv_mining è vuota	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_1
Oracolo	
FileNotFoundError	

Tabella 2.176: Metodo dello script union.py

Test case ID: TCI_4.3_2	Test Frame: CM2
Input	
name_csv_mining è diverso da "csv_mining_final.csv"	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_2
Oracolo	
FileNotFoundError	

Tabella 2.177: Metodo dello script union.py

Test case ID: TCI_4.3_3	Test Frame: CM3 CSM1
Input	

name_csv_soft_m è vuota	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_3
Oracolo	
FileNotFoundError	

Tabella 2.178: Metodo dello script union.py

Test case ID: TCI_4.3_4	Test Frame: CM3 CSM2
Input	
name_csv_soft_m è diversa da mining_results_sm_final.csv	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_4
Oracolo	
FileNotFoundError	

Tabella 2.179: Metodo dello script union.py

Test case ID: TCI_4.3_5	Test Frame: CM3 CSM3 AU1
Input	

Non ci sono permessi in scrittura nel file di unione	
Output	
PermissionError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_5
Oracolo	
PermissionError	

Tabella 2.180: Metodo dello script union.py

Test case ID: TCI_4.3_6	Test Frame: CM3 CSM3 AU2 EMF1
Input	
Il file di Unione non esiste	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_6
Oracolo	
FileNotFoundError	

Tabella 2.181: Metodo dello script union.py

Test case ID: TCI_4.3_7	Test Frame: CM3 CSM3 AU2 EMF2 CMF1 ESMF2 CSMF1
--------------------------------	---

Input	
csv_mining_final è vuoto mining_results_sm_final è vuoto	
Output	
Il file di Unione risulterà vuoto	
Test Driver	
Classe: TestInitialize	Metodo: test_case_7
Oracolo	
Il file di Unione risulterà vuoto	

Tabella 2.182: Metodo dello script union.py

Test case ID: TCI_4.3_8	Test Frame: CM3 CSM3 AU2 EMF2 CMF2 ESMF2 CSMF2
Input	
csv_mining_final è costituito solo dall'header mining_results_sm_final è costituito solo dall'header	
Output	
Il file di Unione conterrà la combinazione delle due intestazioni dei file	
Test Driver	
Classe: TestInitialize	Metodo: test_case_8
Oracolo	
Il file di Unione conterrà la combinazione delle due intestazioni dei file	

Tabella 2.183: Metodo dello script union.py

Test case ID: TCI_4.3_9	Test Frame: CM3 CSM3 AU2 EMF2 CMF3 ESMF1
Input	
Il file di Unione non esiste	
Output	
FileNotFoundException	
Test Driver	
Classe: TestInitialize	Metodo: test_case_9
Oracolo	
FileNotFoundException	

Tabella 2.184: Metodo dello script union.py

Test case ID: TCI_4.3_10	Test Frame: CM3 CSM3 AU2 EMF2 CMF3 ESMF2 CSMF1
Input	
csv_mining_final è costituito da header e da dati mining_results_sm_final è costituito solo dall'header	
Output	
La funzione combina le intestazioni dei due file nel file di Unione	
Test Driver	
Classe: TestInitialize	Metodo: test_case_10
Oracolo	

La funzione combina le intestazioni dei due file nel file di Unione

Tabella 2.185: Metodo dello script union.py

Test case ID: TCI_4.3_11	Test Frame: CM3 CSM3 AU2 EMF2 CMF3 ESMF2 CSMF2
Input	
csv_mining_final è costituito da header mining_results_sm_final è costituito solo dall'header e da dati	
Output	
La funzione combina le intestazioni dei due file nel file di Unione	
Test Driver	
Classe: TestInitialize	Metodo: test_case_11
Oracolo	
La funzione combina le intestazioni dei due file nel file di Unione	

Tabella 2.186: Metodo dello script union.py

Test case ID: TCI_4.3_12	Test Frame: CM3 CSM3 AU2 EMF2 CMF3 ESMF2 CSMF3
Input	
csv_mining_final è costituito da header e da dati mining_results_sm_final è costituito solo dall'header e da dati	
Output	
La funzione combina le intestazioni dei due file nel file di Unione	

Test Driver	
Classe: TestInitialize	Metodo: test_case_12
Oracolo	
La funzione combina le intestazioni e i dati dei due file nel file di Unione, combinando i dati che hanno lo stesso nome nella prima colonna	

Tabella 2.187: Metodo dello script union.py**Union_TMwithASA**

Test case ID: TCI_4.1_1	Test Frame: L1
Input	
La stringa line è vuota	
Output	
La funzione restituisce una stringa vuota	
Test Driver	
Classe: TestGetClass	Metodo: test_case_1
Oracolo	
La funzione restituisce stringa vuota	

Tabella 2.188: Metodo dello script union_TMwithASA.py

Test case ID: TCI_4.1_2	Test Frame: L2 C1
Input	
La stringa line non è vuota e termina con "neg" o "pos"	

Output	
La funzione restituisce l'ultimo elemento della stringa	
Test Driver	
Classe: TestGetClass	Metodo: test_case_2
Oracolo	
La funzione restituisce l'ultimo elemento della stringa	

Tabella 2.189: Metodo dello script union_TMwithASA.py

Test case ID: TCI_4.1_3	Test Frame: L2 C2
Input	
La stringa line non è vuota e non termina con "neg" o "pos"	
Output	
La funzione restituisce l'ultimo elemento della stringa	
Test Driver	
Classe: TestGetClass	Metodo: test_case_3
Oracolo	
La funzione restituisce l'ultimo elemento della stringa	

Tabella 2.190: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.1_1	Test Frame: CE1
Input	

line_asa è None line_tm è None class_element è None	
Output	
AttributeError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_1
Oracolo	
AttributeError	

Tabella 2.191: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.1_2	Test Frame: CE2 ASA2 TM2
Input	
line_asa è vuota line_tm è vuota class_element è vuota	
Output	
La funzione restituisce None	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_2
Oracolo	
La funzione restituisce None	

Tabella 2.192: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.1_3	Test Frame: CE2 ASA1 TM1
Input	
line_asa è None line_tm è None class_element è vuota	
Output	
AttributeError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_3
Oracolo	
AttributeError	

Tabella 2.193: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.1_4	Test Frame: CE3 ASA1 TM3 TMC1
Input	
line_asa è None line_tm esiste e non è vuota class_element esiste e non è vuoto ma non è contenuto in line_tm	
Output	
ValueError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_4
Oracolo	

ValueError

Tabella 2.194: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.1_5	Test Frame: CE3 ASA1 TM3 TMC2
Input	
line_asa è None line_tm esiste e non è vuota class_element esiste e non è vuoto ma è contenuto in line_tm	
Output	
La funzione restituisce line_tm rimuovendo class element	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_5
Oracolo	
La funzione restituisce line_tm rimuovendo class element	

Tabella 2.195: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.1_6	Test Frame: CE3 ASA3 TM1 ASAC1
Input	
line_asa esiste e non è vuota line_tm è None class_element esiste e non è vuoto ma è contenuto in line_tm	
Output	
ValueError	

Test Driver	
Classe: TestAnotherOption	Metodo: test_case_6
Oracolo	
ValueError	

Tabella 2.196: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.1_7	Test Frame: CE3 ASA3 TM1 ASAC1
Input	
line_asa esiste e non è vuota line_tm è None class_element esiste e non è vuoto ma non è contenuto in line_tm	
Output	
La funzione restituisce line_asa rimuovendo class element e il primo elemento	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_7
Oracolo	
La funzione restituisce line_asa rimuovendo class element e il primo elemento	

Tabella 2.197: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.1_8	Test Frame: CE3 ASA3 ASAC2 TM3 TMC2
Input	

line_asa esiste e non è vuota line_tm esiste e non è vuota class_element esiste e non è vuoto ed è contenuto in line_tm e line_asa	
Output	
La funzione restituisce None	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_8
Oracolo	
La funzione restituisce None	

Tabella 2.198: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.2_1	Test Frame: CM1
Input	
name_csv_mining è vuota	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_1
Oracolo	
FileNotFoundError	

Tabella 2.199: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.2_2	Test Frame: CM2
Input	
name_csv_mining è diverso da "csv_mining_final.csv"	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_2
Oracolo	
FileNotFoundError	

Tabella 2.200: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.2_3	Test Frame: CM3 CASA1
Input	
name_csv_soft_m è vuota	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_3
Oracolo	
FileNotFoundError	

Tabella 2.201: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.2_4	Test Frame: CM3 CASA2
Input	
name_csv_soft_m è diversa da csv_ASA_final.csv	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_4
Oracolo	
FileNotFoundError	

Tabella 2.202: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.2_5	Test Frame: CM3 CASA3 AU1
Input	
Non ci sono permessi in scrittura nel file di unione	
Output	
PermissionError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_5
Oracolo	
PermissionError	

Tabella 2.203: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.2_6	Test Frame: CM3 CASA3 AU2 EMF1
Input	
Il file di Unione non esiste	
Output	
FileNotFoundException	
Test Driver	
Classe: TestInitialize	Metodo: test_case_6
Oracolo	
FileNotFoundException	

Tabella 2.204: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.2_7	Test Frame: CM3 CASA3 AU2 EMF2 CMF1 EASAF2 CASAF1
Input	
csv_mining_final è vuoto csv_ASA_final è vuoto	
Output	
Il file di Unione risulterà vuoto	
Test Driver	
Classe: TestInitialize	Metodo: test_case_7
Oracolo	
Il file di Unione risulterà vuoto	

Tabella 2.205: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.2_8	Test Frame: CM3 CASA3 AU2 EMF2 CMF2 EASAF2 CASAF2
Input	
entrambi i file sono costituiti solo dall'header	
Output	
Il file di Unione conterrà la combinazione delle due intestazioni dei file	
Test Driver	
Classe: TestInitialize	Metodo: test_case_8
Oracolo	
Il file di Unione conterrà la combinazione delle due intestazioni dei file	

Tabella 2.206: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.2_9	Test Frame: CM3 CASA3 AU2 EMF2 CMF3 EASAF1
Input	
Il file di Unione non esiste	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_9

Oracolo
FileNotFoundException

Tabella 2.207: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.2_10	Test Frame: CM3 CASA3 AU2 EMF2 CMF3 EASAF2 CASAF1
Input	
csv_mining_final è costituito da header e da dati csv_ASA_final è costituito solo dall'header	
Output	
La funzione scrive nel file di Unione la combinazione delle intestazione dei due file ed i dati del file inerente alle software metrics aggiungendo valori "0" per rimpiazzare i dati asa mancanti	
Test Driver	
Classe: TestInitialize	Metodo: test_case_10
Oracolo	
La funzione scrive nel file di Unione la combinazione delle intestazione dei due file ed i dati del file inerente alle software metrics aggiungendo valori "0" per rimpiazzare i dati asa mancanti	

Tabella 2.208: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.2_11	Test Frame: CM3 CASA3 AU2 EMF2 CMF3 EASAF2 CASAF2
Input	

<p>csv_mining_final è costituito da header csv_ASA_final è costituito solo dall'header e da dati</p>	
Output	
<p>La funzione combina le intestazioni dei due file nel file di Unione e rimpiazza i dati mancanti con '0'</p>	
Test Driver	
Classe: TestInitialize	Metodo: test_case_11
Oracolo	
<p>La funzione combina le intestazioni dei due file nel file di Unione e rimpiazza i dati mancanti con '0'</p>	

Tabella 2.209: Metodo dello script union_TMwithASA.py

Test case ID: TCI_5.2_12	Test Frame: CM3 CASA3 AU2 EMF2 CMF3 EASAF2 CASAF3
Input	
<p>csv_mining_final è costituito da header e da dati csv_ASA_final è costituito dall'header e da dati</p>	
Output	
<p>La funzione combina le intestazioni e i dati dei due file nel file di Unione, combinando i dati che hanno lo stesso nome nella prima colonna. Laddove ci siano dati mancanti per quanto riguarda il csv asa, aggiunge dei valori '0'</p>	
Test Driver	
Classe: TestInitialize	Metodo: test_case_12
Oracolo	

La funzione combina le intestazioni e i dati dei due file nel file di Unione, combinando i dati che hanno lo stesso nome nella prima colonna. Laddove ci siano dati mancanti per quanto riguarda il csv asa, aggiunge dei valori '0'

Tabella 2.210: Metodo dello script union_TMwithASA.py

Union_SMwithASA

Test case ID: TCI_4.1_1	Test Frame: L1
Input	
La stringa line è vuota	
Output	
La funzione restituisce una stringa vuota	
Test Driver	
Classe: TestGetClass	Metodo: test_case_1
Oracolo	
La funzione restituisce stringa vuota	

Tabella 2.211: Metodo dello script union_SMwithASA.py

Test case ID: TCI_4.1_2	Test Frame: L2 C1
Input	
La stringa line non è vuota e termina con "neg" o "pos"	
Output	
La funzione restituisce l'ultimo elemento della stringa	

Test Driver	
Classe: TestGetClass	Metodo: test_case_2
Oracolo	
La funzione restituisce l'ultimo elemento della stringa	

Tabella 2.212: Metodo dello script union_SMwithASA.py

Test case ID: TCI_4.1_3	Test Frame: L2 C2
Input	
La stringa line non è vuota e non termina con "neg" o "pos"	
Output	
La funzione restituisce l'ultimo elemento della stringa	
Test Driver	
Classe: TestGetClass	Metodo: test_case_3
Oracolo	
La funzione restituisce l'ultimo elemento della stringa	

Tabella 2.213: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.1_1	Test Frame: CE1
Input	
line_asa è None line_sm è None class_element è None	

Output	
AttributeError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_1
Oracolo	
AttributeError	

Tabella 2.214: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.1_2	Test Frame: CE2 ASA2 TM2
Input	
line_asa è vuota line_sm è vuota class_element è vuota	
Output	
La funzione restituisce None	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_2
Oracolo	
La funzione restituisce None	

Tabella 2.215: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.1_3	Test Frame: CE2 ASA1 TM1
-------------------------	--------------------------

Input	
line_asa è None line_sm è None class_element è vuota	
Output	
AttributeError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_3
Oracolo	
AttributeError	

Tabella 2.216: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.1_4	Test Frame: CE3 ASA1 TM3 TMC1
Input	
line_asa è None line_sm esiste e non è vuota class_element esiste e non è vuoto ma non è contenuto in line_tm	
Output	
ValueError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_4
Oracolo	
ValueError	

Tabella 2.217: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.1_5	Test Frame: CE3 ASA1 TM3 TMC2
Input	
line_asa è None line_sm esiste e non è vuota class_element esiste e non è vuoto ma è contenuto in line_tm	
Output	
La funzione restituisce line_tm rimuovendo class element	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_5
Oracolo	
La funzione restituisce line_tm rimuovendo class element	

Tabella 2.218: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.1_6	Test Frame: CE3 ASA3 TM1 ASAC1
Input	
line_asa esiste e non è vuota line_sm è None class_element esiste e non è vuoto ma è contenuto in line_tm	
Output	
ValueError	
Test Driver	

Classe: TestAnotherOption	Metodo: test_case_6
Oracolo	
ValueError	

Tabella 2.219: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.1_7	Test Frame: CE3 ASA3 TM1 ASAC1
Input	
line_asa esiste e non è vuota line_sm è None class_element esiste e non è vuoto ma non è contenuto in line_tm	
Output	
La funzione restituisce line_asa rimuovendo class element e il primo elemento	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_7
Oracolo	
La funzione restituisce line_asa rimuovendo class element e il primo elemento	

Tabella 2.220: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.1_8	Test Frame: CE3 ASA3 ASAC2 TM3 TMC2
Input	
line_asa esiste e non è vuota line_sm esiste e non è vuota class_element esiste e non è vuoto ed è contenuto in line_sm e line_asa	

Output	
La funzione restituisce None	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_8
Oracolo	
La funzione restituisce None	

Tabella 2.221: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2_1	Test Frame: CM1
Input	
name_csv_mining è vuota	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_1
Oracolo	
FileNotFoundError	

Tabella 2.222: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2_2	Test Frame: CM2
Input	
name_csv_mining è diverso da "mining_results_sm_final.csv"	

Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_2
Oracolo	
FileNotFoundError	

Tabella 2.223: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2_3	Test Frame: CM3 CASA1
Input	
name_csv_soft_m è vuota	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_3
Oracolo	
FileNotFoundError	

Tabella 2.224: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2_4	Test Frame: CM3 CASA2
Input	
name_csv_soft_m è diversa da csv_ASA_final.csv	

Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_4
Oracolo	
FileNotFoundError	

Tabella 2.225: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2_5	Test Frame: CM3 CASA3 AU1
Input	
Non ci sono permessi in scrittura nel file di unione	
Output	
PermissionError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_5
Oracolo	
PermissionError	

Tabella 2.226: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2_6	Test Frame: CM3 CASA3 AU2 EMF1
Input	
Il file di Unione non esiste	

Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_6
Oracolo	
FileNotFoundError	

Tabella 2.227: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2_7	Test Frame: CM3 CASA3 AU2 EMF2 CMF1 EASAF2 CASAF1
Input	
csv_mining_final è vuoto csv_ASA_final è vuoto	
Output	
Il file di Unione risulterà vuoto	
Test Driver	
Classe: TestInitialize	Metodo: test_case_7
Oracolo	
Il file di Unione risulterà vuoto	

Tabella 2.228: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2_8	Test Frame: CM3 CASA3 AU2 EMF2 CMF2 EASAF2 CASAF2
-------------------------	--

