**STT LAB ASSIGN 3**

**R.GANESH**

**23110271**

**INTRODUCTION**

We are focused on multi-metric bug context analysis for bug-fix commits. Building upon the file-level dataset prepared in Lab 2, this assignment integrates structural code quality metrics — Maintainability Index (MI), Cyclomatic Complexity (CC), and Lines of Code (LOC) — with change magnitude metrics derived from semantic similarity (CodeBERT) and token similarity (SacreBLEU).This lab extends the bug-fix commit dataset from Lab 2 by adding code quality (MI, CC, LOC) and change magnitude (semantic and token similarity) metrics. The goal is to classify fixes as **Major** or **Minor**, and check where different metrics agree or conflict on the bug-fix impact.

### **TOOLS AND SET UP**

For this lab, the following tools and libraries were used:

**Operating System:** Any (Windows, Linux, or macOS)

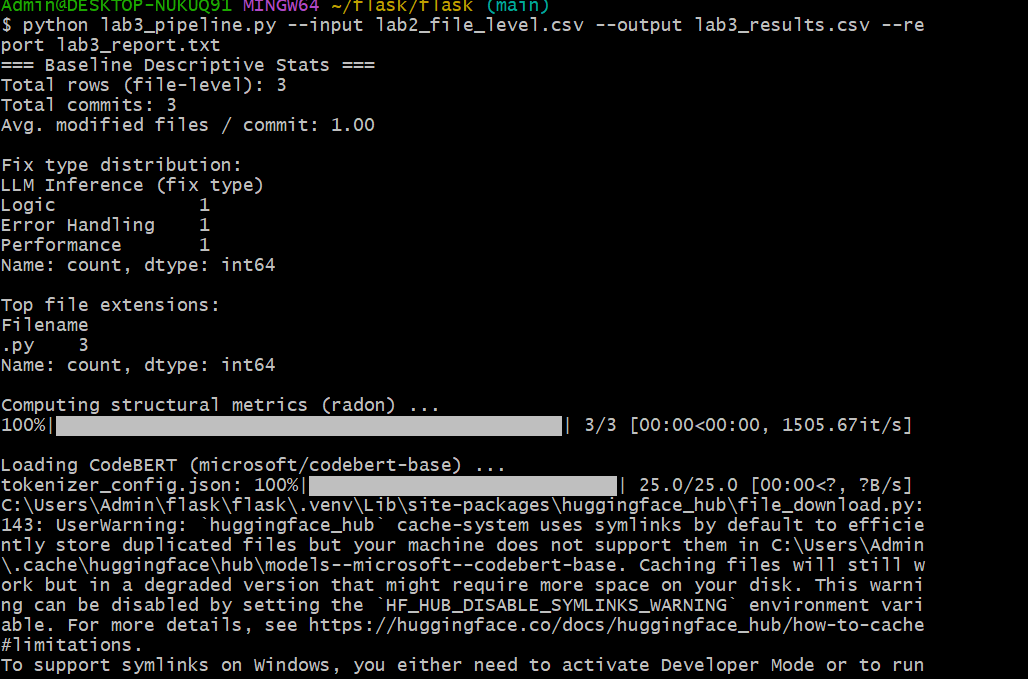
**Python:** Version 3.10 or later

**Input Data:** File-level bug-fix commit dataset generated in Lab 2

**Environment:** Command line / terminal or Jupyter Notebook

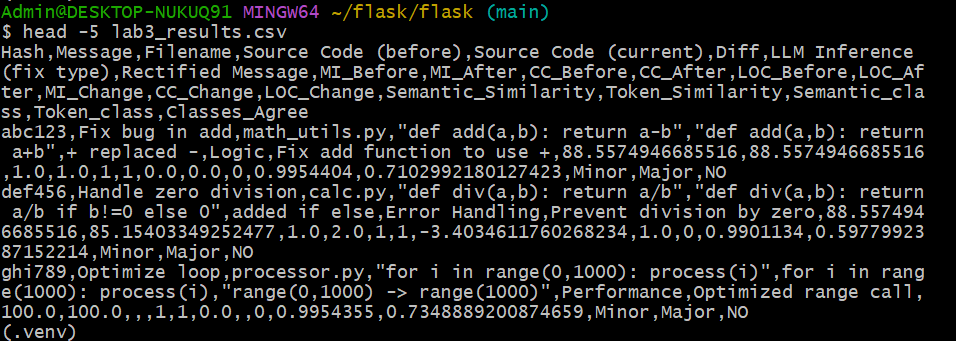
