Report

Assignment 1
Section B

Group Members: **Muhammad Saqib - 19i-0494**
Muhammad Taimoor - 19i-0552

Introduction

This report describes the work done by Taimoor and Saqib to improve the accuracy and security of a naive classifier and test the code continuously. We team used CodeQL for security analysis and Jenkins for continuous integration to automate testing and improve the overall quality of the code.

To improve the accuracy and security of the naive classifier, we forked a GitHub repository that included the initial codebase. They then created two branches, one for Taimoor to refactor the code and another for Saqib to improve the accuracy of the model. We then created a workflow of CodeQL for security analysis, which was integrated with the GitHub repository. Whenever a push was made to the branches, the CodeQL analysis was automatically triggered.

To automate testing, we set up Jenkins and integrated their forked repository to it. We wrote a bash command that runs the test.py file, which was executed by Jenkins in the form of a job. This allowed us to test the code continuously and identify any issues that arose during development.

Taimoor's main contribution was in refactoring the code. He restructured the codebase to make it more modular, removing any redundant or duplicate code. This made the code easier to read and maintain, reducing the likelihood of introducing bugs or vulnerabilities.

Saqib focused on improving the accuracy of the model. He experimented with different algorithms and techniques to enhance the classifier's performance, using both existing literature and his own ideas. He also created a comprehensive test suite that enabled us to verify the accuracy of the model.

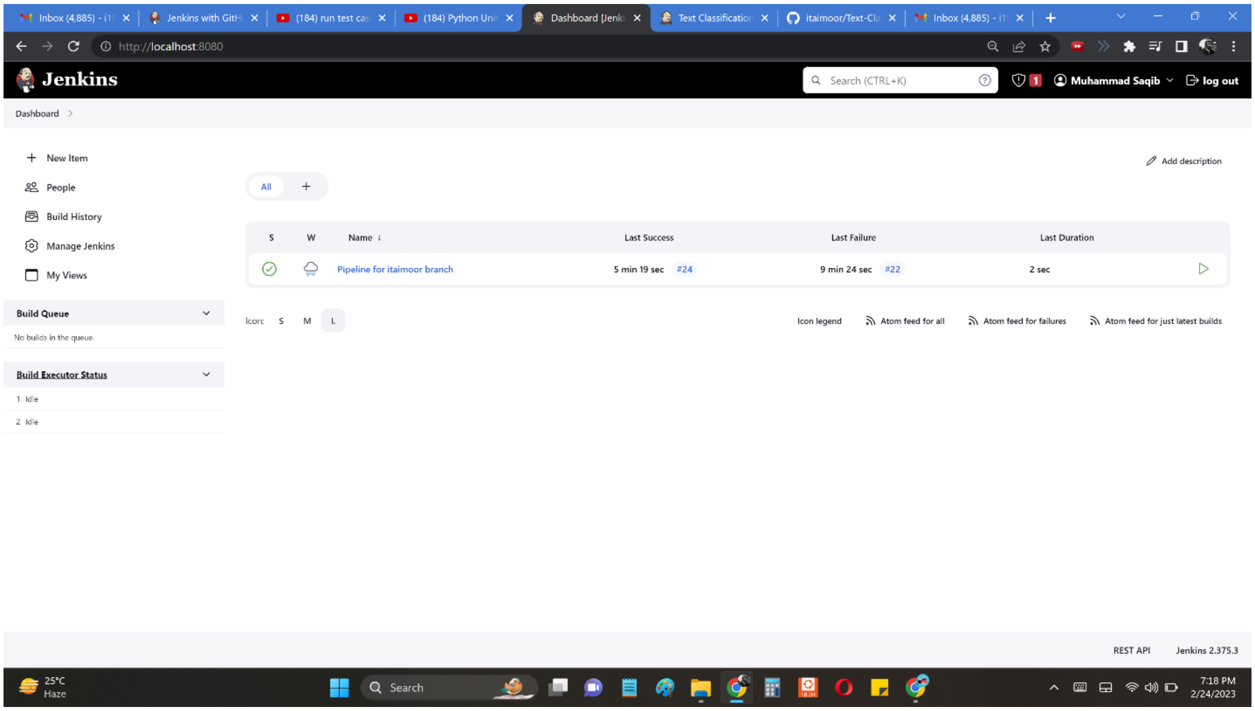
Our efforts resulted in significant improvements to the naive classifier. The security analysis using CodeQL, and the integration of Jenkins enabled continuous testing, which helped us to identify and fix bugs more quickly. The refactor of the codebase made it more maintainable, while the improvements to the accuracy of the model resulted in better performance.