Input	
entrambi i file sono costituiti solo dall'header	
Output	
Il file di Unione conterrà la combinazione delle due intestazioni dei file	
Test Driver	
Classe: TestInitialize	Metodo: test_case_8
Oracolo	
Il file di Unione conterrà la combinazione delle due intestazioni dei file	

Tabella 2.229: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2_9	Test Frame: CM3 CASA3 AU2 EMF2 CMF3 EASAF1
Input	
Il file di Unione non esiste	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_9
Oracolo	
FileNotFoundError	

Tabella 2.230: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2_10	Test Frame: CM3 CASA3 AU2 EMF2 CMF3 EASAF2 CASAF1
Input	
csv_mining_final è costituito da header e da dati csv_ASA_final è costituito solo dall'header	
Output	
La funzione scrive nel file di Unione la combinazione delle intestazione dei due file ed i dati del file inerente alle software metrics aggiungendo valori "0" per rimpiazzare i dati asa mancanti	
Test Driver	
Classe: TestInitialize	Metodo: test_case_10
Oracolo	
La funzione scrive nel file di Unione la combinazione delle intestazione dei due file ed i dati del file inerente alle software metrics aggiungendo valori "0" per rimpiazzare i dati asa mancanti	

Tabella 2.231: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2_11	Test Frame: CM3 CASA3 AU2 EMF2 CMF3 EASAF2 CASAF2
Input	
csv_mining_final è costituito da header csv_ASA_final è costituito solo dall'header e da dati	
Output	
La funzione combina le intestazioni dei due file nel file di Unione e rimpiazza i dati mancanti con '0'	

Test Driver	
Classe: TestInitialize	Metodo: test_case_11
Oracolo	
La funzione combina le intestazioni dei due file nel file di Unione e rimpiazza i dati mancanti con '0'	

Tabella 2.232: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2_12	Test Frame: CM3 CASA3 AU2 EMF2 CMF3 EASAF2 CASAF3
Input	
csv_mining_final è costituito da header e da dati csv_ASA_final è costituito dall'header e da dati	
Output	
La funzione combina le intestazioni e i dati dei due file nel file di Unione, combinando i dati che hanno lo stesso nome nella prima colonna. Laddove ci siano dati mancanti per quanto riguarda il csv asa, aggiunge dei valori '0'	
Test Driver	
Classe: TestInitialize	Metodo: test_case_12
Oracolo	
La funzione combina le intestazioni e i dati dei due file nel file di Unione, combinando i dati che hanno lo stesso nome nella prima colonna. Laddove ci siano dati mancanti per quanto riguarda il csv asa, aggiunge dei valori '0'	

Tabella 2.233: Metodo dello script union_SMwithASA.py**TotalCombination**

Test case ID: TCI_4.1_1	Test Frame: L1
Input	
La stringa line è vuota	
Output	
La funzione restituisce una stringa vuota	
Test Driver	
Classe: TestGetClass	Metodo: test_case_1
Oracolo	
La funzione restituisce stringa vuota	

Tabella 2.234: Metodo dello script TotalCombination.py

Test case ID: TCI_4.1_2	Test Frame: L2 C1
Input	
La stringa line non è vuota e termina con "neg" o "pos"	
Output	
La funzione restituisce l'ultimo elemento della stringa	
Test Driver	
Classe: TestGetClass	Metodo: test_case_2
Oracolo	
La funzione restituisce l'ultimo elemento della stringa	

Tabella 2.235: Metodo dello script TotalCombination.py

Test case ID: TCI_4.1_3	Test Frame: L2 C2
Input	
La stringa line non è vuota e non termina con "neg" o "pos"	
Output	
La funzione restituisce l'ultimo elemento della stringa	
Test Driver	
Classe: TestGetClass	Metodo: test_case_3
Oracolo	
La funzione restituisce l'ultimo elemento della stringa	

Tabella 2.236: Metodo dello script TotalCombination.py

Test case ID: TCI_4.2_1	Test Frame: SM1 TM2 CE1
Input	
line_sm è None line_tm è vuota class_element è None	
Output	
ValueError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_1
Oracolo	
ValueError	

Tabella 2.237: Metodo dello script TotalCombination.py

Test case ID: TCI_4.2_2	Test Frame: SM1 TM3 CE1
Input	
line_sm è None line_tm non è vuota class_element è None	
Output	
ValueError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_2
Oracolo	
ValueError	

Tabella 2.238: Metodo dello script TotalCombination.py

Test case ID: TCI_4.2_3	Test Frame: SM2 TM1
Input	
line_sm non è vuota e contiene meno di 3 elementi line_tm è vuota class_element è None	
Output	
La funzione restituisce stringa vuota	
Test Driver	

Classe: TestAnotherOption	Metodo: test_case_3
Oracolo	
La funzione restituisce stringa vuota	

Tabella 2.239: Metodo dello script TotalCombination.py

Test case ID: TCI_4.2_4	Test Frame: SM3 TM1
Input	
line_sm non è vuota e contiene più di 3 elementi line_tm è vuota class_element è None	
Output	
La funzione restituisce line_sm con la rimozione dei primi due elementi e delle virgole che li seguono, aggiungendo una virgola finale	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_4
Oracolo	
La funzione restituisce line_sm con la rimozione dei primi due elementi e delle virgole che li seguono, aggiungendo una virgola finale	

Tabella 2.240: Metodo dello script TotalCombination.py

Test case ID: TCI_4.2_5	Test Frame: SM1 TM3 CE2
Input	

line_sm è None line_tm non è vuota class_element è vuoto	
Output	
ValueError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_5
Oracolo	
ValueError	

Tabella 2.241: Metodo dello script TotalCombination.py

Test case ID: TCI_4.2_6	Test Frame: SM1 TM3 CE3 TMC1
Input	
line_sm è None line_tm non è vuota class_element è non vuoto e non conenuto in line_tm	
Output	
ValueError	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_6
Oracolo	
ValueError	

Tabella 2.242: Metodo dello script TotalCombination.py

Test case ID: TCI_4.2_7	Test Frame: SM1 TM3 CE3 TMC2
Input	
line_sm è None line_tm non è vuota class_element è non vuoto e contenuto in line_tm	
Output	
La funzione restituisce line_tm con la rimozione della stringa class_element	
Test Driver	
Classe: TestAnotherOption	Metodo: test_case_7
Oracolo	
La funzione restituisce line_tm con la rimozione della stringa class_element	

Tabella 2.243: Metodo dello script TotalCombination.py

Test case ID: TCI_4.3_1	Test Frame: CM1
Input	
name_csv_mining è vuota	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_1
Oracolo	
FileNotFoundError	

Tabella 2.244: Metodo dello script TotalCombination.py

Test case ID: TCI_4.3_2	Test Frame: CM2
Input	
name_csv_mining è diverso da "Union_TM_ASA.csv"	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_2
Oracolo	
FileNotFoundError	

Tabella 2.245: Metodo dello script TotalCombination.py

Test case ID: TCI_4.3_3	Test Frame: CM3 CSM1
Input	
name_csv_soft_m è vuota	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_3
Oracolo	
FileNotFoundError	

Tabella 2.246: Metodo dello script TotalCombination.py

Test case ID: TCI_4.3_4	Test Frame: CM3 CSM2
Input	
name_csv_soft_m è diversa da mining_results_sm_final.csv	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_4
Oracolo	
FileNotFoundError	

Tabella 2.247: Metodo dello script TotalCombination.py

Test case ID: TCI_4.3_5	Test Frame: CM3 CSM3 AU1
Input	
Non ci sono permessi in scrittura nel file di unione	
Output	
PermissionError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_5
Oracolo	
PermissionError	

Tabella 2.248: Metodo dello script TotalCombination.py

Test case ID: TCI_4.3_6	Test Frame: CM3 CSM3 AU2 EMF1
Input	
Il file di Unione non esiste	
Output	
FileNotFoundError	
Test Driver	
Classe: TestInitialize	Metodo: test_case_6
Oracolo	
FileNotFoundError	

Tabella 2.249: Metodo dello script TotalCombination.py

Test case ID: TCI_4.3_7	Test Frame: CM3 CSM3 AU2 EMF2 CMF1 ESMF2 CSMF1
Input	
csv_mining_final è vuoto mining_results_sm_final è vuoto	
Output	
Il file di Unione risulterà vuoto	
Test Driver	
Classe: TestInitialize	Metodo: test_case_7

Oracolo
Il file di Unione risulterà vuoto

Tabella 2.250: Metodo dello script TotalCombination.py

Test case ID: TCI_4.3_8	Test Frame: CM3 CSM3 AU2 EMF2 CMF2 ESMF2 CSMF2
Input	
csv_mining è costituito solo dall'header csv_software_metrics è costituito solo dall'header	
Output	
Il file di Unione conterrà la combinazione delle due intestazioni dei file	
Test Driver	
Classe: TestInitialize	Metodo: test_case_8
Oracolo	
Il file di Unione conterrà la combinazione delle due intestazioni dei file	

