Setting up the Project Repository

☰ Tags	Module-3
≡ Class	2
📅 Created Date	@October 21, 2025
⊙ Course Name	MLOps with Cloud
☰ Related Topics	env-setup mlops-fundamentals modular-workflow project-setup
⊙ Resource Type	guides

Supporting Guide for Module-3 Video-2:

Setting Up an ML Project Repository and Environment:

In this guide, you'll set up a new machine learning project from scratch, including creating a Python virtual environment, initializing a Git repository, and connecting it to GitHub. You'll also learn about important project setup concepts like `.gitignore` files and environment management.

Prerequisites (as per Module-2)

- Python 3.8 or higher installed
- Git installed
- GitHub account created
- Basic command line knowledge

Checklist!

By the end of this guide, you should be able to:

1. Create and manage Python virtual environments
2. Initialize a Git repository
3. Create and understand `.gitignore` files
4. Connect local repositories to GitHub
5. Make your first commit and push

Step 1: Creating the Project Directory and Virtual Environment

```
# Create project directory
mkdir mlops-project # Or, your project name
cd mlops-project

# Create virtual environment
python3 -m venv venv

# Activate virtual environment
# On Windows:
venv\\Scripts\\activate
# On Unix or MacOS:
source venv/bin/activate
```

Understanding Virtual Environments

Virtual environments isolate project dependencies, preventing conflicts between different projects. When activated, packages will be installed only for this project.

Step 2: Git Repository Setup

Now, in the project directory,

Initialize Git Repository

```
git init
```

Creating .gitignore File

Create a new file named `.gitignore` with the following content:

```
# Virtual Environment
venv/
env/
ENV/

# Or, the name of your Virtual Environment
```

Understanding .gitignore

The `.gitignore` file tells Git which files and directories to ignore when tracking changes. This is crucial for:

1. **Security**: Prevents sensitive information (API keys, credentials) from being committed
2. **Efficiency**: Excludes large generated files and directories that can be recreated
3. **Cleanliness**: Keeps your repository clean from temporary files and build artifacts

Common items to ignore:

- **Virtual Environment Files** (`venv/`): These are large and environment-specific
- **Compiled Python Files** (`__pycache__/`): Generated during runtime
- **IDE Settings** (`.vscode/`, `.idea/`): Personal to each developer
- **Log Files** (`.log`): Usually generated during runtime
- **Jupyter Checkpoints** (`.ipynb_checkpoints`): Temporary Jupyter files (We'll also be using Jupyter Notebooks for this module!)

Now, let’s install the required libraries and packages with requirements.txt file inside the directory within the virtual environment,

Step 3: Creating the requirements.txt file

1. Create a file named requirements.txt in the directory
2. Add the following libraries in it,

```
numpy
pandas
```

3. Run the following command,

```
pip install -r requirements.txt
```

Step 4: Creating README.md file

Create a `README.md` file with the following content:

```
# MWC-Module-3-Modular-Workflow-and-Project-Setup-Basics

# Problem Statement:

## Business Context:


The project aims to develop a machine learning system that predicts individual income levels based on demographic and employment data.

The prediction boundary is set at $50,000 annually (binary classification problem).

The solution will help in understanding socio-economic factors affecting income levels.
```

~~~~
Enable data-driven decision making for policy makers and financial institutions.

Identify key socio-economic factors influencing income disparities.

Support targeted intervention programs for economic development

## Key Stakeholders

**Policy Makers:** For evidence-based policy development

**Financial Institutions:** For risk assessment and product development

**Social Services:** For resource allocation and program planning

**Research Organizations:** For socio-economic studies

# Dataset Details:
Let's visualize the data structure and features:

```mermaid
classDiagram
    class Features {
        Demographic_Features
        Employment_Features
        Financial_Features
        Other_Features
    }

    class Demographic_Features {
        age: numeric
        education: categorical
        education-num: numeric
        race: categorical
        sex: categorical
~~~~

~~~
        country: categorical
    }

    class Employment_Features {
        workclass: categorical
        occupation: categorical
        hours-per-week: numeric
        relationship: categorical
        marital-status: categorical
    }

    class Financial_Features {
        fnlwgt: numeric
        capital-gain: numeric
        capital-loss: numeric
    }

    Features --> Demographic_Features
    Features --> Employment_Features
    Features --> Financial_Features
```
~~~

Step 5: Initial Commit and GitHub Setup

```
# Add files to git
git add .gitignore README.md requirements.txt

# Make initial commit
git commit -m "Initial project setup with README and requirements"

# Create new repository on GitHub (do this through GitHub's website)
# Then link your local repository (replace YOUR_USERNAME and REPO_NAME)
```

```
git remote add origin <https://github.com/YOUR_USERNAME/REPO_NAME.git>

# Push to GitHub
git push -u origin main
```

Step 6: Verify Setup

1. Check that your virtual environment is active (you should see `(venv)` in your terminal)
2. Verify that Git is tracking your files:

```
git status
```

3. **Visit your GitHub repository to ensure files were pushed successfully**

Additional Resources

- **Python Virtual Environments Documentation**

12. Virtual Environments and Packages

Introduction: Python applications will often use packages and modules that don’t come as part of the standard library. Applications will sometimes need a specific version of a library,

https://docs.python.org/3/tutorial/venv.html

- **Git Documentation**

Git - Documentation

The official and comprehensive man pages that are included in the Git package itself.

https://git-scm.com/doc

- **GitHub Guides**

GitHub.com Help Documentation

Get started, troubleshoot, and make the most of GitHub. Documentation for new users, developers, administrators, and all of GitHub's products.

https://guides.github.com/