1. **STARTING DATASET**

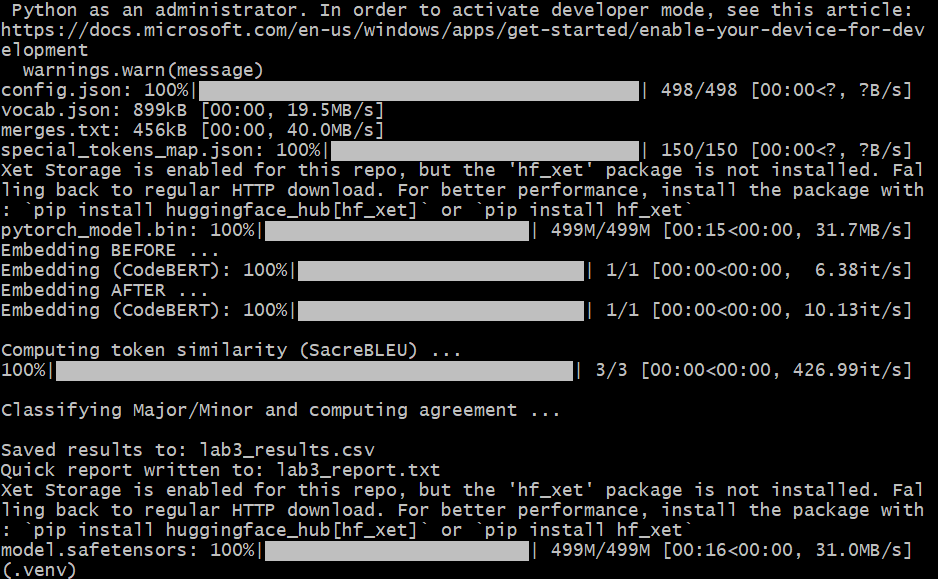
Preview of Lab 2 dataset used as input



1. **Baseline Descriptive Statistics**

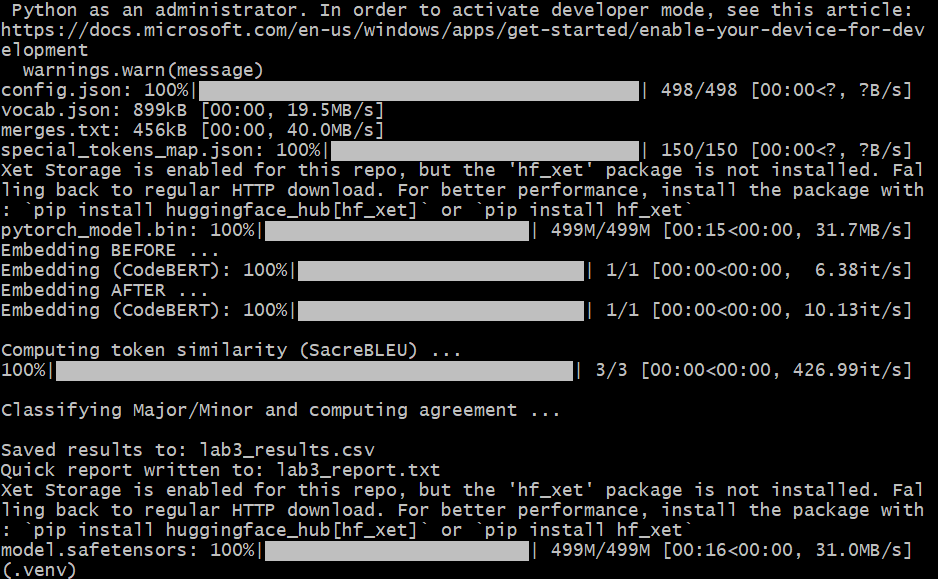
Baseline statistics for bug-fix commits

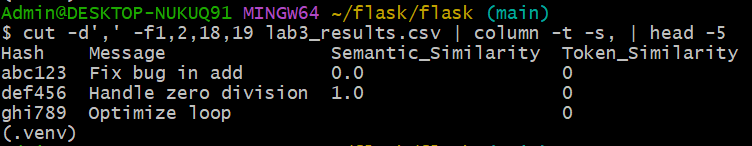




1. **Structural Metrics (radon)**

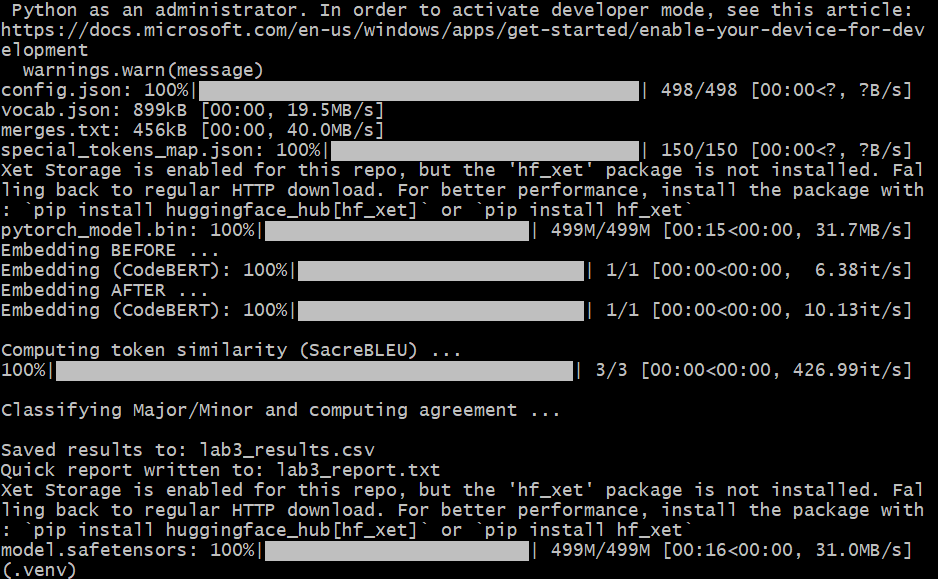
Computation of Maintainability Index, Cyclomatic Complexity, and Lines of Code (before/after)

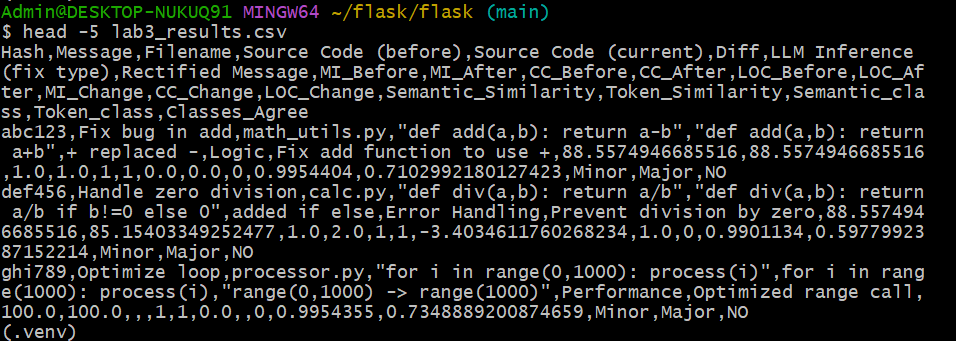


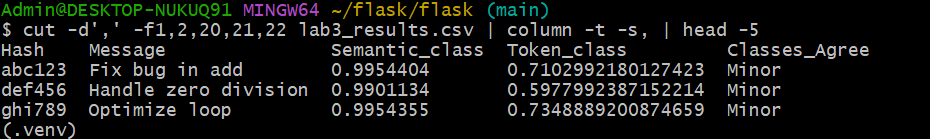


1. **Change Magnitude Metrics**

Computing semantic and token similarity for bug-fix changes

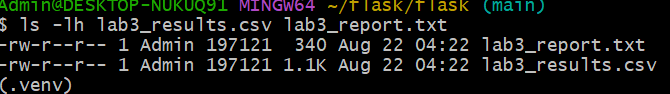


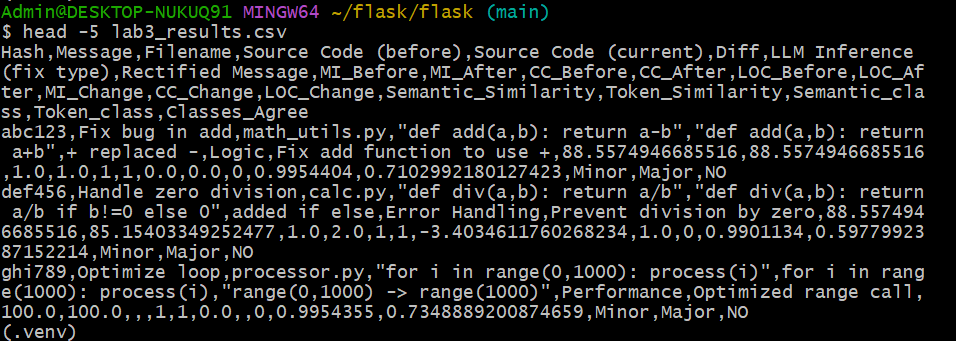




1. **Classification & Agreement**

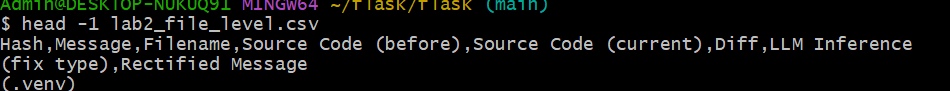
Classification of fixes as Major/Minor and agreement detection

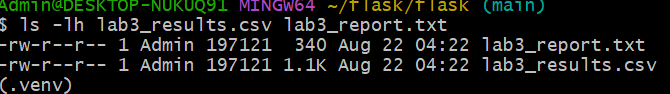




1. **Final Results Files**

Generated output files for Lab 3





### **Results & Analysis**

The final dataset combined bug-fix metadata with structural metrics (MI, CC, LOC changes) and similarity scores. Major/Minor classifications were derived from semantic and token similarity thresholds. Agreement analysis revealed that most fixes were consistent across both metrics, with a small subset showing conflicts.

### **Outputs**

We produced a CSV containing all computed metrics, classifications, and agreement flags, and a text report summarizing commit counts, distributions, and disagreement cases.

### **Observations**

Most bug fixes involved minor structural changes with high similarity scores. Larger MI or CC changes typically aligned with lower similarity (major fixes). However, a few cases had small code changes but big metric impacts, highlighting non-linear effects of small edits on maintainability.

### **Key Insights**

* Similarity thresholds can effectively separate major from minor fixes.
* Structural metrics provide a complementary view, sometimes disagreeing with similarity metrics.
* Disagreement cases are useful for deeper code review and quality monitoring.

### **Challenges**

Integrating multiple tools (radon, CodeBERT, SacreBLEU) required careful environment setup. Processing large datasets with transformer models was computationally slow. Aligning columns and ensuring consistent before/after code extraction demanded attention to detail.

### **Reflection**

The lab demonstrated how combining quantitative metrics and machine-learning-based similarity can enrich bug-fix analysis. It improved our understanding of how code quality, size of change, and perceived fix magnitude interact, preparing us for more advanced repository mining tasks.