Tabella 2.251: Metodo dello script TotalCombination.py

Test case ID: TCI_4.3_9	Test Frame: CM3 CSM3 AU2 EMF2 CMF3 ESMF1
Input	
Il file di Unione non esiste	
Output	
FileNotFoundError	

Test Driver	
Classe: TestInitialize	Metodo: test_case_9
Oracolo	
FileNotFoundException	

Tabella 2.252: Metodo dello script TotalCombination.py

Test case ID: TCI_4.3_10	Test Frame: CM3 CSM3 AU2 EMF2 CMF3 ESMF2 CSMF1
Input	
csv_mining è costituito da header e da dati csv_software_metrics è costituito solo dall'header	
Output	
La funzione combina le intestazioni dei due file nel file di Unione	
Test Driver	
Classe: TestInitialize	Metodo: test_case_10
Oracolo	
La funzione combina le intestazioni dei due file nel file di Unione	

Tabella 2.253: Metodo dello script TotalCombination.py

Test case ID: TCI_4.3_11	Test Frame: CM3 CSM3 AU2 EMF2 CMF3 ESMF2 CSMF2
Input	

csv_mining è costituito da header csv_software_metrics è costituito solo dall'header e da dati	
Output	
La funzione combina le intestazioni dei due file nel file di Unione	
Test Driver	
Classe: TestInitialize	Metodo: test_case_11
Oracolo	
La funzione combina le intestazioni dei due file nel file di Unione	

Tabella 2.254: Metodo dello script TotalCombination.py

Test case ID: TCI_4.3_12	Test Frame: CM3 CSM3 AU2 EMF2 CMF3 ESMF2 CSMF3
Input	
csv_mining è costituito da header e da dati csv_software_metrics è costituito solo dall'header e da dati	
Output	
La funzione combina le intestazioni dei due file nel file di Unione	
Test Driver	
Classe: TestInitialize	Metodo: test_case_12
Oracolo	
La funzione combina le intestazioni e i dati dei due file nel file di Unione, combinando i dati che hanno lo stesso nome nella prima colonna	

Tabella 2.255: Metodo dello script TotalCombination.py

2.2 Test Case Specification di integrazione

2.2.1 Repo Mining

Test case ID: TCI_1.1	Test Frame: EI2
Input	
<p><code>initial_dataset.csv</code> e la directory <code>Dataset_Divided</code> non esistono. La directory <code>mining_results</code> esiste.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
FileNotFoundError	
Test Driver	
Classe: <code>TestRepoMiningIntegration</code>	Metodo: <code>test_case_1</code>
Oracolo	
FileNotFoundError	

Tabella 2.256: Metodo dello script `test_repo_mining_integration.py`

Test case ID: TCI_1.2	Test Frame: EM2
Input	
<p><code>initial_dataset.csv</code> ha un formato valido, ma nessun record.</p> <p>La directory <code>Dataset_Divided</code> esiste.</p> <p>La directory <code>mining_results</code> non esiste.</p>	
Output	
FileNotFoundError	
Test Driver	

Classe: TestRepoMiningIntegration	Metodo: test_case_2
Oracolo	
FileNotFoundError	

Tabella 2.257: Metodo dello script test_repo_mining_integration.py

Test case ID: TCI_1.3	Test Frame: EI1 FD2
Input	
<p>initial_dataset.csv esiste ma ha un formato non valido.</p> <p>La directory Dataset_Divided esiste.</p> <p>La directory mining_results non esiste.</p>	
Output	
KeyError	
Test Driver	
Classe: TestRepoMiningIntegration	Metodo: test_case_3
Oracolo	
KeyError	

Tabella 2.258: Metodo dello script test_repo_mining_integration.py

Test case ID: TCI_4	Test Frame: EI1 FI1 N1 ED1 EM1 EC2 EE2
Input	
<p>initial_dataset.csv ha un formato valido, ma nessun record.</p> <p>Le directory Dataset_Divided e mining_results esistono.</p> <p>CHECK.txt e ERRORS.txt non esistono.</p>	

Output	
FileNotFoundError	
Test Driver	
Classe: TestRepoMiningIntegration	Metodo: test_case_4
Oracolo	
Dataset_divided contiene 1.csv e mining_results è vuota	

Tabella 2.259: Metodo dello script test_repo_mining_integration.py

Test case ID: TCI_5	Test Frame: EI1 FI1 N2 CL3
Input	
<p>initial_dataset.csv ha un formato valido, ha 50 record e contiene link non valido.</p> <p>La directory Dataset_Divided non esiste.</p> <p>La directory mining_results esiste.</p> <p>CHECK.txt e ERRORS.txt esistono.</p>	
Output	
Dataset_Divided creata e Missing Schema	
Test Driver	
Classe: TestRepoMiningIntegration	Metodo: test_case_5
Oracolo	
Dataset_Divided creata e Missing Schema	

Tabella 2.260: Metodo dello script test_repo_mining_integration.py

Test case ID: TCI_6	Test Frame: EI1 FI1 N2 CL2
---------------------	----------------------------

Input	
<p><code>initial_dataset.csv</code> ha un formato valido, ha 50 record e contiene link non esistente.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> non esistono.</p>	
Output	
Connection Error	
Test Driver	
Classe: <code>TestRepoMiningIntegration</code>	Metodo: <code>test_case_6</code>
Oracolo	
Connection Error	

Tabella 2.261: Metodo dello script `test_repo_mining_integration.py`

Test case ID: TCI_1.7	Test Frame: EI1 FI1 N3 CL1 CR2 ED1 EM1 EC2 EE2
Input	
<p><code>initial_dataset.csv</code> ha un formato valido, ha 150 record e contiene link esistente con repo non valida.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> non esistono.</p>	
Output	
FileNotFoundError	
Test Driver	
Classe: <code>TestRepoMiningIntegration</code>	Metodo: <code>test_case_7</code>

Oracolo
CHECKS.txt indica che la repo non è disponibile.

Tabella 2.262: Metodo dello script test_repo_mining_integration.py

Test case ID: TCI_1.8	Test Frame: EI1 FI1 N2 CL1 CR1 CC3 ED1 EM1 EC2 EE2
Input	
<p>initial_dataset.csv ha un formato valido, ha 50 record e contiene link esistente con repo valida ma commit non valido senza risposta.</p> <p>Le directory Dataset_Divided e mining_results esistono.</p> <p>CHECK.txt e ERRORS.txt non esistono.</p>	
Output	
CHECKS.txt indica che il commit non è esistente.	
Test Driver	
Classe: TestRepoMiningIntegration	Metodo: test_case_8
Oracolo	
CHECKS.txt indica che il commit non è esistente.	

Tabella 2.263: Metodo dello script test_repo_mining_integration.py

Test case ID: TCI_1.9	Test Frame: EI1 FI1 N2 CL1 CR1 CC4 ED1 EM1 EC1 EE1
Input	

<p><code>initial_dataset.csv</code> ha un formato valido, ha 50 record e contiene link esistente con repo valida ma commit non valido senza risposta.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> esistono.</p>	
Output	
<code>CHECKS.txt</code> e <code>ERRORS.txt</code> indicano che il commit non è definito.	
Test Driver	
Classe: <code>TestRepoMiningIntegration</code>	Metodo: <code>test_case_9</code>
Oracolo	
<code>CHECKS.txt</code> e <code>ERRORS.txt</code> indicano che il commit non è definito.	

Tabella 2.264: Metodo dello script `test_repo_mining_integration.py`

Test case ID: TCI_1.10	Test Frame: EI1 FI1 N2 CL1 CR1 CC2 ED1 EM1 EC2 EE2
Input	
<p><code>initial_dataset.csv</code> ha un formato valido, ha 50 record e contiene link esistente con repo valida ma commit valido con nome file eccedente limite Windows.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> non esistono.</p>	
Output	
GitCommandError	
Test Driver	
Classe: <code>TestRepoMiningIntegration</code>	Metodo: <code>test_case_10</code>
Oracolo	

GitCommandError

Tabella 2.265: Metodo dello script test_repo_mining_integration.py

Test case ID: TCI_1.11	Test Frame: EI1 FI1 N2 CL1 CR1 CC1 CM1 ED1 EM1 EC2 EE2
Input	
<p>initial_dataset.csv ha un formato valido, ha 50 record e contiene link esistente con repo valida, con commit senza modifiche.</p> <p>Le directory Dataset_Divided e mining_results esistono.</p> <p>CHECK.txt e ERRORS.txt non esistono.</p>	
Output	
CHECKS.txt indica status ok.	
Test Driver	
Classe: TestRepoMiningIntegration	Metodo: test_case_11
Oracolo	
CHECKS.txt indica status ok.	

