troubleshooting

January 6, 2025

1 Oh dear!

If you've got here, then you're still having problems setting up your environment. I'm so sorry! Hang in there and we should have you up and running in no time.

Setting up a Data Science environment can be challenging because there's a lot going on under the hood. But we will get there.

And please remember - I'm standing by to help out. Message me or email ed@edwarddonner.com and I'll get on the case. The very last cell in this notebook has some diagnostics that will help me figure out what's happening.

2 Before we begin

2.1 Checking your internet connection

First let's check that there's no VPN or Firewall or Certs problem.

Click in the cell below and press Shift+Return to run it.
If this gives you problems, then please try working through these instructions to address:
https://chatgpt.com/share/676e6e3b-db44-8012-abaa-b3cf62c83eb3

[1]:
```
import urllib.request

try:
    response = urllib.request.urlopen("https://www.google.com", timeout=10)
    if response.status != 200:
        print("Unable to reach google - there may be issues with your internet /
 ↪ VPN / firewall?")
    else:
        print("Connected to the internet and can reach Google")
except Exception as e:
    print(f"Failed to connect with this error: {e}")
```

```
Connected to the internet and can reach Google
```

3 Step 1

Try running the next 2 cells (click in the cell under this one and hit shift+return, then shift+return again).

If this gives an error, then you're likely not running in an "activated" environment. Please check back in Part 5 of the SETUP guide for PC or Mac for setting up the Anaconda (or virtualenv) environment and activating it, before running `jupyter lab`.

If you look in the Anaconda prompt (PC) or the Terminal (Mac), you should see `(llms)` in your prompt where you launch `jupyter lab` - that's your clue that the llms environment is activated.

If you are in an activated environment, the next thing to try is to restart everything: 1. Close down all Jupyter windows, like this one 2. Exit all command prompts / Terminals / Anaconda 3. Repeat Part 5 from the SETUP instructions to begin a new activated environment and launch `jupyter lab` from the `llm_engineering` directory
4. Come back to this notebook, and do Kernel menu » Restart Kernel and Clear Outputs of All Cells 5. Try the cell below again.

If **that** doesn't work, then please contact me! I'll respond quickly, and we'll figure it out. Please run the diagnostics (last cell in this notebook) so I can debug. If you used Anaconda, it might be that for some reason your environment is corrupted, in which case the simplest fix is to use the virtualenv approach instead (Part 2B in the setup guides).

[2]:
```
# Some quick checks that your Conda environment or VirtualEnv is as expected
# The Environment Name should be: llms

import os
conda_name, venv_name = "", ""

conda_prefix = os.environ.get('CONDA_PREFIX')
if conda_prefix:
    print("Anaconda environment is active:")
    print(f"Environment Path: {conda_prefix}")
    conda_name = os.path.basename(conda_prefix)
    print(f"Environment Name: {conda_name}")

virtual_env = os.environ.get('VIRTUAL_ENV')
if virtual_env:
    print("Virtualenv is active:")
    print(f"Environment Path: {virtual_env}")
    venv_name = os.path.basename(virtual_env)
    print(f"Environment Name: {venv_name}")

if conda_name != "llms" and virtual_env != "llms":
    print("Neither Anaconda nor Virtualenv seem to be activated with the␣
 ↪expected name 'llms'")
    print("Did you run 'jupyter lab' from an activated environment with (llms)␣
 ↪showing on the command line?")
    print("If in doubt, close down all jupyter lab, and follow Part 5 in the␣
 ↪SETUP-PC or SETUP-mac guide.")
```

```
Anaconda environment is active:
Environment Path: /Users/markku/miniconda3/envs/udemy-llm-engineering
```

```
Environment Name: udemy-llm-engineering
Neither Anaconda nor Virtualenv seem to be activated with the expected name
'llms'
Did you run 'jupyter lab' from an activated environment with (llms) showing on
the command line?
If in doubt, close down all jupyter lab, and follow Part 5 in the SETUP-PC or
SETUP-mac guide.
```

[3]:
```
# And now, this should run with no output - no import errors.
# Import errors might indicate that you started jupyter lab without your␣
 ↪environment activated? See SETUP part 5.
# Or you might need to restart your Kernel and Jupyter Lab.
# Or it's possible that something is wrong with Anaconda. Please try SETUP Part␣
 ↪2B, the alternative to Part 2.
# If you're unsure, please run the diagnostics (last cell in this notebook) and␣
 ↪then email me at ed@edwarddonner.com

from openai import OpenAI
```

4 Step 2

Let's check your .env file exists and has the OpenAI key set properly inside it.
Please run this code and check that it prints a successful message, otherwise follow its instructions.

