

# DATA 3402 — Lab 3 Merge Conflict Resolution Guide

## Why This Happens

When both you and the instructor modify the same Jupyter notebook file, Git cannot automatically merge the JSON structure of the notebook. This creates a merge conflict, even if the changes appear simple.

## Goal

- Keep your completed Lab 3 work
- Keep your previously submitted labs
- Pull the latest lectures and updates from upstream
- Avoid corrupting the notebook file

## Step-by-Step Safe Fix (Recommended)

Step 1 – Backup Lab 3:

```
cp Labs/Lab.3/Lab.3.ipynb ~/Lab3_backup.ipynb
```

Step 2 – Checkout main:

```
git checkout main
```

Step 3 – Fetch upstream:

```
git fetch upstream
```

Step 4 – Reset main to instructor version:

```
git reset --hard upstream/main
```

Step 5 – Restore your Lab 3:

```
cp ~/Lab3_backup.ipynb Labs/Lab.3/Lab.3.ipynb
```

Step 6 – Commit your restored lab:

```
git add Labs/Lab.3/Lab.3.ipynb  
git commit -m "Restore Lab 3 after upstream sync"
```

Step 7 – Push to your fork:

```
git push origin main
```

---

## Important: Prevent Future Issues

1. Always keep main clean and synced with upstream.
2. Do each lab in a separate branch (e.g., git checkout -b lab3).
3. Never manually edit notebook merge markers.
4. Add this to .gitignore:  
.ipynb\_checkpoints/

If something goes wrong, run 'git status' and review which files are marked as conflicted before taking further action.