Tabella 2.266: Metodo dello script test_repo_mining_integration.py

Test case ID: TCI_1.12	Test Frame: EI1 FI1 N2 CL1 CR1 CC1 CM2 ED1 EM1 EC2 EE2
Input	

<p><code>initial_dataset.csv</code> ha un formato valido, ha 50 record e contiene link esistente con repo valida, con commit senza modifiche a classi <code>.java</code>.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> non esistono.</p>	
Output	
<code>CHECKS.txt</code> indica status ok.	
Test Driver	
Classe: <code>TestRepoMiningIntegration</code>	Metodo: <code>test_case_12</code>
Oracolo	
<code>CHECKS.txt</code> indica status ok.	

Tabella 2.267: Metodo dello script `test_repo_mining_integration.py`

Test case ID: TCI_1.13	Test Frame: EI1 FI1 N2 CL1 CR1 CC1 CM3 ED1 EM1 EC2 EE2
Input	
<p><code>initial_dataset.csv</code> ha un formato valido, ha 50 record e contiene link esistente con repo valida, con commit con modiche solo a classi <code>.java</code> non presenti prima.</p> <p>Le directory <code>Dataset_Divided</code> e <code>mining_results</code> esistono.</p> <p><code>CHECK.txt</code> e <code>ERRORS.txt</code> non esistono.</p>	
Output	
<code>FileNotFoundException</code>	
Test Driver	
Classe: <code>TestRepoMiningIntegration</code>	Metodo: <code>test_case_13</code>
Oracolo	

CHECKS.txt indica status ok e la directory con id commit creata e vuota.

Tabella 2.268: Metodo dello script test_repo_mining_integration.py

Test case ID: TCI_14	Test Frame: EI1 FI1 N2 CL1 CR1 CC1 CM4 ED1 EM1 EC2 EE2
Input	
<p>initial_dataset.csv ha un formato valido, ha 50 record e contiene link esistente con repo valida, con commit con modifiche a classi .java già presenti.</p> <p>Le directory Dataset_Divided e mining_results esistono.</p> <p>CHECK.txt e ERRORS.txt non esistono.</p>	
Output	
FileNotFoundException	
Test Driver	
Classe: TestRepoMiningIntegration	Metodo: test_case_14
Oracolo	
<p>CHECKS.txt indica status ok e la directory con id commit contiene i file .java modificati e già esistenti nella repo.</p>	

Tabella 2.269: Metodo dello script test_repo_mining_integration.py

2.2.2 Text Mining

Test case ID: TCI_2.1	Test Frame: DF1
Input	
La directory "/mining_results" non esiste.	

Output	
Viene lanciata l'eccezione <code>FileNotFoundError</code> .	
Test Driver	
Classe: <code>TestTextMiningIntegration</code>	Metodo: <code>test_case_1</code>
Oracolo	
Viene lanciata l'eccezione <code>FileNotFoundError</code> .	

Tabella 2.270: Metodo dello script `test_text_mining_integration.py`

Test case ID: TCI_2.2	Test Frame: DF2 ES1
Input	
La directory <code>"/mining_results"</code> esiste, ma è vuota.	
Output	
Viene lanciata l'eccezione <code>FileNotFoundError</code> .	
Test Driver	
Classe: <code>TestTextMiningIntegration</code>	Metodo: <code>test_case_2</code>
Oracolo	
Viene lanciata l'eccezione <code>FileNotFoundError</code> .	

Tabella 2.271: Metodo dello script `test_text_mining_integration.py`

Test case ID: TCI_2.3	Test Frame: DF2 ES2 CN1 ESD2 ESF2 ESC2
Input	
La directory <code>/mining_results</code> esiste e non è vuota. Repo è presente, ma vuota.	

Output	
Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione	
Test Driver	
Classe: TestTextMiningIntegration	Metodo: test_case_3
Oracolo	
Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione	

Tabella 2.272: Metodo dello script test_text_mining_integration.py

Test case ID: TCI_2.4	Test Frame: DF2 ES2 CN2 STC1 ESD2 ESF2 ESC2
Input	
La variabile <code>cvd_id</code> è un file di testo.	
Output	
Viene lanciata l'eccezione <code>NotADirectoryError</code> .	
Test Driver	
Classe: TestTextMiningIntegration	Metodo: test_case_4
Oracolo	
Viene lanciata l'eccezione <code>NotADirectoryError</code> .	

Tabella 2.273: Metodo dello script test_text_mining_integration.py

Test case ID: TCI_2.5	Test Frame: DF2 ES2 CN2 STC2 ESD2 ESF2 ESC2
Input	
La variabile <code>cvd_id</code> è un file non ammesso, denominato CHECK.txt.	
Output	
Il file <code>text_mining.txt</code> non viene creato, <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> contengono un dizionario vuoto, il csv ha solo Name nell'intestazione	
Test Driver	
Classe: <code>TestTextMiningIntegration</code>	Metodo: <code>test_case_5</code>
Oracolo	
Il file <code>text_mining.txt</code> non viene creato, <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> contengono un dizionario vuoto, il csv ha solo Name nell'intestazione	

Tabella 2.274: Metodo dello script `test_text_mining_integration.py`

Test case ID: TCI_2.6	Test Frame: DF2 ES2 CN2 STC3 CNC1 ESD2 ESF2 ESC2
Input	
La variabile <code>cvd_id</code> è una directory vuota.	
Output	
Il file di <code>text_mining.txt</code> non viene creato, ma tutti gli altri file vengono creati vuoti.	
Test Driver	
Classe: <code>TestTextMiningIntegration</code>	Metodo: <code>test_case_6</code>
Oracolo	

Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione

Tabella 2.275: Metodo dello script test_text_mining_integration.py

Test case ID: TCI_2.7	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF1 ESD2 ESF2 ESC2
Input	
La variabile <code>cvd_id</code> è una directory non vuota e contiene un file (<code>folder</code>).	
Output	
NotADirectoryError	
Test Driver	
Classe: TestTextMiningIntegration	Metodo: <code>test_case_7</code>
Oracolo	
NotADirectoryError	

Tabella 2.276: Metodo dello script test_text_mining_integration.py

Test case ID: TCI_2.8	Test Frame: DF2 ES2 CN2 STC3 STF2 ESD2 ESF2 ESC2
Input	
La variabile <code>cvd_id</code> è una directory non vuota e contiene solo un file non ammesso (<code>folder</code> ha <code>.DS_Store</code> come nome).	
Output	

Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione	
Test Driver	
Classe: TestTextMiningIntegration	Metodo: test_case_8
Oracolo	
Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione	

Tabella 2.277: Metodo dello script test_text_mining_integration.py

Test case ID: TCI_2.9	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF1 ESD2 ESF2 ESC2
Input	
La variabile (folder è una directory vuota.	
Output	
Il file text_mining.txt non viene creato, text_mining_dict.txt e FilteredTextMining.txt contengono un dizionario vuoto, il csv ha solo Name nell'intestazione	
Test Driver	
Classe: TestTextMiningIntegration	Metodo: test_case_9
Oracolo	
Il file di text mining non viene creato, mentre tutti gli altri vengono creati vuoti.	

Tabella 2.278: Metodo dello script test_text_mining_integration.py

Test case ID: TCI_2.10	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI1 ESD2 ESF2 ESC2
Input	
La variabile <code>folder</code> è una directory non vuota, che contiene un file non ammesso (file denominato <code>.DS_Store</code>).	
Output	
Il file <code>text_mining.txt</code> non viene creato, <code>text_mining_dict.txt</code> e <code>FilteredTextMining.txt</code> contengono un dizionario vuoto, il csv ha solo Name nell'intestazione	
Test Driver	
Classe: <code>TestTextMiningIntegration</code>	Metodo: <code>test_case_10</code>
Oracolo	
Il file di text mining non viene creato, mentre tutti gli altri vengono creati vuoti.	

Tabella 2.279: Metodo dello script `test_text_mining_integration.py`

Test case ID: TCI_2.11	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI2 ESD2 ESF2 ESC2
Input	
La variabile <code>folder</code> è una directory non vuota, che contiene una sottocartella (file).	
Output	
PermissionError	
Test Driver	
Classe: <code>TestTextMiningIntegration</code>	Metodo: <code>test_case_11</code>
Oracolo	

PermissionError

Tabella 2.280: Metodo dello script test_text_mining_integration.py

Test case ID: TCI_2.12	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ1 ESTM2 ESD2 ESF2 ESC2
Input	
La variabile <code>file</code> è un file Java vuoto. <code>text_mining_dict.txt</code> , <code>Filtered_text_mining.txt</code> e <code>csv_mining_final.txt</code> non esistono.	
Output	
Il file <code>text_mining.txt</code> viene creato vuoto, mentre gli altri file vengono creati con un dizionario vuoto.	
Test Driver	
Classe: <code>TestTextMiningIntegration</code>	Metodo: <code>test_case_13</code>
Oracolo	
Il file <code>text_mining.txt</code> viene creato vuoto, mentre gli altri file vengono creati un dizionario vuoto.	

Tabella 2.281: Metodo dello script test_text_mining_integration.py

Test case ID: TCI_2.13	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ1 ESTM1 CFTM1 ESD2 ESF2 ESC2
Input	
La variabile <code>folder</code> contiene un file java vuoto e il rispettivo file di text mining vuoto.	
Output	

Il file text_mining.txt viene sovrascritto vuoto, mentre gli altri file vengono creati con un dizionario vuoto.	
Test Driver	
Classe: TestTextMiningIntegration	Metodo: test_case_14
Oracolo	
Il file text_mining.txt viene sovrascritto vuoto, mentre gli altri file vengono creati con un dizionario vuoto.	

Tabella 2.282: Metodo dello script test_text_mining_integration.py

Test case ID: TCI_2.14	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ1 ESTM1 CFTM2 ESD2 ESF2 ESC2
Input	
La variabile <code>folder</code> contiene un file java vuoto e il rispettivo file di text mining non vuoto.	
Output	
Il file text_mining.txt viene sovrascritto vuoto, come anche gli altri file	
Test Driver	
Classe: TestTextMiningIntegration	Metodo: test_case_15
Oracolo	
Il file text_mining.txt viene sovrascritto vuoto, come anche gli altri file	

Tabella 2.283: Metodo dello script test_text_mining_integration.py

Test case ID: TCI_2.15	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ2 ESTM2 ESD2 ESF2 ESC2
Input	
La variabile <code>file</code> è un file Java non vuoto. <code>text_mining_dict.txt</code> , <code>Filtered_text_mining.txt</code> e <code>csv_mining_final.txt</code> non esistono.	
Output	
Il file <code>text_mining.txt</code> viene sovrascritto, col dizionario relativo, e anche gli altri file vengono creati	
Test Driver	
Classe: <code>TestTextMiningIntegration</code>	Metodo: <code>test_case_16</code>
Oracolo	
Il file <code>text_mining.txt</code> viene creato, col dizionario relativo, e anche gli altri file vengono creati	

Tabella 2.284: Metodo dello script `test_text_mining_integration.py`

Test case ID: TCI_2.16	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ2 ESTM2 ESD1 ESF1 ESC1
Input	
La variabile <code>file</code> è un file Java non vuoto. <code>text_mining_dict.txt</code> , <code>Filtered_text_mining.txt</code> e <code>csv_mining_final.txt</code> esistono.	
Output	
Il file <code>text_mining.txt</code> viene creato, col dizionario relativo, e anche gli altri file vengono sovrascritti.	
Test Driver	

Classe: TestTextMiningIntegration	Metodo: test_case_17
Oracolo	
Il file text_mining.txt viene creato, col dizionario relativo, e anche gli altri file vengono sovrascritti.	

Tabella 2.285: Metodo dello script test_text_mining_integration.py

Test case ID: TCI_2.17	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ2 ESTM1 CFTM1 ESD2 ESF2 ESC2
Input	
La variabile <code>folder</code> contiene un file java non vuoto e il rispettivo file di text mining vuoto.	
Output	
Il file text_mining.txt viene sovrascritto, non contiene più un dizionario vuoto, col dizionario relativo, e anche gli altri file vengono creati. Il csv ha 68 righe, una per ogni file java e il relativo text mining.	
Test Driver	
Classe: TestTextMiningIntegration	Metodo: test_case_18
Oracolo	
Il file text_mining.txt viene sovrascritto, non contiene più un dizionario vuoto, col dizionario relativo, e anche gli altri file vengono creati. Il csv deve avere 34 righe.	

Tabella 2.286: Metodo dello script test_text_mining_integration.py

Test case ID: TCI_2.18	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI3 CFJ2 ESTM1 CFTM2 ESD2 ESF2 ESC2
Input	
La variabile <code>folder</code> contiene un file java non vuoto e il rispettivo file di text mining vuoto.	
Output	
Il file <code>text_mining.txt</code> viene sovrascritto, non contiene più un dizionario vuoto, col dizionario relativo, e anche gli altri file vengono creati. Il csv ha 68 righe, una per ogni file java e il relativo text mining.	
Test Driver	
Classe: <code>TestTextMiningIntegration</code>	Metodo: <code>test_case_19</code>
Oracolo	
Il file <code>text_mining.txt</code> viene sovrascritto, non contiene più un dizionario vuoto, col dizionario relativo, e anche gli altri file vengono creati. Il csv deve avere 34 righe.	

Tabella 2.287: Metodo dello script `test_text_mining_integration.py`

Test case ID: TCI_2.19	Test Frame: DF2 ES2 CN2 STC3 CNC2 STF3 CNF2 CNFI4 ESD2 ESF2 ESC2
Input	
La variabile <code>file</code> è un file di testo non vuoto.	
Output	
Il file <code>text_mining.txt</code> viene creato, non contiene più un dizionario vuoto, col dizionario relativo, e anche gli altri file vengono creati.	
Test Driver	

Classe: TestTextMiningIntegration	Metodo: test_case_20
Oracolo	
Il file text_mining.txt viene creato, non contiene più un dizionario vuoto, col dizionario relativo, e anche gli altri file vengono creati.	

Tabella 2.288: Metodo dello script test_text_mining_integration.py

2.2.3 Rifinitura dati analisi statica

Test case ID: TCI_3.1	Test Frame: EN2 EP1 NP2 FCP1 TCP2
Input	
RepositoryMining_ASAResults_neg.csv non è presente. RepositoryMining_ASAResults_pos.csv è presente, è valido e contiene vulnerabilità	
Output	
csv_ASA_final.csv contiene solo componenti con classe 'pos'.	
Test Driver	
Classe: TestASAIntegration	Metodo: test_case_1
Oracolo	
csv_ASA_final.csv contiene solo componenti con classe 'pos'.	

Tabella 2.289: Metodo dello script test_asa_integration.py

Test case ID: TCI_3.2	Test Frame: EN1 NN2 FCN1 TCN2 EP2
Input	
RepositoryMining_ASAResults_neg.csv esiste, è valido e contiene vulnerabilità. RepositoryMining_ASAResults_pos.csv non è presente.	

Output	
csv_ASA_final.csv contiene solo componenti con classe 'neg'.	
Test Driver	
Classe: TestASAIntegration	Metodo: test_case_2
Oracolo	
csv_ASA_final.csv contiene solo componenti con classe 'neg'.	

Tabella 2.290: Metodo dello script test_asa_integration.py

Test case ID: TCI_3.3	Test Frame: EN1 FCN2
Input	
RepositoryMining_ASAResults_neg.csv esiste ma è in formato non valido. RepositoryMining_ASAResults_pos.csv esiste, è valido e contiene vulnerabilità.	
Output	
IndexError	
Test Driver	
Classe: TestASAIntegration	Metodo: test_case_3
Oracolo	
IndexError	

Tabella 2.291: Metodo dello script test_asa_integration.py

Test case ID: TC_3.4	Test Frame: EP1 FCP2
Input	

RepositoryMining_ASAResults_neg.csv esiste, è valido e contiene vulnerabilità. RepositoryMining_ASAResults_pos.csv esiste ma è in formato non valido.	
Output	
IndexError	
Test Driver	
Classe: TestASAIntegration	Metodo: test_case_4
Oracolo	
IndexError	

Tabella 2.292: Metodo dello script test_asa_integration.py

Test case ID: TC_3.5	Test Frame: EN1 NN2 FCN1 TCN1 EP1 NP2 FCP1 TCP1
Input	
RepositoryMining_ASAResults_neg.csv e RepositoryMining_ASAResults_pos.csv esistono, sono validi e non contengono vulnerabilità.	
Output	
csv_ASA_final.csv contiene solo l'intestazione 'Name, class'.	
Test Driver	
Classe: TestASAIntegration	Metodo: test_case_5
Oracolo	
csv_ASA_final.csv contiene solo l'intestazione 'Name, class'.	

Tabella 2.293: Metodo dello script test_asa_integration.py

Test case ID: TC_3.6	Test Frame: EN1 NN2 FCN1 TCN2 EP1 NP2 FCP1 TCP2
Input	
RepositoryMining_ASAResults_neg.csv e RepositoryMining_ASAResults_pos.csv esistono, sono validi e contengono vulnerabilità non ripetute per componenti.	
Output	
csv_ASA_final.csv contiene ciascun componente con vulnerabilità e frequenza pari a 1 per la rispettiva regola.	

Tabella 2.294: Metodo dello script test_asa_integration.py.

Test Driver	
Classe: TestASAIntegration	Metodo: test_case_6
Oracolo	
csv_ASA_final.csv contiene ciascun componente con vulnerabilità e frequenza pari a 1 per la rispettiva regola.	

Tabella 2.294: Metodo dello script test_asa_integration.py

Test case ID: TC_3.7	Test Frame: EN1 NN3 FCN1 TCN3 EP1 NP3 FCP1 TCP3
Input	
RepositoryMining_ASAResults_neg.csv e RepositoryMining_ASAResults_pos.csv esistono, sono validi e contengono vulnerabilità ripetute per componenti.	
Output	

csv_ASA_final.csv contiene ciascun componente con vulnerabilità e frequenza maggiore di 1 per la regola ripetuta.	
Test Driver	
Classe: TestASAIntegration	Metodo: test_case_7
Oracolo	
csv_ASA_final.csv contiene ciascun componente con vulnerabilità e frequenza maggiore di 1 per la regola ripetuta.	

Tabella 2.295: Metodo dello script test_asa_integration.py

Test case ID: TC_8	Test Frame: EN1 NN2 FCN1 TCN2 EP1 NP1 FCP1
Input	
RepositoryMining_ASAResults_neg.csv e RepositoryMining_ASAResults_pos.csv esistono e sono validi. RepositoryMining_ASAResults_pos.csv è vuoto, RepositoryMining_ASAResults_neg.csv contiene una vulnerabilità	
Output	
csv_ASA_final.csv con frequenza pari a 1 in corrispondenza delle regole ripetute per componente di classe neg.	
Test Driver	
Classe: TestASAIntegration	Metodo: test_case_8
Oracolo	
csv_ASA_final.csv con frequenza pari a 1 in corrispondenza delle regole ripetute per componente di classe neg.	

Tabella 2.296: Metodo dello script test_asa_integration.py

Test case ID: TC_9	Test Frame: EN1 NN1 FCN1 EP1 NP2 FCP1 TCP2
Input	
RepositoryMining_ASAResults_neg.csv e RepositoryMining_ASAResults_pos.csv esistono e sono validi. RepositoryMining_ASAResults_neg.csv è vuoto, RepositoryMining_ASAResults_pos.csv contiene una vulnerabilità	
Output	
csv_ASA_final.csv con frequenza pari a 1 in corrispondenza delle regole ripetute per componente di classe pos.	
Test Driver	
Classe: TestASAIntegration	Metodo: test_case_9
Oracolo	
csv_ASA_final.csv ccon frequenza = 1 in corrispondenza delle regole ripetute per componente di classe pos.	

Tabella 2.297: Metodo dello script test_asa_integration.py

Test case ID: TC_10	Test Frame: EN1 NN1 FCN1 EP1 NP1 FCP1
Input	
RepositoryMining_ASAResults_neg.csv e RepositoryMining_ASAResults_pos.csv esistono, sono validi e sono vuoti	
Output	

csv_ASA_final.csv contiene solo l'intestazione 'Name, class'.	
Test Driver	
Classe: TestASAIntegration	Metodo: test_case_10
Oracolo	
csv_ASA_final.csv contiene solo l'intestazione 'Name, class'.	

Tabella 2.298: Metodo dello script test_asa_integration.py

2.2.4 Union

TotalCombination - Union_TMwithASA

Test case ID: TCI_4.1	Test Frame: EM2
Input	
Il file csv_mining_final.csv non esiste	
Output	
FileNotFoundError	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_1
Oracolo	
FileNotFoundError	

Tabella 2.299: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.2	Test Frame: EA2
------------------------------	------------------------

Input	
Il file csv_ASA_final.csv non esiste	
Output	
FileNotFoundException	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_2
Oracolo	
FileNotFoundException	

Tabella 2.300: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.3	Test Frame: EU2
Input	
Il file Union_TM_ASA.csv non esiste	
Output	
FileNotFoundException	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_3
Oracolo	
FileNotFoundException	

Tabella 2.301: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.4	Test Frame: ES2
-----------------------	-----------------

Input	
Il file mining_results_sm_final.csv non esiste	
Output	
FileNotFoundException	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_4
Oracolo	
FileNotFoundException	

Tabella 2.302: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.5	Test Frame: EM1 NM2 FM1 EA1 NA2 FA1 EU1 NU2 FU1 ES1 NS2 FS1
Input	
Il file csv_mining_final.csv esiste, e contiene solo l'intestazione Il file csv_ASA_final.csv esiste, e contiene solo l'intestazione Il file Union_TM_ASA.csv esiste, e contiene solo l'intestazione Il file mining_results_sm_final.csv esiste, e contiene solo l'intestazione	
Output	
Il file di Unione conterrà la combinazione delle intestazioni dei file	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_5

Oracolo
Il file di Unione conterrà la combinazione delle intestazioni dei file

Tabella 2.303: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.6	Test Frame: EM1 NM3 FM2
Input	
Il file csv_mining_final.csv ha un formato non valido	
Output	
UnicodeDecodeError	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_6
Oracolo	
UnicodeDecodeError	

Tabella 2.304: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.7	Test Frame: EA1 NA3 FA2
Input	
Il file csv_ASA_final.csv ha un formato non valido	
Output	
UnicodeDecodeError	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_7

Oracolo
UnicodeDecodeError

Tabella 2.305: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.8	Test Frame: EU1 NU3 FU2
Input	
Il file Union_TM_ASA.csv ha un formato non valido	
Output	
UnicodeDecodeError	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_8
Oracolo	
UnicodeDecodeError	

Tabella 2.306: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.9	Test Frame: ES1 NS3 FS2
Input	
Il file mining_results_sm_final.csv ha un formato non valido	
Output	
UnicodeDecodeError	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_9

Oracolo
UnicodeDecodeError

Tabella 2.307: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.10	Test Frame: EM1 NM1 FM1 EA1 NA1 FA1 EU1 NU1 FU1 ES1 NS1 FS1
Input	
Il file csv_mining_final.csv esiste ed è vuoto Il file csv_ASA_final.csv esiste ed è vuoto Il file Union_TM_ASA.csv esiste ed è vuoto Il file mining_results_sm_final.csv esiste ed è vuoto	
Output	
Union_TM_ASA.csv viene generato vuoto così come 3COMBINATION.csv	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_10
Oracolo	
Union_TM_ASA.csv viene generato vuoto così come 3COMBINATION.csv	

Tabella 2.308: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.11	Test Frame: EM1 NM1 FM1 EA1 NA1 FA1 EU1 NU1 FU1 ES1 NS3 FS1
------------------------	--

Input	
<p>Il file csv_mining_final.csv esiste ed è vuoto</p> <p>Il file csv_ASA_final.csv esiste ed è vuoto</p> <p>Il file Union_TM_ASA.csv esiste ed è vuoto</p> <p>Il file mining_results_sm_final.csv esiste e non è vuoto</p>	
Output	
Union_TM_ASA.csv viene generato vuoto così come 3COMBINATION.csv	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_11
Oracolo	
Union_TM_ASA.csv viene generato vuoto così come 3COMBINATION.csv	

Tabella 2.309: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.12	Test Frame: EM1 NM3 FM1 EA1 NA3 FA1 EU1 NU3 FU1 ES1 NS1 FS1
Input	
<p>Il file csv_mining_final.csv esiste e non è vuoto</p> <p>Il file csv_ASA_final.csv esiste e non è vuoto</p> <p>Il file Union_TM_ASA.csv esiste e non è vuoto</p> <p>Il file mining_results_sm_final.csv esiste ed è vuoto</p>	
Output	
<p>Union_TMwithAsa.py e TotalCombination.py eseguono correttamente l'unione tra i file che conterrà l'unione tra i primi due CSV</p>	

Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_12
Oracolo	
Union_TMwithAsa.py e TotalCombination.py eseguono correttamente l'unione tra i file che conterrà l'unione tra i primi due CSV	

Tabella 2.310: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.13	Test Frame: EM1 NM3 FM1 EA1 NA3 FA1 EU1 NU3 FU1 ES1 NS3 FS1
Input	
Il file csv_mining_final.csv esiste e non è vuoto Il file csv_ASA_final.csv esiste e non è vuoto Il file Union_TM_ASA.csv esiste e non è vuoto Il file mining_results_sm_final.csv esiste e non è vuoto	
Output	
Union_TMwithAsa.py e TotalCombination.py eseguono correttamente, generando i file Union_TM_ASA.csv e 3COMBINATION.csv, entrambi validi e contenenti più di un record.	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_13
Oracolo	
Union_TMwithAsa.py e TotalCombination.py eseguono correttamente, generando i file Union_TM_ASA.csv e 3COMBINATION.csv, entrambi validi e contenenti più di un record.	

Tabella 2.311: Integrazione tra TotalCombination e Union_TMwithASA

Test case ID: TCI_4.14	Test Frame: EM1 NM3 FM1 EA1 NA1 FA1 EU1 NU3 FU1 ES1 NS3 FS1
Input	
<p>Il file csv_mining_final.csv esiste e non è vuoto</p> <p>Il file csv_ASA_final.csv esiste ed è vuoto</p> <p>Il file Union_TM_ASA.csv esiste e non è vuoto</p> <p>Il file mining_results_sm_final.csv esiste e non è vuoto</p>	
Output	
<p>Union_TMwithAsa.py e TotalCombination.py eseguono correttamente, generando i file Union_TM_ASA.csv e 3COMBINATION.csv, entrambi validi e contenenti più di un record.</p>	
Test Driver	
Classe: TestTotalCombinationIntegration	Metodo: test_case_14
Oracolo	
<p>Union_TMwithAsa.py e TotalCombination.py eseguono correttamente, generando i file Union_TM_ASA.csv e 3COMBINATION.csv, entrambi validi e contenenti più di un record.</p>	

Tabella 2.312: Integrazione tra TotalCombination e Union_TMwithASA**Union_SMwithASA**

Test case ID: TCI_5.1	Test Frame: EA2
Input	
<p>Il file csv_ASA_final non esiste</p>	
Output	

FileNotFoundException	
Test Driver	
Classe: TestUnionSMwithASAIntegration	Metodo: test_case_1
Oracolo	
FileNotFoundException	

Tabella 2.313: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.2	Test Frame: EA1 NA1 ES1 NS1
Input	
csv_ASA_final.csv esiste ed è vuoto mining_results_sm_final.csv esiste ed è vuoto	
Output	
Il file di unione è valido e vuoto	
Test Driver	
Classe: TestUnionSMwithASAIntegration	Metodo: test_case_2
Oracolo	
Il file di unione è valido e vuoto	

Tabella 2.314: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.3	Test Frame: EA1 NA2 FA1 ES1 NS2 FS1
Input	

csv_ASA_final.csv esiste e contiene solo l'intestazione mining_results_sm_final.csv esiste e contiene solo l'intestazione	
Output	
Il file di Unione è valido, e contiene la combinazione delle due intestazioni	
Test Driver	
Classe: TestUnionSMwithASAIntegration	Metodo: test_case_3
Oracolo	
Il file di Unione è valido, e contiene la combinazione delle due intestazioni	

Tabella 2.315: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.4	Test Frame: EA1 NA3 FA2
Input	
csv_ASA_final.csv esiste e contiene un formato un valido	
Output	
UnicodeDecodeError	
Test Driver	
Classe: TestUnionSMwithASAIntegration	Metodo: test_case_4
Oracolo	
UnicodeDecodeError	

Tabella 2.316: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.5	Test Frame: EA1 NA3 FA1 ES1 NS3 FS2
Input	
mining_results_sm_final.csv contiene un formato non valido	
Output	
UnicodeDecodeError	
Test Driver	
Classe: TestUnionSMwithASAIntegration	Metodo: test_case_5
Oracolo	
UnicodeDecodeError	

Tabella 2.317: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.6	Test Frame: EA1 NA2 FA1 ES1 NS2 FS1
Input	
csv_ASA_final.csv esiste e contiene l'intestazione ed i dati mining_results_sm_final.csv esiste e contiene l'intestazione ed i dati	
Output	
Il file di Unione è valido, e contiene la combinazione delle due intestazioni e dei dati	
Test Driver	
Classe: TestUnionSMwithASAIntegration	Metodo: test_case_6
Oracolo	

Il file di Unione è valido, e contiene la combinazione delle due intestazioni

Tabella 2.318: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.7	Test Frame: EA1 NA1 FA1 ES1 NS3 FS1
Input	
csv_ASA_final.csv esiste ed è vuoto mining_results_sm_final.csv esiste e contiene l'intestazione ed i dati	
Output	
Lo script esegue l'unione e rimpiazza i dati mancanti nel file di unione con valori '0'	
Test Driver	
Classe: TestUnionSMwithASAIntegration	Metodo: test_case_7
Oracolo	
Lo script esegue l'unione e rimpiazza i dati mancanti nel file di unione con valori '0'	

Tabella 2.319: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.8	Test Frame: EA1 NA3 FA1 ES1 NS1 FS1
Input	
csv_ASA_final.csv esiste e contiene intestazione e dati mining_results_sm_final.csv esiste ed è vuoto	
Output	
Lo script genera un file di output vuoto	

Test Driver	
Classe: TestUnionSMwithASAIntegration	Metodo: test_case_8
Oracolo	
Lo script genera un file di output vuoto	

Tabella 2.320: Metodo dello script union_SMwithASA.py

Test case ID: TCI_5.9	Test Frame: EA1 NA3 FA1 ES2
Input	
Il file mining_results_sm_final.csv non esiste	
Output	
FileNotFoundError	
Test Driver	
Classe: TestUnionSMwithASAIntegration	Metodo: test_case_9
Oracolo	
FileNotFoundError	

Tabella 2.321: Metodo dello script union_SMwithASA.py**Union**

Test case ID: TCI_6.1	Test Frame: ET2
Input	
Il file csv_mining_final non esiste	

Output	
FileNotFoundError	
Test Driver	
Classe: TestUnionIntegration	Metodo: test_case_1
Oracolo	
FileNotFoundError	

Tabella 2.322: Metodo dello script Union.py

Test case ID: TCI_6.2	Test Frame: ET1 NT1 ES1 NS1
Input	
csv_mining_final.csv esiste ed è vuoto mining_results_sm_final.csv esiste ed è vuoto	
Output	
Il file di unione è valido e vuoto	
Test Driver	
Classe: TestUnionIntegration	Metodo: test_case_2
Oracolo	
Il file di unione è valido e vuoto	

Tabella 2.323: Metodo dello script Union.py

Test case ID: TCI_6.3	Test Frame: ET1 NT2 FT1 ES1 NS2 FS1
-----------------------	--

Input	
csv_mining_final.csv esiste e contiene solo l'intestazione mining_results_sm_final.csv esiste e contiene solo l'intestazione	
Output	
Il file di Unione è valido, e contiene la combinazione delle due intestazioni	
Test Driver	
Classe: TestUnionIntegration	Metodo: test_case_3
Oracolo	
Il file di Unione è valido, e contiene la combinazione delle due intestazioni	

Tabella 2.324: Metodo dello script Union.py

Test case ID: TCI_6.4	Test Frame: ET1 NT3 FT2
Input	
csv_mining_final.csv esiste e contiene un formato un valido	
Output	
UnicodeDecodeError	
Test Driver	
Classe: TestUnionIntegration	Metodo: test_case_4
Oracolo	
UnicodeDecodeError	

Tabella 2.325: Metodo dello script Union.py

Test case ID: TCI_6.5	Test Frame: ET1 NT3 FT1 ES1 NS3 FS2
Input	
mining_results_sm_final.csv contiene un formato non valido	
Output	
UnicodeDecodeError	
Test Driver	
Classe: TestUnionIntegration	Metodo: test_case_5
Oracolo	
UnicodeDecodeError	

Tabella 2.326: Metodo dello script Union.py

Test case ID: TCI_6.6	Test Frame: ET1 NT2 FT1 ES1 NS2 FS1
Input	
csv_mining_final.csv esiste e contiene l'intestazione ed i dati mining_results_sm_final.csv esiste e contiene l'intestazione ed i dati	
Output	
Il file di Unione è valido, e contiene la combinazione delle due intestazioni e dei dati	
Test Driver	
Classe: TestUnionIntegration	Metodo: test_case_6
Oracolo	

Il file di Unione è valido, e contiene la combinazione delle due intestazioni

Tabella 2.327: Metodo dello script Union.py

Test case ID: TCI_6.7	Test Frame: ET1 NT1 FT1 ES1 NS3 FS1
Input	
csv_mining_final.csv esiste ed è vuoto mining_results_sm_final.csv esiste e contiene l'intestazione ed i dati	
Output	
Il file di Unione contiene i dati di mining_results_sm_final.csv	
Test Driver	
Classe: TestUnionIntegration	Metodo: test_case_7
Oracolo	
Il file di Unione contiene i dati di mining_results_sm_final.csv	

Tabella 2.328: Metodo dello script Union.py

Test case ID: TCI_6.8	Test Frame: ET1 NT3 FT1 ES1 NS1 FS1
Input	
csv_mining_final.csv esiste e contiene intestazione e dati mining_results_sm_final.csv esiste ed è vuoto	
Output	
Lo script genera un file di output vuoto	

Test Driver	
Classe: TestUnionIntegration	Metodo: test_case_8
Oracolo	
Lo script genera un file di output vuoto	

Tabella 2.329: Metodo dello script Union.py

Test case ID: TCI_6.9	Test Frame: ET1 NT3 FT1 ES2
Input	
Il file mining_results_sm_final.csv non esiste	
Output	
FileNotFoundError	
Test Driver	
Classe: TestUnionIntegration	Metodo: test_case_9
Oracolo	
FileNotFoundError	

Tabella 2.330: Metodo dello script Union.py