day1

January 6, 2025

1 Instant Gratification

1.1 Your first Frontier LLM Project!

Let's build a useful LLM solution - in a matter of minutes.

By the end of this course, you will have built an autonomous Agentic AI solution with 7 agents that collaborate to solve a business problem. All in good time! We will start with something smaller…

Our goal is to code a new kind of Web Browser. Give it a URL, and it will respond with a summary. The Reader's Digest of the internet!!

Before starting, you should have completed the setup for PC or Mac and you hopefully launched this jupyter lab from within the project root directory, with your environment activated.

1.2 If you're new to Jupyter Lab

Welcome to the wonderful world of Data Science experimentation! Once you've used Jupyter Lab, you'll wonder how you ever lived without it. Simply click in each "cell" with code in it, such as the cell immediately below this text, and hit Shift+Return to execute that cell. As you wish, you can add a cell with the + button in the toolbar, and print values of variables, or try out variations.

I've written a notebook called Guide to Jupyter to help you get more familiar with Jupyter Labs, including adding Markdown comments, using `!` to run shell commands, and `tqdm` to show progress.

1.3 If you'd prefer to work in IDEs

If you're more comfortable in IDEs like VSCode or Pycharm, they both work great with these lab notebooks too.
If you'd prefer to work in VSCode, here are instructions from an AI friend on how to configure it for the course.

1.4 If you'd like to brush up your Python

I've added a notebook called Intermediate Python to get you up to speed. But you should give it a miss if you already have a good idea what this code does:
`yield from {book.get("author") for book in books if book.get("author")}`

1.5 I am here to help

If you have any problems at all, please do reach out.
I'm available through the platform, or at ed@edwarddonner.com, or at

https://www.linkedin.com/in/eddonner/ if you'd like to connect (and I love connecting!)

1.6 More troubleshooting

Please see the troubleshooting notebook in this folder to diagnose and fix common problems. At the very end of it is a diagnostics script with some useful debug info.

1.7 If this is old hat!

If you're already comfortable with today's material, please hang in there; you can move swiftly through the first few labs - we will get much more in depth as the weeks progress.

Please read - important note

The way I collaborate with you may be different to other courses you've taken. I prefer not to type code while you watch. Rather, I execute Jupyter Labs, like this, and give you an intuition for what's going on. My suggestion is that you do this with me, either at the same time, or (perhaps better) right afterwards. Add print statements to understand what's going on, and then come up with your own variations. If you have a Github account, use this to showcase your variations. Not only is this essential practice, but it demonstrates your skills to others, including perhaps future clients or employers…

Business value of these exercises

A final thought. While I've designed these notebooks to be educational, I've also tried to make them enjoyable. We'll do fun things like have LLMs tell jokes and argue with each other. But fundamentally, my goal is to teach skills you can apply in business. I'll explain business implications as we go, and it's worth keeping this in mind: as you build experience with models and techniques, think of ways you could put this into action at work today. Please do contact me if you'd like to discuss more or if you have ideas to bounce off me.

[2]:
```
# imports

import os
import requests
from dotenv import load_dotenv
from bs4 import BeautifulSoup
from IPython.display import Markdown, display
from openai import OpenAI

# If you get an error running this cell, then please head over to the␣
↪troubleshooting notebook!
```

2 Connecting to OpenAI

The next cell is where we load in the environment variables in your `.env` file and connect to OpenAI.

2.1 Troubleshooting if you have problems:

Head over to the troubleshooting notebook in this folder for step by step code to identify the root cause and fix it!

If you make a change, try restarting the "Kernel" (the python process sitting behind this notebook) by Kernel menu » Restart Kernel and Clear Outputs of All Cells. Then try this notebook again, starting at the top.

Or, contact me! Message me or email ed@edwarddonner.com and we will get this to work.

Any concerns about API costs? See my notes in the README - costs should be minimal, and you can control it at every point. You can also use Ollama as a free alternative, which we discuss during Day 2.

[3]:
```
# Load environment variables in a file called .env

load_dotenv(override=True)
api_key = os.getenv('OPENAI_API_KEY')

# Check the key

if not api_key:
    print("No API key was found - please head over to the troubleshooting␣
 ↪notebook in this folder to identify & fix!")
elif not api_key.startswith("sk-proj-"):
    print("An API key was found, but it doesn't start sk-proj-; please check␣
 ↪you're using the right key - see troubleshooting notebook")
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 - see troubleshooting␣
 ↪notebook")
else:
    print("API key found and looks good so far!")
```

```
API key found and looks good so far!
```

[4]:
```
openai = OpenAI()

# If this doesn't work, try Kernel menu >> Restart Kernel and Clear Outputs Of␣
 ↪All Cells, then run the cells from the top of this notebook down.
# If it STILL doesn't work (horrors!) then please see the Troubleshooting␣
 ↪notebook in this folder for full instructions
```

3 Let's make a quick call to a Frontier model to get started, as a preview!

[5]:
```
# To give you a preview -- calling OpenAI with these messages is this easy. Any␣
 ↪problems, head over to the Troubleshooting notebook.

message = "Hello, GPT! This is my first ever message to you! Hi!"
response = openai.chat.completions.create(model="gpt-4o-mini",␣
 ↪messages=[{"role":"user", "content":message}])
print(response.choices[0].message.content)
```

```
Hello! Welcome! I'm glad to hear from you. How can I assist you today?
```

3.1 OK onwards with our first project

[6]:
```
# A class to represent a Webpage
# If you're not familiar with Classes, check out the "Intermediate Python"␣
 ↪notebook

# Some websites need you to use proper headers when fetching them:
headers = {
 "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36␣
 ↪(KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36"
}

class Website:

    def __init__(self, url):
        """
        Create this Website object from the given url using the BeautifulSoup␣
 ↪library
        """
        self.url = url
        response = requests.get(url, headers=headers)
        soup = BeautifulSoup(response.content, 'html.parser')
        self.title = soup.title.string if soup.title else "No title found"
        for irrelevant in soup.body(["script", "style", "img", "input"]):
            irrelevant.decompose()
        self.text = soup.body.get_text(separator="\n", strip=True)
```

[7]:
```
# Let's try one out. Change the website and add print statements to follow␣
 ↪along.

ed = Website("https://edwarddonner.com")
print(ed.title)
print(ed.text)
```

Home - Edward Donner
Home
Outsmart
An arena that pits LLMs against each other in a battle of diplomacy and deviousness
About
Posts
Well, hi there.
I'm Ed. I like writing code and experimenting with LLMs, and hopefully you're here because you do too. I also enjoy DJing (but I'm badly out of practice), amateur electronic music production (
very
amateur) and losing myself in
Hacker News
, nodding my head sagely to things I only half understand.
I'm the co-founder and CTO of
Nebula.io
. We're applying AI to a field where it can make a massive, positive impact: helping people discover their potential and pursue their reason for being. Recruiters use our product today to source, understand, engage and manage talent. I'm previously the founder and CEO of AI startup untapt,
acquired in 2021
.
We work with groundbreaking, proprietary LLMs verticalized for talent, we've patented
our matching model, and our award-winning platform has happy customers and tons of press coverage.
Connect
with me for more!
December 21, 2024
Welcome, SuperDataScientists!
November 13, 2024
Mastering AI and LLM Engineering - Resources
October 16, 2024
From Software Engineer to AI Data Scientist - resources
August 6, 2024
Outsmart LLM Arena - a battle of diplomacy and deviousness
Navigation
Home
Outsmart
An arena that pits LLMs against each other in a battle of diplomacy and deviousness
About
Posts
Get in touch
ed [at] edwarddonner [dot] com
www.edwarddonner.com
Follow me

```
LinkedIn
Twitter
Facebook
Subscribe to newsletter
Type your email…
Subscribe
```

3.2 Types of prompts

You may know this already - but if not, you will get very familiar with it!

Models like GPT4o have been trained to receive instructions in a particular way.

They expect to receive:

A system prompt that tells them what task they are performing and what tone they should use

A user prompt – the conversation starter that they should reply to

[8]:
```
# Define our system prompt - you can experiment with this later, changing the␣
↪last sentence to 'Respond in markdown in Spanish."

system_prompt = "You are an assistant that analyzes the contents of a website \
and provides a short summary, ignoring text that might be navigation related. \
Respond in markdown."
```

[9]:
```
# A function that writes a User Prompt that asks for summaries of websites:

def user_prompt_for(website):
    user_prompt = f"You are looking at a website titled {website.title}"
    user_prompt += "\nThe contents of this website is as follows; \
please provide a short summary of this website in markdown. \
If it includes news or announcements, then summarize these too.\n\n"
    user_prompt += website.text
    return user_prompt
```

[10]:
```
print(user_prompt_for(ed))
```

```
You are looking at a website titled Home - Edward Donner
The contents of this website is as follows; please provide a short summary of
this website in markdown. If it includes news or announcements, then summarize
these too.

Home
Outsmart
An arena that pits LLMs against each other in a battle of diplomacy and
deviousness
About
Posts
Well, hi there.
```

I’m Ed. I like writing code and experimenting with LLMs, and hopefully you’re here because you do too. I also enjoy DJing (but I’m badly out of practice), amateur electronic music production (
very
amateur) and losing myself in
Hacker News
, nodding my head sagely to things I only half understand.
I’m the co-founder and CTO of
Nebula.io
. We’re applying AI to a field where it can make a massive, positive impact: helping people discover their potential and pursue their reason for being. Recruiters use our product today to source, understand, engage and manage talent. I’m previously the founder and CEO of AI startup untapt,
acquired in 2021
.
We work with groundbreaking, proprietary LLMs verticalized for talent, we’ve patented
our matching model, and our award-winning platform has happy customers and tons of press coverage.
Connect
with me for more!
December 21, 2024
Welcome, SuperDataScientists!
November 13, 2024
Mastering AI and LLM Engineering - Resources
October 16, 2024
From Software Engineer to AI Data Scientist - resources
August 6, 2024
Outsmart LLM Arena - a battle of diplomacy and deviousness
Navigation
Home
Outsmart
An arena that pits LLMs against each other in a battle of diplomacy and deviousness
About
Posts
Get in touch
ed [at] edwarddonner [dot] com
www.edwarddonner.com
Follow me
LinkedIn
Twitter
Facebook
Subscribe to newsletter
Type your email…
Subscribe

3.3 Messages

The API from OpenAI expects to receive messages in a particular structure. Many of the other APIs share this structure:

“‘ [{“role”: “system”, “content”: “system message goes here”}, {“role”: “user”, “content”: “user message goes here”}]

To give you a preview, the next 2 cells make a rather simple call - we won’t stretch the might GPT (yet!)

[11]:
```
messages = [
    {"role": "system", "content": "You are a snarky assistant"},
    {"role": "user", "content": "What is 2 + 2?"}
]
```

[12]:
```
# To give you a preview -- calling OpenAI with system and user messages:

response = openai.chat.completions.create(model="gpt-4o-mini",␣
 ↪messages=messages)
print(response.choices[0].message.content)
```

```
Oh, you're really hitting me with the tough ones today! The answer is 4.
Shocking, I know!
```

3.4 And now let’s build useful messages for GPT-4o-mini, using a function

[13]:
```
# See how this function creates exactly the format above

def messages_for(website):
    return [
        {"role": "system", "content": system_prompt},
        {"role": "user", "content": user_prompt_for(website)}
    ]
```

[14]:
```
# Try this out, and then try for a few more websites

messages_for(ed)
```

[14]:
```
[{'role': 'system',
  'content': 'You are an assistant that analyzes the contents of a website and
provides a short summary, ignoring text that might be navigation related.
Respond in markdown.'},
 {'role': 'user',
  'content': 'You are looking at a website titled Home - Edward Donner\nThe
contents of this website is as follows; please provide a short summary of this
website in markdown. If it includes news or announcements, then summarize these
too.\n\nHome\nOutsmart\nAn arena that pits LLMs against each other in a battle
of diplomacy and deviousness\nAbout\nPosts\nWell, hi there.\nI’m Ed. I like
```

writing code and experimenting with LLMs, and hopefully you're here because you do too. I also enjoy DJing (but I'm badly out of practice), amateur electronic music production (\nvery\namateur) and losing myself in\nHacker News\n, nodding my head sagely to things I only half understand.\nI'm the co-founder and CTO of\nNebula.io\n. We're applying AI to a field where it can make a massive, positive impact: helping people discover their potential and pursue their reason for being. Recruiters use our product today to source, understand, engage and manage talent. I'm previously the founder and CEO of AI startup untapt,\nacquired in 2021\n.\nWe work with groundbreaking, proprietary LLMs verticalized for talent, we've\npatented\nour matching model, and our award-winning platform has happy customers and tons of press coverage.\nConnect\nwith me for more!\nDecember 21, 2024\nWelcome, SuperDataScientists!\nNovember 13, 2024\nMastering AI and LLM Engineering - Resources\nOctober 16, 2024\nFrom Software Engineer to AI Data Scientist - resources\nAugust 6, 2024\nOutsmart LLM Arena - a battle of diplomacy and deviousness\nNavigation\nHome\nOutsmart\nAn arena that pits LLMs against each other in a battle of diplomacy and deviousness\nAbout\nPosts\nGet in touch\ned [at] edwarddonner [dot] com\nwww.edwarddonner.com\nFollow me\nLinkedIn\nTwitter\nFacebook\nSubscribe to newsletter\nType your email…\nSubscribe'}]

3.5 Time to bring it together - the API for OpenAI is very simple!

[15]:
```
# And now: call the OpenAI API. You will get very familiar with this!

def summarize(url):
    website = Website(url)
    response = openai.chat.completions.create(
        model = "gpt-4o-mini",
        messages = messages_for(website)
    )
    return response.choices[0].message.content
```

[16]:
```
summarize("https://edwarddonner.com")
```

[16]: "# Summary of Edward Donner's Website\n\nEdward Donner's website serves as a personal platform showcasing his interests and achievements in the field of technology, specifically focusing on large language models (LLMs) and artificial intelligence (AI). He is the co-founder and CTO of Nebula.io, which aims to harness AI for talent discovery and management. The site highlights his previous experience as the founder and CEO of the AI startup untapt, acquired in 2021.\n\n## Notable Projects\n- **Outsmart**: An innovative arena where LLMs compete in diplomacy and strategy.\n\n## Recent Posts and Announcements\n- **December 21, 2024**: Welcome message for SuperDataScientists.\n- **November 13, 2024**: Guide on mastering AI and LLM engineering, providing resources.\n- **October 16, 2024**: Resources for transitioning from a software engineer role to an AI data scientist.\n- **August 6, 2024**: Announcement of the Outsmart LLM Arena. \n\nThe website reflects Ed's passion for coding, music, and engaging

```
with technology communities."
```

[17]:
```
# A function to display this nicely in the Jupyter output, using markdown

def display_summary(url):
    summary = summarize(url)
    display(Markdown(summary))
```

[18]:
```
display_summary("https://edwarddonner.com")
```

4 Summary of Edward Donner's Website

Edward Donner's website provides insight into his interests and professional background. He is a coder and LLM (Large Language Model) experimenter, as well as a co-founder and CTO of Nebula.io, which utilizes AI to improve talent discovery and management. Previously, he founded untapt, an AI startup acquired in 2021. In addition to his professional endeavors, he mentions a passion for DJing and electronic music production.

4.1 Features of the Website:

- **Outsmart**: A project where LLMs compete in diplomacy and strategy.
- **Blog Posts**: Various posts offering resources on AI and data science, transitioning from software engineering, and promoting the Outsmart initiative.

4.1.1 Recent Announcements:

- **August 6, 2024**: Introduction to "Outsmart LLM Arena."
- **October 16, 2024**: Resources for transitioning from Software Engineer to AI Data Scientist.
- **November 13, 2024**: Resources for mastering AI and LLM Engineering.
- **December 21, 2024**: Welcome message for "SuperDataScientists!"

5 Let's try more websites

Note that this will only work on websites that can be scraped using this simplistic approach.

Websites that are rendered with Javascript, like React apps, won't show up. See the community-contributions folder for a Selenium implementation that gets around this. You'll need to read up on installing Selenium (ask ChatGPT!)

Also Websites protected with CloudFront (and similar) may give 403 errors - many thanks Andy J for pointing this out.

But many websites will work just fine!

[19]:
```
display_summary("https://cnn.com")
```

6 Summary of CNN Website

CNN is a leading global news outlet that provides breaking news, in-depth reports, and various types of multimedia content across multiple topics including politics, business, health, and entertainment. The site features news sections for both US and international stories, along with dedicated categories for topics such as science, climate, and sports. CNN also offers video content, live TV, and audio podcasts for diverse audience engagement.

6.1 Latest News Highlights

- **Jimmy Carter**: Emphasis on the humanitarian impact and legacy of the former President, with details about his funeral plans.
- **Trump Cases**: Federal appeals court supports a sexual abuse case against Trump, with discussions on potential implications for his future.
- **South Korean Airlines**: Investigations are ongoing related to a recent fatal accident.
- **Walmart's Birkin Lookalike**: Highlighting the internet buzz around Walmart's imitation of luxury fashion items.
- **Taliban Policies**: An announcement indicating the closure of NGOs employing Afghan women, raising significant humanitarian concerns.
- **China's Military Movements**: Notable increase in Russian flights hinting towards strategic actions in the Mediterranean.

The website generally underscores essential current events and announcements, reflecting the most pressing global news articles and insights.

[20]:
```
display_summary("https://anthropic.com")
```

7 Summary of Anthropic Website

Anthropic is an AI safety and research company based in San Francisco, specializing in creating reliable and beneficial AI systems. The website showcases their AI model, Claude, with the latest version being **Claude 3.5 Sonnet**, recently launched.

7.1 Key Highlights

- **Claude 3.5 Sonnet**: This is the most intelligent model offered by Anthropic and is available for interaction and integration into applications via their API.
- **New Announcements**:
  - On **October 22, 2024**, Anthropic introduced updates related to computer use, including the release of **Claude 3.5 Sonnet** and **Claude 3.5 Haiku**.
  - They also emphasize their commitment to AI safety through ongoing research and safety-oriented practices.

7.2 Research Focus

The company is involved in various research initiatives, including: - **Constitutional AI**: Working towards harmlessness in AI through feedback mechanisms. - Insights into AI safety, including their foundational beliefs and practices regarding AI alignment.

Anthropic aims to foster innovation while prioritizing the safety and ethical implications of AI technology.

[21]:
```
display_summary("https://markkulaine.com")
```

8 Summary of markkulaine.com

Markku Laine's personal website is primarily a link to his Linktree profile, which can be found at linktr.ee/dr.climber. This suggests that the site serves as a gateway to various platforms or content associated with Markku Laine, rather than offering extensive information or resources directly on the website itself.

[22]:
```
display_summary("https://linktr.ee/dr.climber")
```

9 Summary of @dr.climber | Linktree

The website is a professional profile for @dr.climber, who is a Data Scientist with a PhD and a hybrid athlete. The individual has a strong passion for data, artificial intelligence, and analytics.

9.1 Professional Highlights

- **Education**: Google Advanced Data Analytics Professional Certificate (2023).
- **Research & Publications**: Includes a doctoral dissertation (2018) and various project analyses, such as:
  - Analyzing Job Postings with SQL (2024)
  - Predicting Employee Attrition (2023)
  - AI-assisted Design Tools (2023)
  - Visual Attention in Browsing (2023)

9.2 Sports Achievements

- Finnish Calisthenics Champion (2023)
- Vice World Pull-Up Champion (2022)
- 5th place at Beast of the Barz (2022)
- 8A in Bouldering (2016)

The website serves as a hub for connecting through email, GitHub, and LinkedIn, showcasing a unique blend of data science expertise and athletic accomplishments.

Business applications

In this exercise, you experienced calling the Cloud API of a Frontier Model (a leading model at the frontier of AI) for the first time. We will be using APIs like OpenAI at many stages in the course, in addition to building our own LLMs.

More specifically, we've applied this to Summarization - a classic Gen AI use case to make a summary. This can be applied to any business vertical - summarizing the news, summarizing financial performance, summarizing a resume in a cover letter - the applications are limitless. Consider how you could apply Summarization in your business, and try prototyping a solution.

Before you continue - now try yourself

Use the cell below to make your own simple commercial example. Stick with the summarization use case for now. Here's an idea: write something that will take the contents of an email, and will suggest an appropriate short subject line for the email. That's the kind of feature that might be built into a commercial email tool.

[23]:
```
# Step 1: Create your prompts

system_prompt = "You are an assistant whose task is to come up with a witty␣
 ↪subject for the given email."
user_prompt = """
    Hyvä BK Climbersin jäsen!

Kiitos, että olet ollut jäsenemme vuonna 2024! Jäsenyyden voit taas␣
 ↪omatoimisesti uusia vuodelle 2025 nettisivuilla olevan lomakkeen kautta.␣
 ↪Näin se jatkuu katkeamattomana tulevaan vuoteen.

Jäsenmaksun pääsee maksamaan heti nettisivuilla, joten jäsenyyden ja␣
 ↪vakuutuksen saa saman tien voimaan vuoden vaihtuessa. Maksettuasi␣
 ↪jäsenmaksun, jäsenyys ja vakuutus ovat voimassa kalenterivuoden loppuun␣
 ↪saakka. Sporttiturva-vakuutus on voimassa vielä seuraavan vuoden tammikuun␣
 ↪loppuun asti.

Todistuksena jäsenyydestä toimii sähköpostiin tuleva maksuvahvistus eli pidä se␣
 ↪hyvässä tallessa. Tapaturman sattuessa maksukuittia näyttämällä saat␣
 ↪palvelua Pihlajalinnasta (ent. Pohjola Sairaala). Lue lisäohjeita täältä!
"""

# Step 2: Make the messages list

messages = [
    {"role": "system", "content": system_prompt},
    {"role": "user", "content": user_prompt}
] # fill this in

# Step 3: Call OpenAI

response = openai.chat.completions.create(model="gpt-4o-mini",␣
 ↪messages=messages)

# Step 4: print the result

print(response.choices[0].message.content)
```

```
"Uusi vuosi, uusi seikkailu: Uusi jäsenyys BK Climbersille!"
```

9.3 An extra exercise for those who enjoy web scraping

You may notice that if you try `display_summary("https://openai.com")` - it doesn't work! That's because OpenAI has a fancy website that uses Javascript. There are many ways around this that some of you might be familiar with. For example, Selenium is a hugely popular framework that runs a browser behind the scenes, renders the page, and allows you to query it. If you have experience with Selenium, Playwright or similar, then feel free to improve the Website class to use them. In the community-contributions folder, you'll find an example Selenium solution from a student (thank you!)

10 Sharing your code

I'd love it if you share your code afterwards so I can share it with others! You'll notice that some students have already made changes (including a Selenium implementation) which you will find in the community-contributions folder. If you'd like add your changes to that folder, submit a Pull Request with your new versions in that folder and I'll merge your changes.

If you're not an expert with git (and I am not!) then GPT has given some nice instructions on how to submit a Pull Request. It's a bit of an involved process, but once you've done it once it's pretty clear. As a pro-tip: it's best if you clear the outputs of your Jupyter notebooks (Edit » Clean outputs of all cells, and then Save) for clean notebooks.

PR instructions courtesy of an AI friend: https://chatgpt.com/share/670145d5-e8a8-8012-8f93-39ee4e248b4c

[ ]: