

LLM Engineering - Master AI and LLMs

Setup instructions for Windows

Welcome, PC people!

I should confess up-front: setting up a powerful environment to work at the forefront of AI is not as simple as I'd like. For most people these instructions will go great; but in some cases, for whatever reason, you'll hit a problem. Please don't hesitate to reach out - I am here to get you up and running quickly. There's nothing worse than feeling *stuck*. Message me, email me or LinkedIn message me and I will unstick you quickly!

Email: ed@edwarddonner.com

LinkedIn: <https://www.linkedin.com/in/eddonner/>

I use a platform called Anaconda to set up your environment. It's a powerful tool that builds a complete science environment. Anaconda ensures that you're working with the right version of Python and all your packages are compatible with mine, even if our systems are completely different. It takes more time to set up, and it uses more hard drive space (5+ GB) but it's very reliable once its working.

Having said that: if you have any problems with Anaconda, I've provided an alternative approach. It's faster and simpler and should have you running quickly, with less of a guarantee around compatibility.

Before we begin - Heads up!

If you are relatively new to using the Command Prompt, here is an excellent guide with instructions and exercises. I'd suggest you work through this first to build some confidence.

There are 4 common gotchas to developing on Windows to be aware of:

1. Permissions. Please take a look at this tutorial on permissions on Windows
2. Anti-virus, Firewall, VPN. These can interfere with installations and network access; try temporarily disabling them as needed
3. The evil Windows 260 character limit to filenames - here is a full explanation and fix!
4. If you've not worked with Data Science packages on your computer before, you might need to install Microsoft Build Tools. Here are instructions.

Part 1: Clone the Repo

This gets you a local copy of the code on your box.

1. **Install Git** (if not already installed):

- Download Git from <https://git-scm.com/download/win>
- Run the installer and follow the prompts, using default options (press OK lots of times!)

2. **Open Command Prompt:**

- Press Win + R, type `cmd`, and press Enter

3. **Navigate to your projects folder:**

If you have a specific folder for projects, navigate to it using the `cd` command. For example:

```
cd C:\Users\YourUsername\Documents\Projects
```

Replacing `YourUsername` with your actual Windows user

If you don't have a projects folder, you can create one:

```
mkdir C:\Users\YourUsername\Documents\Projects
cd C:\Users\YourUsername\Documents\Projects
```

4. **Clone the repository:**

Enter this in the command prompt in the Projects folder:

```
git clone https://github.com/ed-donner/llm_engineering.git
```

This creates a new directory `llm_engineering` within your Projects folder and downloads the code for the class. Do `cd llm_engineering` to go into it. This `llm_engineering` directory is known as the “project root directory”.

Part 2: Install Anaconda environment

If this Part 2 gives you any problems, there is an alternative Part 2B below that can be used instead.

1. **Install Anaconda:**

- Download Anaconda from <https://docs.anaconda.com/anaconda/install/windows/>
- Run the installer and follow the prompts. Note that it takes up several GB and take a while to install, but it will be a powerful platform for you to use in the future.

2. **Set up the environment:**

- Open **Anaconda Prompt** (search for it in the Start menu)
- Navigate to the “project root directory” by entering something like `cd C:\Users\YourUsername\Documents\Projects\llm_engineering` using the actual path to your `llm_engineering` project root directory. Do a `dir` and check you can see subdirectories for each week of the course.
- Create the environment: `conda env create -f environment.yml`

- Wait for a few minutes for all packages to be installed - in some cases, this can literally take 20-30 minutes if you've not used Anaconda before, and even longer depending on your internet connection. Important stuff is happening! If this runs for more than 1 hour 15 mins, or gives you other problems, please go to Part 2B instead.
- You have now built an isolated, dedicated AI environment for engineering LLMs, running vector datastores, and so much more! You now need to **activate** it using this command: `conda activate llms`

You should see (llms) in your prompt, which indicates you've activated your new environment.

3. Start Jupyter Lab:

- In the Anaconda Prompt, from within the `llm_engineering` folder, type: `jupyter lab`

... and Jupyter Lab should open up in a browser. If you've not seen Jupyter Lab before, I'll explain it in a moment! Now close the jupyter lab browser tab, and close the Anaconda prompt, and move on to Part 3.

Part 2B - Alternative to Part 2 if Anaconda gives you trouble

1. Open Command Prompt

Press Win + R, type `cmd`, and press Enter

Run `python --version` to find out which python you're on.

Ideally you'd be using a version of Python 3.11, so we're completely in sync.

I believe Python 3.12 works also, but (as of Feb 2025) Python 3.13 does **not** yet work as several Data Science dependencies are not yet ready for Python 3.13.

If you need to install Python or install another version, you can download it here:

<https://www.python.org/downloads/>

2. Navigate to the "project root directory" by entering something like `cd C:\Users\YourUsername\Documents\Projects\llm_engineering` using the actual path to your `llm_engineering` project root directory. Do a `dir` and check you can see subdirectories for each week of the course.

Then, create a new virtual environment with this command:

```
python -m venv llms
```

3. Activate the virtual environment with `llms\Scripts\activate` You should see (llms) in your command prompt, which is your sign that things are going well.
4. Run `python -m pip install --upgrade pip` followed by `pip install -r requirements.txt`
This may take a few minutes to install.
If you see an error like this:

Microsoft Visual C++ 14.0 or greater is required. Get it with “Microsoft C++ Build Tools”: <https://visualstudio.microsoft.com/visual-cpp-build-tools/>

Then please follow the link and install Microsoft C++ Build Tools.

In the very unlikely event that this step doesn’t go well, you should try the bullet-proof (but slower) version:

```
pip install --retries 5 --timeout 15 --no-cache-dir --force-reinstall  
-r requirements.txt
```

6. Start Jupyter Lab:

From within the `llm_engineering` folder, type: `jupyter lab`
... and Jupyter Lab should open up, ready for you to get started. Open the `week1` folder and double click on `day1.ipynb`. Success! Now close down jupyter lab and move on to Part 3.

If there are any problems, contact me!

Part 3 - OpenAI key (OPTIONAL but recommended)

Particularly during weeks 1 and 2 of the course, you’ll be writing code to call the APIs of Frontier models (models at the forefront of AI).

For week 1, you’ll only need OpenAI, and you can add the others if you wish later on.

1. Create an OpenAI account if you don’t have one by visiting:
<https://platform.openai.com/>
2. OpenAI asks for a minimum credit to use the API. For me in the US, it’s \$5. The API calls will spend against this \$5. On this course, we’ll only use a small portion of this. I do recommend you make the investment as you’ll be able to put it to excellent use. But if you’d prefer not to pay for the API, I give you an alternative in the course using Ollama.

You can add your credit balance to OpenAI at Settings > Billing:
<https://platform.openai.com/settings/organization/billing/overview>

I recommend you disable the automatic recharge!

3. Create your API key

The webpage where you set up your OpenAI key is at <https://platform.openai.com/api-keys> - press the green ‘Create new secret key’ button and press ‘Create secret key’. Keep a record of the API key somewhere private; you won’t be able to retrieve it from the OpenAI screens in the future. It should start `sk-proj-`.

In week 2 we will also set up keys for Anthropic and Google, which you can do here when we get there.

- Claude API at <https://console.anthropic.com/> from Anthropic - Gemini API at <https://ai.google.dev/gemini-api> from Google

Later in the course you'll be using the fabulous HuggingFace platform; an account is available for free at <https://huggingface.co> - you can create an API token from the Avatar menu » Settings » Access Tokens.

And in Week 6/7 you'll be using the terrific Weights & Biases at <https://wandb.ai> to watch over your training batches. Accounts are also free, and you can set up a token in a similar way.

PART 4 - .env file

When you have these keys, please create a new file called `.env` in your project root directory. The filename needs to be exactly the four characters `“.env”` rather than `“my-keys.env”` or `“.env.txt”`. Here's how to do it:

1. Open the Notepad (Windows + R to open the Run box, enter `notepad`)
2. In the Notepad, type this, replacing `xxxx` with your API key (starting `sk-proj-`).

```
OPENAI_API_KEY=xxxx
```

If you have other keys, you can add them too, or come back to this in future weeks:

```
GOOGLE_API_KEY=xxxx
ANTHROPIC_API_KEY=xxxx
DEEPSEEK_API_KEY=xxxx
HF_TOKEN=xxxx
```

Double check there are no spaces before or after the `=` sign, and no spaces at the end of the key.

3. Go to File > Save As. In the “Save as type” dropdown, select All Files. In the “File name” field, type exactly `.env` as the filename. Choose to save this in the project root directory (the folder called `llm_engineering`) and click Save.
4. Navigate to the folder where you saved the file in Explorer and ensure it was saved as `“.env”` not `“.env.txt”` - if necessary rename it to `“.env”` - you might need to ensure that “Show file extensions” is set to “On” so that you see the file extensions. Message or email me if that doesn't make sense!

This file won't appear in Jupyter Lab because jupyter hides files starting with a dot. This file is listed in the `.gitignore` file, so it won't get checked in and your keys stay safe.

Part 5 - Showtime!!

- Open **Anaconda Prompt** (search for it in the Start menu) if you used Anaconda, otherwise open a Powershell if you used the alternative approach in Part 2B
- Navigate to the “project root directory” by entering something like `cd C:\Users\YourUsername\Documents\Projects\llm_engineering` using the actual path to your `llm_engineering` project root directory. Do a `dir` and check you can see subdirectories for each week of the course.
- Activate your environment with `conda activate llms` if you used Anaconda or `llms\Scripts\activate` if you used the alternative approach in Part 2B
- You should see `(llms)` in your prompt which is your sign that all is well. And now, type: `jupyter lab` and Jupyter Lab should open up, ready for you to get started. Open the `week1` folder and double click on `day1.ipynb`.

And you’re off to the races!

Note that any time you start `jupyter lab` in the future, you’ll need to follow these Part 5 instructions to start it from within the `llm_engineering` directory with the `llms` environment activated.

For those new to Jupyter Lab / Jupyter Notebook, it’s a delightful Data Science environment where you can simply hit `shift+return` in any cell to run it; start at the top and work your way down! There’s a notebook in the `week1` folder with a Guide to Jupyter Lab, and an Intermediate Python tutorial, if that would be helpful. When we move to Google Colab in Week 3, you’ll experience the same interface for Python runtimes in the cloud.

If you have any problems, I’ve included a notebook in `week1` called `troubleshooting.ipynb` to figure it out.

Please do message me or email me at ed@edwarddonner.com if this doesn’t work or if I can help with anything. I can’t wait to hear how you get on.