Day 2: Version Control System (Git)

Version control is essential for tracking changes, collaborating across teams, and rolling back when experiments or data preprocessing need correction. Use cases: - Rolling back to previous data preprocessing when model results are unsatisfactory. - Collaboration across different time zones and engineering teams.

Git Workflow Overview

Workspace → Staging → Local Repository → Remote Repository 1. Workspace: Untracked files. 2. git init: Start tracking; move files to staging. 3. git add : Stage changes. 4. git commit -m "msg": Save version to local repo. 5. git push origin main: Push to remote repo. (main/master branch)

Common Commands

```
git status          # Show untracked, staged, and modified files
git log             # Show all commits and branches
git log --oneline   # Condensed commit view
git log --stat      # Show changes per commit
git log -p          # Show patch (diff) per commit
git diff            # Show differences between working tree and index
git checkout <hash> # Roll back to a specific commit
```

Branching in Git

```
- git branch <name>               # Create a new branch
- git branch                      # List all branches
- git log --oneline --all --graph # View commit graph across branches

Merging:
1. git checkout main            # Switch to main branch
2. git merge <branch>           # Merge feature branch into main
3. Resolve conflicts if any.

To branch from a past commit:
git branch <new_branch> <commit_hash>
```

Setting up Remote Repository

- git remote add origin # Link local to remote - git push -u origin main # Push initial commits and set upstream