Download code from

https://scikit-learn.org/stable/_downloads/1a55101a8e49ab5d3213dadb31332045/plot_digits_classification.py

– Run:

```
wget https://scikit-learn.org/stable/_downloads/1a55101a8e49ab5d3213dadb31332045/plot_digits_classification.py
```

Run: python plot_digits_classification.py

Option 2:-

```
python3 plot_digits_classification.py
```

Is there any issue?
– Create a new Conda Environment

```
conda create -n digit python=3.13
conda activate digit
```

– Create requirements.txt to include missing packages
Mention the packages names

– Install all packages in requirements.txt

Option 1 :-

```
pip install matplotlib
pip install scikit-learn
pip install pandas numpy
```

Option 2:-

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

Run: python plot_digits_classification.py

– Create an empty repo on github.com named "DigitClassification"
git clone "SSH Link "

- Open file nano " file name "
After that update the file

– Clone the repo into your local system (clone via ssh not https)

– Move plot_digits_classification.py and requirements.txt into the repo folder

```
mv requirements.txt DigitClassification/
mv plot_digits_classification.py DigitClassification/
```

Option 2:-

```
mv ~/Downloads/"plot_digits_classification .py" ./plot_digits_classification.py
```

Push the changes into github
– git add .
– git commit -m “Added code and requirements”
(Optional - incase git push directly after commit doesnot work)
– git remote set-url origin git@github.com:DhirajRajIitJodhpur/DigitClassification.git
Replace username with your github username
– git push
Verify the changes on github.com

Is there anything missing? – ADD a README file describing how to install packages and run the code

Task: Repeat for a new task and new repo for the following code

regression.py

```
====================
import numpy as np
import statsmodels.api as sm
# Sample data
X = np.array([1, 2, 3, 4, 5, 6, 7, 8])
y = np.array([2, 4, 5, 4, 5, 6, 5, 6])
# Add constant for intercept term
X = sm.add_constant(X) # adds a column of ones
# Fit the model
model = sm.OLS(y, X).fit()
# View the summary
print(model.summary())
# Predict
y_pred = model.predict(X)
print("Predictions:", y_pred)
=====================
```

1. Setting Up the Repository

Clone the repository from GitHub:

```
git clone <repo-url>
```

The repository will have a default branch (usually 'main') connected to the remote 'origin'.

2. Creating a New Branch

Create and switch to a new branch:

```
git checkout -b branch1
```

At this point, the new branch only exists locally. If you try 'git push' directly, Git will complain that there is no upstream branch.

3. Setting the Upstream Branch

An upstream branch is the remote branch (e.g., origin/branch1) that your local branch is connected to. This allows 'git push' and 'git pull' to work without specifying the remote and branch each time.

Set upstream and push the branch for the first time:

```
git push -u origin branch1
```

4. Modifying Files and Updating Branch

Edit README.md or any file, then run:

```
git add README.md
```

```
git status
```

```
git commit -m "updated readme in branch1"
```

```
git push
```

5. Syncing with Remote Changes

If changes were made directly on GitHub (e.g., creating README.md in main), you need to pull them into your local branch.

```
git fetch origin

git merge origin/main
```

If you run 'git pull' and see the error 'Need to specify how to reconcile divergent branches', it means Git wants you to decide how to handle differences between local and remote. Options:

```
git pull --no-rebase    # merge (default)

git pull --rebase       # rebase for linear history

git pull --ff-only      # only if fast-forward is possible
```

6. Merging Branch into Main

When ready, merge branch1 into main:

```
git checkout main

git merge branch1

git push
```

7. Deleting a Branch

Delete locally:

```
git branch -d branch1
```

Delete on remote:

```
git push origin --delete branch1
```

8. Key Concepts Recap

- origin → alias for your remote repository (on GitHub).

- upstream branch → the remote branch your local branch is tracking (e.g., origin/main).

- git push -u origin branch1 → pushes and sets upstream branch so future git push/pull works without extra args.

- git pull → combines fetch + merge (or rebase). If local and remote diverge, Git asks how to reconcile.