Conclusion

The team's approach of using CodeQL for security analysis and Jenkins for continuous integration proved to be effective in improving the accuracy and security of the naive classifier.

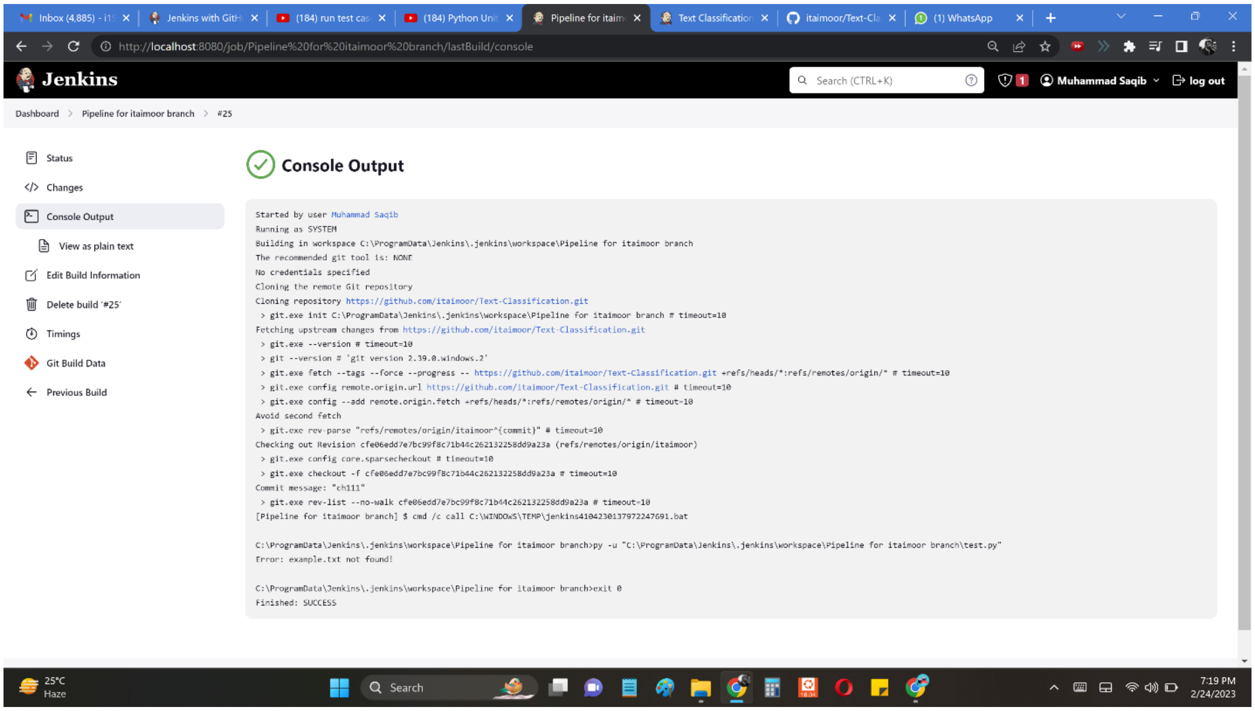
Screenshots

Jobs



S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☁	Pipeline for itaimoor branch	5 min 19 sec #24	9 min 24 sec #22	2 sec

Output



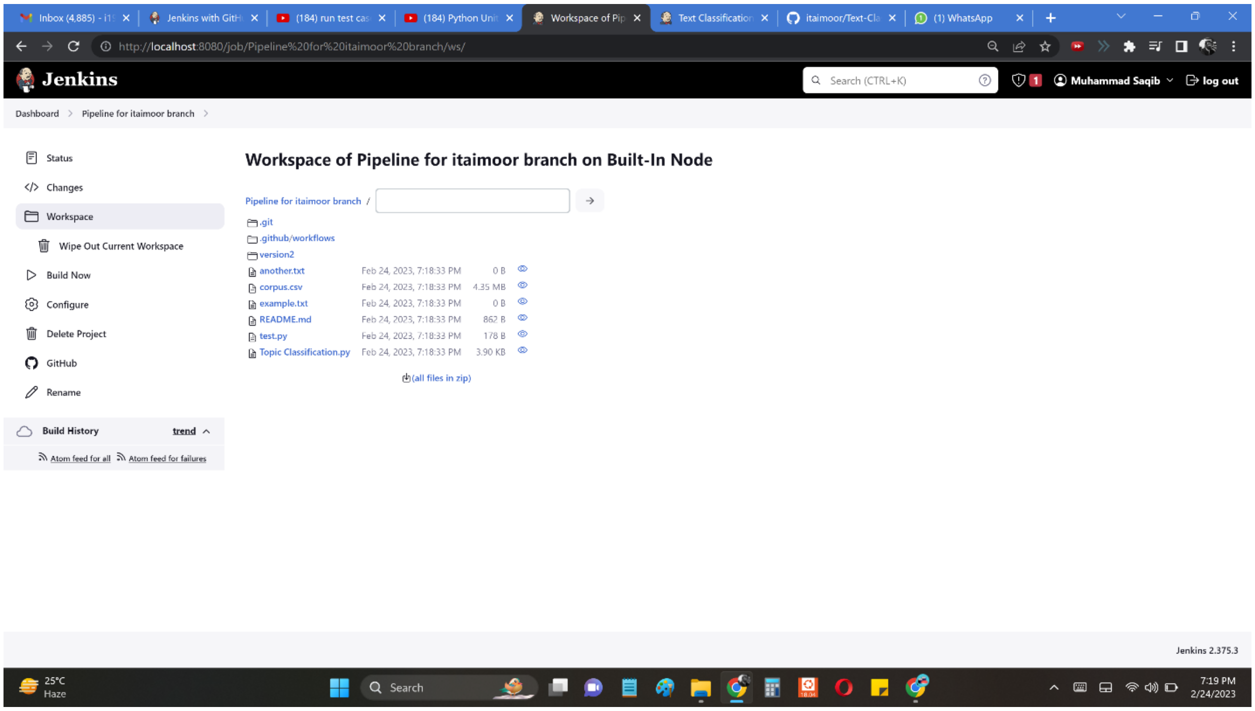
Console Output

```
Started by user Muhammad Saqib
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Pipeline for itaimoor branch
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/itaimoor/Text-Classification.git
 > git.exe init C:\ProgramData\Jenkins\.jenkins\workspace\Pipeline for itaimoor branch # timeout=10
Fetching upstream changes from https://github.com/itaimoor/Text-Classification.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.39.0.windows.2'
 > git.exe fetch --tags --force --progress -- https://github.com/itaimoor/Text-Classification.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe config remote.origin.url https://github.com/itaimoor/Text-Classification.git # timeout=10
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git.exe rev-parse "refs/remotes/origin/itaimoor^{commit}" # timeout=10
Checking out Revision cfe06edd7e7bc99f8c71b44c262132258dd9a23a (refs/remotes/origin/itaimoor)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f cfe06edd7e7bc99f8c71b44c262132258dd9a23a # timeout=10
Commit message: "ch111"
 > git.exe rev-list --no-walk cfe06edd7e7bc99f8c71b44c262132258dd9a23a # timeout=10
[Pipeline for itaimoor branch] $ cmd /c call C:\WINDOWS\TEMP\jenkins4104230137972247691.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Pipeline for itaimoor branch>py -u "C:\ProgramData\Jenkins\.jenkins\workspace\Pipeline for itaimoor branch\test.py"
Error: example.txt not found!

C:\ProgramData\Jenkins\.jenkins\workspace\Pipeline for itaimoor branch>exit 0
Finished: SUCCESS
```

Workplace



Workspace of Pipeline for itaimoor branch on Built-In Node

Pipeline for itaimoor branch /

.git			
.github/workflows			
version2			
another.txt	Feb 24, 2023, 7:18:33 PM	0 B	
corpus.csv	Feb 24, 2023, 7:18:33 PM	4.35 MB	
example.txt	Feb 24, 2023, 7:18:33 PM	0 B	
README.md	Feb 24, 2023, 7:18:33 PM	862 B	
test.py	Feb 24, 2023, 7:18:33 PM	178 B	
Topic Classification.py	Feb 24, 2023, 7:18:33 PM	3.90 KB	

(all files in zip)