Homework 0: Machine Learning Tools
CPE 486/586

Instructor: Rahul Bhadani

Due: Aug 25, 2025, 11:59 PM
110 points

You are allowed to use a generative model-based AI tool for your assignment. However, you must submit an accompanying reflection report detailing how you used the AI tool, the specific query you made, and how it improved your understanding of the subject. You are also required to submit screenshots of your conversation with any large language model (LLM) or equivalent conversational AI, clearly showing the prompts and your login avatar. Some conversational AIs provide a way to share a conversation link, and such a link is desirable for authenticity. Failure to do so may result in actions taken in compliance with the plagiarism policy.

Additionally, you must include your thoughts on how you would approach the assignment if such a tool were not available. Failure to provide a reflection report for every assignment where an AI tool is used may result in a penalty, and subsequent actions will be taken in line with the plagiarism policy.

Submission instruction:

Upload a .pdf on Canvas with the format `CPE486586-LastFirst-HW-XX`. For example, if your name is Sam Wells, your file name should be `CPE486586-LastFirst-HW-XX.pdf`. If there is a programming assignment, then you should include your source code along with your PDF files in a zip file `CPE486586-LastFirst-HW-XX.zip`. Your submission must contain your name and UAH Charger ID or your UAH email address. Please number your pages as well.

All CPE 586 students are required to submit their response in a Latex-generated PDF.

---

1 Working with GitHub (25 points)

Start your screen recorder, either using Zoom or anything equivalent, and capture the following tasks:

1. Create a new GitHub account with a random ID name using English keywords, e.g **orange-unicorn** followed by gh. For example, if your keyword is **orangeunicorn**, then your Github

---

ID should **orangeunicorngh**. Make sure that **no** offensive or NSFW keywords should be used. You should report your unique keyword in the submitted PDF. <u>Do not</u> share this unique keyword with your classmates.   **(2 Points)**

2. Install Git on your computer if it's not already installed.

3. Configure Git with your name and email. You do not need to use your UAH email for this purpose. I recommend creating a new Gmail account with the same username **orangeunicorn@gmail.com**. You will use this unique keyword for the entirety of the course. Replace **orangeunicorn** with your unique keyword.   **(4 Points)**

```
git config --global user.name "orangeunicorn"
git config --global user.email "orangeunicorn@gmail.com"
```

Replace the values as per your preference.

4. Create a new repository on GitHub named **hw00orangeunicorn**. Make sure that the repository you have created is public.   **(2 Points)**

5. Clone it locally to your computer: **(2 Points)**

```
git clone https://github.com/orangeunicorngh/hw00orangeunicorn
```

Place a `hello.py` file in `hw00orangeunicorn` folder with a text `print("Hello!)"`.

You should replace the repo URL accordingly. Change your current working directory to `hw00orangeunicorn` folder using the command line.   **(2 Points)**

6. Create a new branch named "hw00":   **(2 Points)**

```
git checkout -b hw00
```

7. Add your `hello.py` file to this branch.   **(2 Points)**

```
git add hello.py
git commit -m "Add hw 00 hello.py"
```

8. Create a file `.gitignore`. What purpose does `.gitignore` serve?   **(4 Points)**

Add following content to the `.gitignore` file:

```
# --------------------------
# PYTHON
#---------------------------
.ipynb_checkpoints/
dist/
*.egg-info/
## ----------------------------
# Visual Studio Code
#----------------------------
.vscode/
```

Explain what would adding the above content to the `.gitignore` file do?

Next, add `.gitignore` file to git and commit it:

```
git add .gitignore
git commit -m "Added gitignore file"
```

9. Push your branch to GitHub: **(2 Points)**

```
git push origin hw00
```

or you might need to do

```
git push -set-upstream origin hw00
```

Finally, as a part of your submission, upload your screen-recorded video to YouTube with visibility set to *unlisted* or *public*, and provide its URL in the submission.   **(3 Points)**

2 Working with UV (45 points)

A part of this assignment asks you to install **UV** and create a Python package. It is recommended that you read the documentation at https://docs.astral.sh/uv/. Start your screen recording.

Install UV in your machine (or alternatively, you can install it in your Blackhawk server account).

1. Linux/Mac/WSL on Windows: **(2 Points)**

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

2. Create a package (replace **orangeunicorn** with your unique keyword.) **(2 Points)**

```
# Create a uv project
uv init --lib --package orangeunicorn --build-backend maturin --author-from git
cd orangeunicorn
# install python
uv python install 3.12
# create a virtual environment
uv venv --python 3.12
# activate virtual environment
source .venv/bin/activate
```

3. Let's look at the structure of the package: **(5 Points)**

```
|-- Cargo.toml
|-- pyproject.toml
|-- README.md
\|-- src
 |-- lib.rs
 \|-- myawesomepackage
 |-- _core.pyi
 |-- __init__.py
 |-- py.typed
```

Explain what each of those file mean and what their purpose is.

4. Create a subpackage called **differential**. **(2 Points)**

5. In subpackage, create a module named `discrete.py` and implement a function `diff` in Python for computing a discrete derivative of a timeseries data as follows. Consider a timeseries $x(t)$. The discrete derivative $v(t)$ of the timeseries is given by

$$v(t) = \frac{x(t_k) - x(t_{k-1})}{t_k - t_{k-1}}.$$

For this purpose, assume that the function `diff` takes two Python arrays as input (do not use Numpy Array) of equal length: (i) The first array should be the time value t_k, k is the index of time; (ii) The second array should be the signal value x. The function should be able to check for the equality of lengths. The function should return the computed discrete derivative $v(t)$ as a Python array. Make sure to add an inline comment to the function as shown in the example in the code snippet included in the Lecture slides. **(10 Points)**

It is your responsibility to make sure that the function is *importable*. See Lecture Slides 2 for the reference.

6. Build your UV package using   **(2 Points)**

```
uv build
```

7. Commit your UV package to your GitHub account created earlier. You must provide all the necessary commands you executed to commit the package to GitHub.   **(2 Points)**

8. Add .whl file generated in `dist` folder as a result of `uv build` as a release to your GitHub repo.   **(2 Points)**

9. Create another UV project for testing your package: **(2 Points)**

```
uv init difftest
cd difftest
uv venv --python 3.12
source .venv/bin/activate
uv add <enter the URL of your .whl release file>
```

10. Prepare your UV Virtual Environment for Jupyter Notebook.   **(2 Points)**

```
cd difftest
source .venv/bin/activate
uv add --dev jupyter
uv add --dev ipykernel
# create a kernel for jupyter
uv run ipython kernel install --user --env VIRTUAL_ENV $(pwd)/.venv \
 --name=difftest

# Launch Jupyter Notebook
jupyter notebook
```

11. Create a new ipynb file with kernel set to `difftest`.   **(2 Points)**

12. Import your UV package `orangeunicorn` (Change this name to your chosen keyword) and its required module to test `diff` function. Take the test data as   **(2 Points)**

```
t = [0, 0.1, 0.3, 0.4, 0.55, 0.67, 0.71]
x = [23.1, 22.5, 23.5, 21.88, 22.5, 23.5, 24.88]
```

13. Create two line plots with markers : (i) x vs t, (ii) v vs t, where v is the discrete derivative. Save the plot as a PDF and include it in your submission report.   **(5 Points)**

    Provide the YouTube Video URL for this part of your assignment in the submission.   **(3 Points)**

3 Install Visual Studio Code and Important Extensions (15 Points)

Note: The following assignment is meant for macOS or Ubuntu Linux. If you have a Windows OS, please use WSL.

Start your screen recorder, either using Zoom or anything equivalent, and capture the following tasks:

1. Download and install Visual Studio Code.   **(2 Points)**

2. Go to your test UV environment `difftest`, activate the virtual environment there, and install the following packages: **(2 Points)**

   ```
   uv pip install pandas matplotlib torch seaborn scikit-learn pygame
   ```

3. Open Visual Studio Code by typing   **(2 Points)**

   ```
   code
   ```

   in the command line,

   and click on the Extensions icon in the left sidebar (or press Ctrl+Shift+X).

4. Search for and install the following extensions: **(2 Points)**

   (a) Jupyter (Microsoft)

   (b) Jupyter Notebook Renderers (Microsoft)

   (c) Python (Microsoft)

   (d) Pylance (Microsoft)

   (e) Python Debugger (Microsoft)

   (f) Python Environments (Microsoft)

   (g) Remote – SSH (Microsoft)

   (h) Remote – SSH: Editing Configuration Files (Microsoft)

   (i) Remote Explorer (Microsoft)

5. Create a new file and save it as `vscodehw00.ipynb`. On the top right corner, click **Select Kernel** and choose the appropriate kernel you created earlier.   **(2 Points)**

6. Enter the following code into your notebook and execute it

```python
import torch
import numpy as np
import matplotlib.pyplot as plt


print("Hello, Jupyter!")
```

Make sure the notebook is executed successfully.   **(2 Points)**

Provide the YouTube Video URL for this part of your assignment in the submission. **(3 Points)**

4 Working with Google Colab (25 Points)

Start your screen recorder, either using Zoom or anything equivalent, and capture the following tasks:

Create a Google Colab notebook on https://colab.research.google.com/, name it `hw00.ipynb`.

1. Create a markdown cell, and enter your name and email address you created earlier.   **(2 Points)**

2. Install some Python packages as follows by creating a new code cell and executing:

```
%reload_ext autoreload
%autoreload 2
! pip install pandas matplotlib torch seaborn scikit-learn pygame
```

Please describe, what does   **(2 Points)**

```
%reload_ext autoreload
%autoreload 2
```

3. Next, create another new code cell, and execute the following   **(2 Points)**

```python
import torch
import numpy as np
```

```python
import matplotlib.pyplot as plt


print("Hello, World!")
```

4. Create an account in https://pypi.org/ with the unique keyword as your username. This will prepare you to upload a Python package to be installed using the `pip install` command. **(2 Points)**

5. Go back to your UV Project `orangeunicorn` (replace the project name with your chosen keyword) and upload the built package to PYPI using **(5 Points)**

```
uv add twine
twine upload dist/*
```

Here, you may encounter two issues: (i) how to generate a Token as a password, (ii) naming issues with your .whl or .tar.gz file. You must figure out how to solve this as part of your assignment.

6. Go back to the Google Colab notebook.

   In one of the cells, enter.

```
!pip install orangeunicorn
```

   Restart the notebook Kernel. Test your function `diff` with the test data provided earlier and create two plots. Provide a publicly accessible URL of your Notebook as a part of your submission. **(5 Points)**

Provide a public URL to your Google Colab notebook. **(2 Points)**

Further, upload your screen-recorded video to YouTube with visibility set to *unlisted* or *public*, and provide its URL in the submission. **(5 Points)**