If it isn't successful, then it's not able to find a file called `.env` in the `llm_engineering` folder.
The name of the file must be exactly `.env` - it won't work if it's called `my-keys.env` or `.env.doc`.
Is it possible that `.env` is actually called `.env.txt`? In Windows, you may need to change a setting in the File Explorer to ensure that file extensions are showing ("Show file extensions" set to "On").
You should also see file extensions if you type `dir` in the `llm_engineering` directory.

Nasty gotchas to watch out for:
- In the .env file, there should be no space between the equals sign and the key. Like: `OPENAI_API_KEY=sk-proj-...` - If you copied and pasted your API key from another application, make sure that it didn't replace hyphens in your key with long dashes

Note that the `.env` file won't show up in your Jupyter Lab file browser, because Jupyter hides files that start with a dot for your security; they're considered hidden files. If you need to change the name, you'll need to use a command terminal or File Explorer (PC) / Finder Window (Mac). Ask ChatGPT if that's giving you problems, or email me!

If you're having challenges creating the `.env` file, we can also do it with code! See the cell after the next one.

It's important to launch `jupyter lab` from the project root directory, `llm_engineering`. If you didn't do that, this cell might give you problems.

[4]:
```
from pathlib import Path

parent_dir = Path("..")
```

```
env_path = parent_dir / ".env"

if env_path.exists() and env_path.is_file():
    print(".env file found.")

    # Read the contents of the .env file
    with env_path.open("r") as env_file:
        contents = env_file.readlines()

    key_exists = any(line.startswith("OPENAI_API_KEY=") for line in contents)
    good_key = any(line.startswith("OPENAI_API_KEY=sk-proj-") for line in␣
 ↪contents)

    if key_exists and good_key:
        print("SUCCESS! OPENAI_API_KEY found and it has the right prefix")
    elif key_exists:
        print("Found an OPENAI_API_KEY although it didn't have the expected␣
 ↪prefix sk-proj- \nPlease double check your key in the file..")
    else:
        print("Didn't find an OPENAI_API_KEY in the .env file")
else:
    print(".env file not found in the llm_engineering directory. It needs to␣
 ↪have exactly the name: .env")

    possible_misnamed_files = list(parent_dir.glob("*.env*"))

    if possible_misnamed_files:
        print("\nWarning: No '.env' file found, but the following files were␣
 ↪found in the llm_engineering directory that contain '.env' in the name.␣
 ↪Perhaps this needs to be renamed?")
        for file in possible_misnamed_files:
            print(file.name)
```

```
.env file found.
SUCCESS! OPENAI_API_KEY found and it has the right prefix
```

4.1 Fallback plan - python code to create the .env file for you

Only run the next cell if you're having problems making the .env file.
Replace the text in the first line of code with your key from OpenAI.

[5]:
```
# Only run this code in this cell if you want to have a .env file created for␣
 ↪you!

# Put your key inside the quote marks
make_me_a_file_with_this_key = "put your key here inside these quotes.. it␣
 ↪should start sk-proj-"
```

```
# Change this to True if you already have a .env file and you want me to␣
 ↪replace it
overwrite_if_already_exists = False

from pathlib import Path

parent_dir = Path("..")
env_path = parent_dir / ".env"

if env_path.exists() and not overwrite_if_already_exists:
    print("There is already a .env file - if you want me to create a new one,␣
 ↪change the variable overwrite_if_already_exists to True above")
else:
    try:
        with env_path.open(mode='w', encoding='utf-8') as env_file:
            env_file.write(f"OPENAI_API_KEY={make_me_a_file_with_this_key}")
        print(f"Successfully created the .env file at {env_path}")
        if not make_me_a_file_with_this_key.startswith("sk-proj-"):
            print(f"The key that you provided started with␣
 ↪'{make_me_a_file_with_this_key[:8]}' which is different to sk-proj- is that␣
 ↪what you intended?")
        print("Now rerun the previous cell to confirm that the file is created␣
 ↪and the key is correct.")
    except Exception as e:
        print(f"An error occurred while creating the .env file: {e}")
```

```
There is already a .env file - if you want me to create a new one, change the
variable overwrite_if_already_exists to True above
```

5 Step 3

Now let's check that your API key is correct set up in your `.env` file, and available using the dotenv package. Try running the next cell.

[9]:
```
# This should print your API key to the output - please follow the instructions␣
 ↪that get printed

import os
from dotenv import load_dotenv
load_dotenv(override=True)

api_key = os.getenv("OPENAI_API_KEY")

if not api_key:
    print("No API key was found - please try Kernel menu >> Restart Kernel And␣
 ↪Clear Outputs of All Cells")
```

```
elif not api_key.startswith("sk-proj-"):
    print(f"An API key was found, but it starts with {api_key[:8]} rather than␣
 ↪sk-proj- please double check this is as expected.")
elif api_key.strip() != api_key:
    print("An API key was found, but it looks like it might have space or tab␣
 ↪characters at the start or end - please remove them")
else:
    print("API key found and looks good so far!")

if api_key:
    problematic_unicode_chars = ['\u2013', '\u2014', '\u201c', '\u201d',␣
 ↪'\u2026', '\u2018', '\u2019']
    forbidden_chars = ["'", " ", "\n", "\r", '"']

    if not all(32 <= ord(char) <= 126 for char in api_key):
        print("Potential problem: there might be unprintable characters␣
 ↪accidentally included in the key?")
    elif any(char in api_key for char in problematic_unicode_chars):
        print("Potential problem: there might be special characters, like long␣
 ↪hyphens or curly quotes in the key - did you copy it via a word processor?")
    elif any(char in api_key for char in forbidden_chars):
        print("Potential problem: there are quote marks, spaces or empty lines␣
 ↪in your key?")
    else:
        print("The API key contains valid characters")

print(f"\nHere is the key --> {api_key} <--")
print()
print("If this key looks good, please go to the Edit menu >> Clear Cell Output␣
 ↪so that your key is no longer displayed here!")
```

```
API key found and looks good so far!
The API key contains valid characters

If this key looks good, please go to the Edit menu >> Clear Cell Output so that
your key is no longer displayed here!
```

5.1 It should print some checks including something like:

```
Here is the key --> sk-proj-blahblahblah <--
```

If it didn't print a key, then hopefully it's given you enough information to figure this out. Or contact me!

There is a final fallback approach if you wish: you can avoid using .env files altogether, and simply always provide your API key manually.
Whenever you see this in the code:
`openai = OpenAI()`
You can replace it with:

```
openai = OpenAI(api_key="sk-proj-xxx")
```

6 Step 4

Now run the below code and you will hopefully see that GPT can handle basic arithmetic!!

If not, see the cell below.

[10]:
```
from openai import OpenAI
from dotenv import load_dotenv
load_dotenv(override=True)

my_api_key = os.getenv("OPENAI_API_KEY")

print(f"Using API key --> {my_api_key} <--")

openai = OpenAI()
completion = openai.chat.completions.create(
    model='gpt-4o-mini',
    messages=[{"role":"user", "content": "What's 2+2?"}],
)
print(completion.choices[0].message.content)
print("Now go to Edit menu >> Clear Cell Output to remove the display of your␣
 ↪key.")
```

```
2 + 2 equals 4.
Now go to Edit menu >> Clear Cell Output to remove the display of your key.
```

6.1 If the key was set correctly, and this still didn't work

6.1.1 If there's an error from OpenAI about your key, or a Rate Limit Error, then there's something up with your API key!

First check this webpage to make sure you have a positive credit balance. OpenAI requires that you have a positive credit balance and it has minimums, typically around $5 in local currency. My salespitch for OpenAI is that this is well worth it for your education: for less than the price of a music album, you will build so much valuable commercial experience. But it's not required for this course at all; the README has instructions to call free open-source models via Ollama whenever we use OpenAI.

OpenAI billing page with credit balance is here:
https://platform.openai.com/settings/organization/billing/overview
OpenAI can take a few minutes to enable your key after you top up your balance.
A student outside the US mentioned that he needed to allow international payments on his credit card for this to work.

It's unlikely, but if there's something wrong with your key, you could also try creating a new key (button on the top right) here:
https://platform.openai.com/api-keys

6.1.2 If there's a cert related error

If you encountered a certificates error like:
`ConnectError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to get local issuer certificate (_ssl.c:1000)`
Then please replace: `openai = OpenAI()`
with:

```
import httpx
openai = OpenAI(http_client=httpx.Client(verify=False))
```

And if that works, you're in good shape. You'll just have to change the labs in the same way any time you hit this cert error.

6.2 If all else fails:

(1) Try pasting your error into ChatGPT or Claude! It's amazing how often they can figure things out

(2) Contact me! Please run the diagnostics in the cell below, then email me your problems to ed@edwarddonner.com

Thanks so much, and I'm sorry this is giving you bother!

7 Gathering Essential Diagnostic information

7.1 Please run this next cell to gather some important data

Please run the next cell; it should take a minute or so to run. Most of the time is checking your network bandwidth. Then email me the output of the last cell to ed@edwarddonner.com.
Alternatively: this will create a file called report.txt - just attach the file to your email.

[8]:
```
# Run my diagnostics report to collect key information for debugging
# Please email me the results. Either copy & paste the output, or attach the␣
 ↪file report.txt

!pip install -q requests speedtest-cli psutil setuptools
from diagnostics import Diagnostics
Diagnostics().run()
```

```
Starting diagnostics at 2025-01-06 22:58:22

===== System Information =====
Operating System: Darwin
MacOS Version: 10.16
Architecture: ('64bit', '')
Machine: x86_64
Processor: i386
Total RAM: 16.00 GB
Available RAM: 7.29 GB
Free Disk Space: 135.86 GB
```

```
===== File System Information =====
Current Directory: /Users/markku/Codes/GitHub/llm_engineering/week1
Write permission: OK

Files in Current Directory:
 - .DS_Store
 - .ipynb_checkpoints
 - Guide to Jupyter.ipynb
 - Intermediate Python.ipynb
 - Intermediate Python.pdf
 - __pycache__
 - community-contributions
 - day1.ipynb
 - day1.pdf
 - day2 EXERCISE.ipynb
 - day2 EXERCISE.pdf
 - day5.ipynb
 - diagnostics.py
 - report.txt
 - solutions
 - troubleshooting.ipynb
 - week1 EXERCISE.ipynb

===== Git Repository Information =====
Git Repository Root: /Users/markku/Codes/GitHub/llm_engineering
Current Commit: 01253a79e75b774ec851fd38496eab5c7605b9d7
Remote Origin: git@github.com:mplaine/llm_engineering.git

===== Environment File Check =====
.env file exists at: /Users/markku/Codes/GitHub/llm_engineering/.env
OPENAI_API_KEY found in .env file

===== Anaconda Environment Check =====
Anaconda environment is active:
Environment Path: /Users/markku/miniconda3/envs/udemy-llm-engineering
Environment Name: udemy-llm-engineering
Conda Version: conda 23.1.0

Python Environment:
Python Version: 3.11.11 (main, Dec 11 2024, 10:28:39) [Clang 14.0.6 ]
Python Executable: /Users/markku/miniconda3/envs/udemy-llm-
engineering/bin/python

Required Package Versions:
openai: 1.58.1
python-dotenv: 1.0.1
requests: 2.32.3
gradio: 5.9.1
```

```
transformers: 4.47.1

===== Virtualenv Check =====
No active virtualenv detected

===== Network Connectivity Check =====
SSL Version: OpenSSL 3.0.15 3 Sep 2024
  Connected to https://www.google.com
   Response time: 0.60s

Performing bandwidth test using speedtest-cli…
Download speed: 31.31 Mbps
Upload speed: 27.51 Mbps

===== Environment Variables Check =====

PYTHONPATH is not set.

Python sys.path:
 - /Users/markku/miniconda3/envs/udemy-llm-engineering/lib/python311.zip
 - /Users/markku/miniconda3/envs/udemy-llm-engineering/lib/python3.11
 - /Users/markku/miniconda3/envs/udemy-llm-engineering/lib/python3.11/lib-
dynload
 -
 - /Users/markku/miniconda3/envs/udemy-llm-engineering/lib/python3.11/site-
packages
 - /Users/markku/miniconda3/envs/udemy-llm-engineering/lib/python3.11/site-
packages/setuptools/_vendor
OPENAI_API_KEY is set after calling load_dotenv()

===== Additional Diagnostics =====
Temp directory is writable: /var/folders/l4/484f84ms4w76b45tpq_149zw0000gn/T

  All diagnostics passed successfully!


Completed diagnostics at 2025-01-06 22:58:50


Please send these diagnostics to me at ed@edwarddonner.com
Either copy & paste the above output into an email, or attach the file
report.txt that has been created in this directory.
```

[ ]: