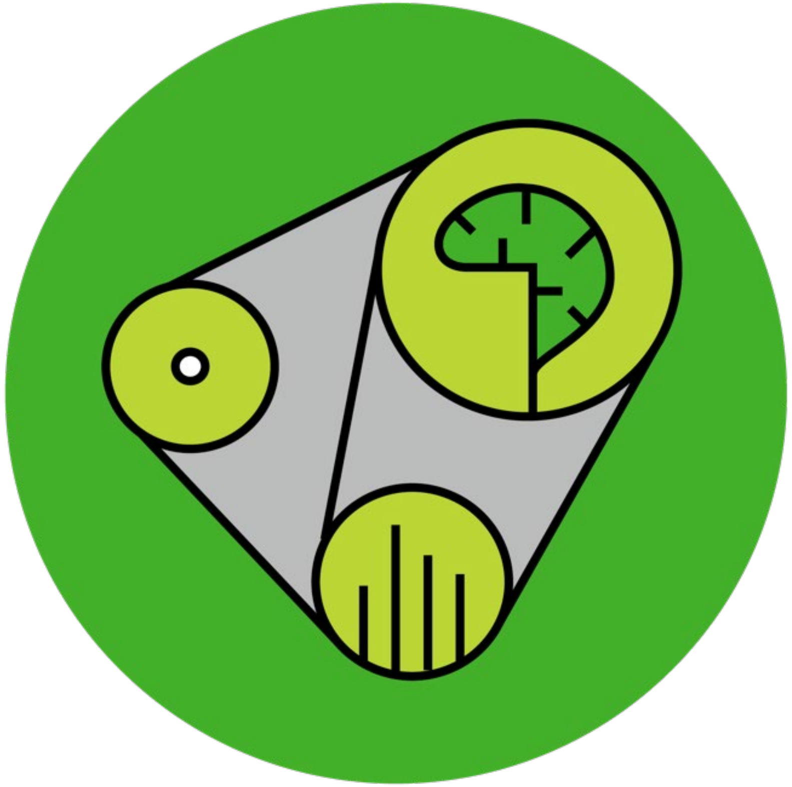
Deloitte.



AI Guild | GenAI Practicum

Session 7 - Use Cases

MAKING AN IMPACT THAT MATTERS since 1845

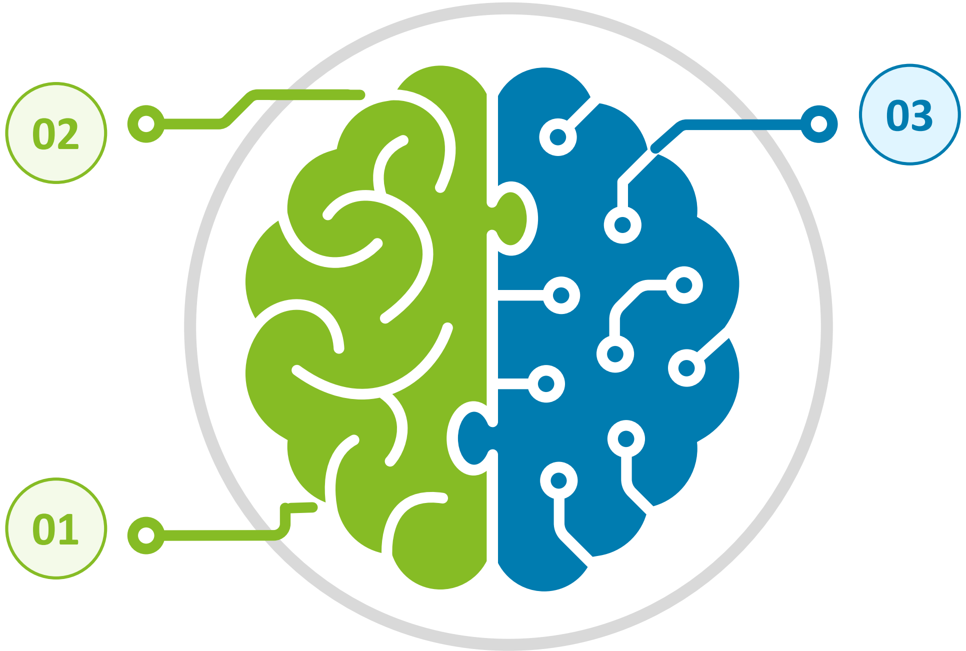
Week Agenda

Topic

- 01 Sidekick
- 02 Coding a Use Case
- 03 Building a Use Case

Learning objectives

By the end of session, you should be able to



Use Sidekick to work on a use case

Know how to build a working POC for a use case

Understand Licensing / Pricing, Risk, Ethics, Guardrails, Benefits and Costs of different models

Prerequisites

- Completion of AI Academy individual tracks or Pathway 0
- This practicum requires a prior knowledge of Python coding and the basics of NLP

Final Considerations

Understand Licensing / Pricing, Risks, and Guardrails

Licensing

Different Cloud Vendors/Projects offer different Licensing models

- OpenAI, Microsoft, Google, and Amazon all offer commercial licensing, with contract terms that vary
- They also offer provisioned capacity, on-demand, and batch processing options
- Open-source models are also licensed, but typically under an o source standard license, such as the Apache 2.0 license
- Meta’s LLaMA 2 is “open”, but licensed from Meta under the LLaMA license, so is it really open-source? “Open Source Isn’ Ready for Generative AI” or is it, see “Octoverse: The state of source and rise of AI in 2023”
- A significant advantage of the open-source models is that most of them expose their codebase to developers so that engineers can understand how the model is trained, and what the weights used in that training were, making open-source models less of a black box

Open-source Licensing

MIT/BSD License

- Encourages wide use and modification with minimal restrictions. Ideal for projects seeking maximum freedom

GNU General Public License (GPL)

- Ensures all derivative works remain open source. Best for projects dedicated to open-source continuity

Apache License 2.0

- Balances openness with protection, offering patent rights. Suitable for projects that require patent protection

Pricing

Different Cloud Vendors offer different Pricing models

- OpenAI charges are per token per model, with different rates for prompts and responses.
  - gpt-4-1106-preview is $0.01 / 1K tokens input, and $0.03 output
- Azure only offers the OpenAI models and prices are a function of OpenAI's prices
- Google only offers their own models and prices are per character per model.
  - PaLM 2 for Text is $0.0005/1,000 characters
- AWS has the most diverse offering, and different models are priced by vendor, region, tokens, and storage.
  - Amazon Titan Text Lite is $0.0003/1K tokens input, and $0.0004 output,
  - Anthropic Claude is $0.008/1K input, and $0.024 output in US East and US West regions.
- Open-source models are "free". But costs vary depending on the hardware and cloud services required to host and run the various open-source models.

Risks to Monitor

- **Lack of transparency:** Generative AI and ChatGPT models are unpredictable, and not even the companies behind them always understand everything about how they work.

- **Accuracy:** Generative AI systems sometimes produce inaccurate and fabricated answers. Assess all outputs for accuracy, appropriateness and actual usefulness before relying on or publicly distributing information.

- **Bias:** You need policies or controls in place to detect biased outputs and deal with them in a manner consistent with company policy and any relevant legal requirements.

- **Intellectual property (IP) and copyright:** There are currently no verifiable data governance and protection assurances regarding confidential enterprise information. Users should assume that any data or queries they enter into the ChatGPT and its competitors will become public information, and we advise enterprises to put in place controls to avoid inadvertently exposing IP.

- **Cybersecurity and fraud:** Enterprises must prepare for malicious actors' use of generative AI systems for cyber and fraud attacks, such as those that use deep fakes for social engineering of personnel, and ensure mitigating controls are put in place. Confer with your cyber-insurance provider to verify the degree to which your existing policy covers AI-related breaches.

- **Sustainability:** Generative AI uses significant amounts of electricity. Choose vendors that reduce power consumption and leverage high-quality renewable energy to mitigate the impact on your sustainability goals.

Guardrails

- How can you manage the prompt if a user can ask anything?
- How can you govern the response?
- Building guardrails into your prompting process will help.
- Developing an appropriate escalation process will be necessary.
- Some guardrails will be built in to the LLM. Which ones? Who manages/controls that?
- How can you prevent copy-and-paste of confidential information?
- Who can see/read/leak your prompts?
- What types of prompts should your Gen AI tool prevent/allow/escalate?

Use Case Final Project

Use Case Final Project

The Final Project for the US Track 1 Cohort 2 is due May 5th

Submit the link to GitHub in the survey that will be sent for this purpose.

- The case study that you're building in session 6 and 7 forms your assessment submission for the practicum
- You can use one of your existing forked repos to write the code. Create a new folder called "assessment" in which all the associated code and data would reside
- Since you are a team contributing to the same project, please ensure that the GitHub repo can see contributions to the code from **all** members of the team.
- Team members will only get credit if you have a commit in the contents of the assessment folder
- At the end, every participant will be asked to submit the GitHub repo link. Here all members of the same team can submit the same GitHub repo link.
- Ensure that the GitHub repo has been created inside the DEP-Training GitHub org
- Inside the "assessment" folder of your repo, have a file called "README.md" that contains details of your project and the list of contributors from your team along with email IDs
- Please use only the provided tools for this assessment: GitHub Copilot, Azure OpenAI API, LangChain, sidekick and a Vector database of your choice. The solution should run inside GitHub Codespaces, within the DEP-Training GitHub org. In the Readme, please provide a step-by-step instructions to run the code.
- Additional collaboration tools, such as VS Live Share can be used as well.

Use Case Final Project

The Final Project for the US Track 1 Cohort 2 is due May 5th

- The final project is your opportunity to build something with what you've learned in this program. We expect that you are now well suited to prepare and discuss a use case of some relevance to you and your teammates. And to begin to build it.
- Despite having a large cohort, we expect you to self-organize to some degree. In session 6, you will have the opportunity to meet with your peers, and identify a use case. The group numbers are just suggestions as they are randomly assigned. Please communicate with each other in Teams chat to find up to 3-5 people who are interested in working on a single use case.
- All team members are expected to contribute to the final project. A small agile team of 3-5 people should be able to accomplish quite a lot in the time allotted.
- You have two weeks to complete this assignment. It is expected that each team member will be spend 3-8 hours over that period to complete their contribution. This should be a collaborative effort and we encourage you to have zoom calls after session 7 to work on several aspects of the project
- The documentation should appear in a project readme.md or PowerPoint deck. Although the documentation should be clean and professional, it is not intended for you to spend hours and hours on a deck. The documentation, the code, and any required supporting files should be submitted in the GitHub repo.

Use Case Final Project

The Final Project for the US Track 1 Cohort 2 is due May 5th

Item	Possible Points
Use Case Identified and Elaborated in readme.md or pptx file	10
Ethical Considerations Discussed and Mitigated	10
Requirements defined, for example, as User Stories	15
Tasks outlined and assigned to team members	5
Jupyter Notebook runs in GitHub Codespace	20
Code formatted and commented to be easily understood	10
Code includes elements from the course	25
An explanation of how GenAI was utilized in the creation of the project	5

- See the Speaker Notes below for more explanation.

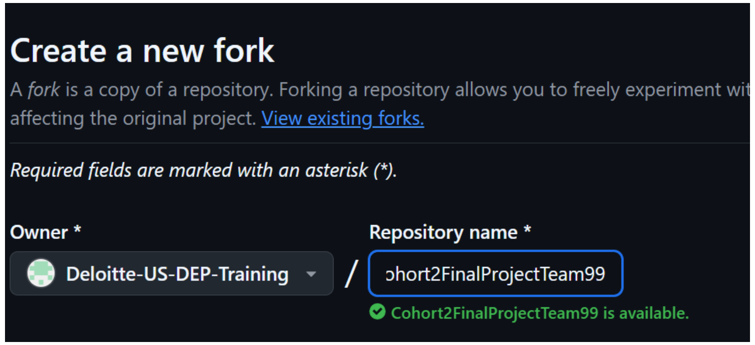
GitHub Mechanics

Pick one member of your team to be the host of the shared repo.

The host should be someone who's Codespace is working.

In GitHub, in the host's fork, create a new fork named Cohort2FinalProjectTeamNN where NN is the team number that you have assembled in Session 4, 5, 6, or even today.

For example,



Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment wit
affecting the original project. View existing forks.

Required fields are marked with an asterisk (*).

Owner * Deloitte-US-DEP-Training / Repository name * ohort2FinalProjectTeam99

Cohort2FinalProjectTeam99 is available.

Next, you need to share the new repo with your team.

Do this in the Forked Repo's **Settings** tab.

Then, on the left, select **Collaborators and teams**.

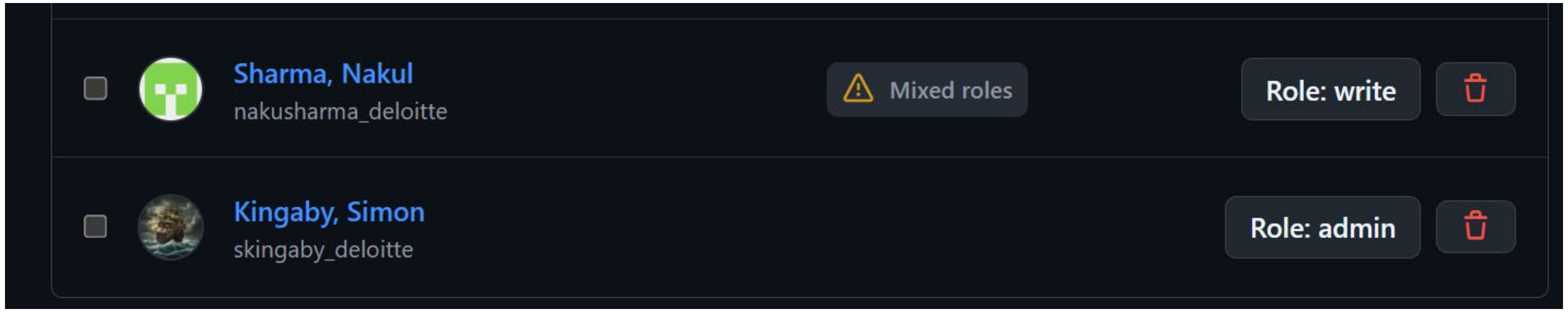
Then, on the right, halfway down the screen, next to the heading Manage access, click on **Add people**.

After entering the team member's name, you will need to select a role, choose one of these:

- Write
- Maintain
- Admin

Whatever works for your team. As long as everyone has at least Write access, the project should go smoothly.

For example,



Sharma, Nakul
nakusharma_deloitte
Mixed roles
Role: write

Kingaby, Simon
skingaby_deloitte
Role: admin

Now, everyone in the team needs to create their own branch in the host's fork. Then create a codespace in that branch.

Then, when you are in the codespace, you will need to work in your branch and merge your branch into the main branch.

At this point it is worth mentioning that Git is a tool the devil created, and GitHub is his playground.

There are numerous social media videos and other trainings available to sort out the GitHub repos, forks, branches, and pull requests. Most of which is outside the scope of this session.

Lab 1

Review the Sample Solution

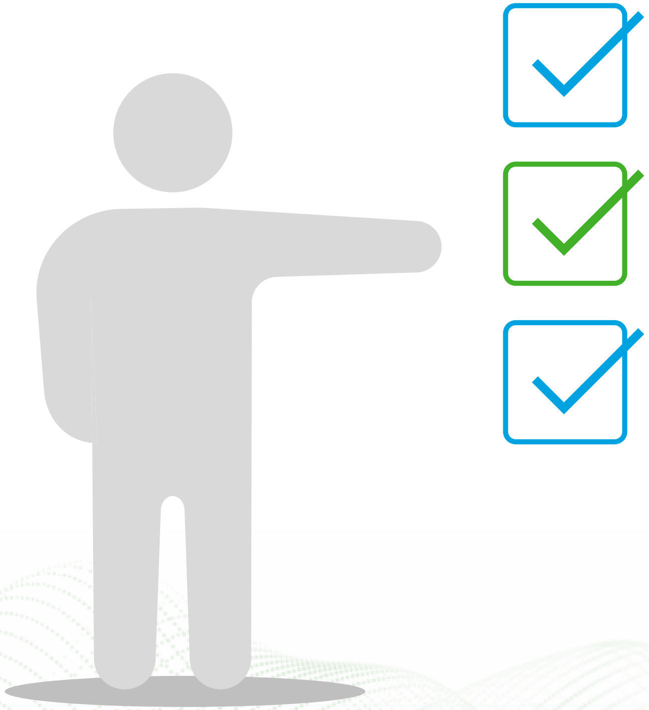
- 15 minutes

Review the following:

- Sample Solution – A basic chatbot
- Sidekick https://sidekick.deloitte.com or https://sidekick.gps.deloitte.com
- Prompt https://kx.deloitte
- Deloitte Assistant https://assistant.deloitte.com

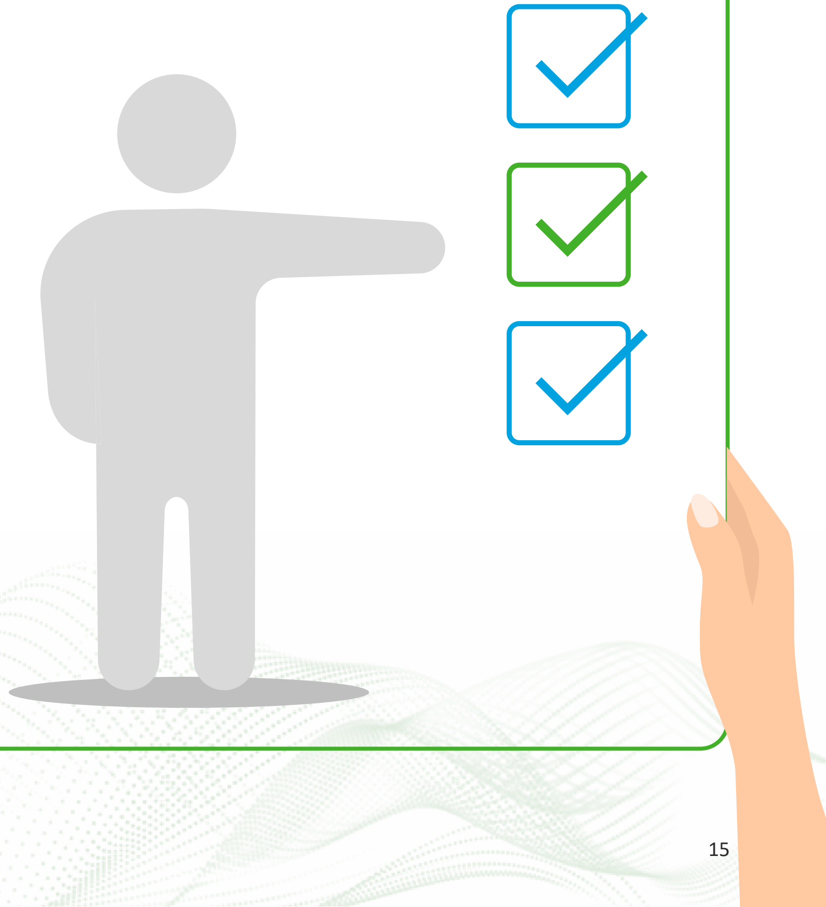
Sidekick, Prompt, and Deloitte Assistant are three of the firm's GenAI tools.

Observe how they are designed and the features or functions you might like to include in your solution.



Lab 2
Identify the Requirements

- 20 Minutes
- In the same groups of 3-5 people
- Select a breakout room
- Document the 5-10 Features and their User Stories for your Use Case



Lab 3

Building a GenAI Use Case

- 10 minutes

Last week you elaborated a specific GenAI Use Case. This week you will build a Proof-of-Concept in Python for that Use Case.

Divide the Tasks

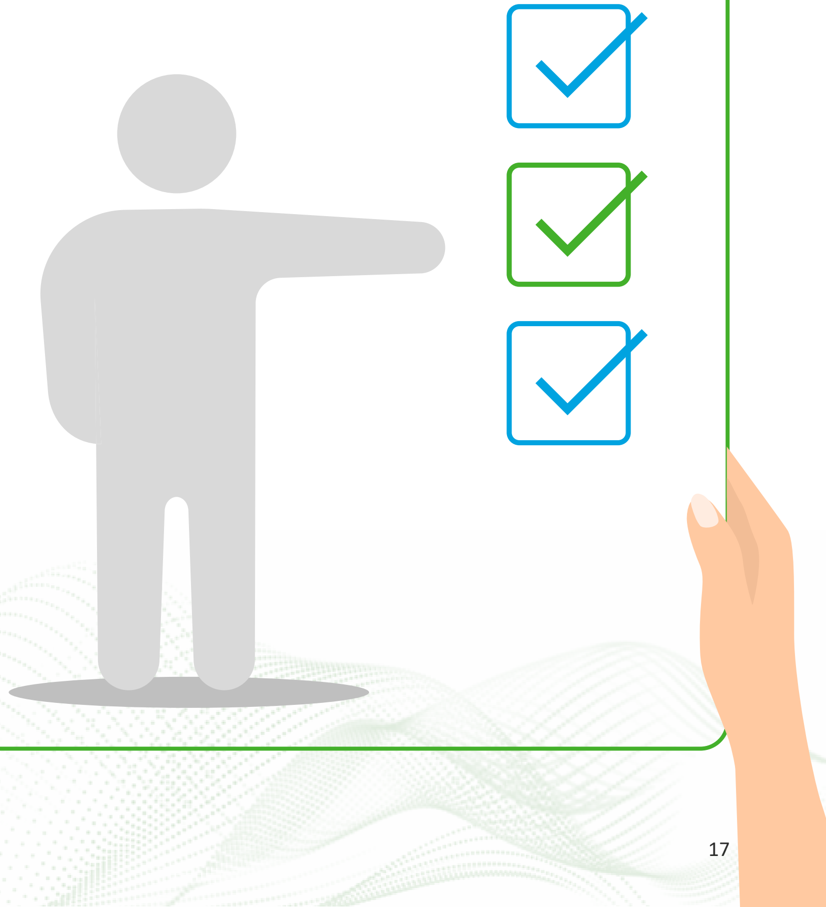
- Spend this time figuring out who will do what to bring the POC together

Things to Consider:

- Use the OpenAI API syntax when calling GenAI functions
- Use LangChain as necessary
- Select a Vector database from the ones covered in the RAG sections

Lab 4
Start to Build the Proof-of-Concept

- 30 Minutes
- In the same groups of 3-5 people
- Select a breakout room
- Using what you have learned, create a Python notebooks with some working code to develop a POC instance of your solution
- With 10 minutes left, make sure you can get your code into GitHub.
- Your homework is to complete the POC with your team.
- US must remit a repository link by Dec 9th
- USI must remit a repository link by Dec 13th



Q & A

Session Takeaways

Share key Takeaways

01 LLM's can do some incredible things with the right prompting

02 LangChain can be integrated to power action-based applications

03 RAG is a way to supplement the LLM with your own data

04 Multimodal models are enabling new and interesting ways to be creative and to see and hear the world around us

05 Gen AI has a role to play in every industry and every field

Appendix

Open-source Licensing

Open-source licenses for generative AI, like other software, come in various forms, each with its own set of rules and implications. Here's a comparison and contrast of some of the most common open-source licenses used in generative AI projects:

1. MIT License:

Permissiveness: Highly permissive.

Main Features: Allows users to do almost anything they want with the software, including using, copying, modifying, merging, publishing, distributing, sublicensing, and/or selling copies.

Requirements: License and copyright notice must be included in all copies or substantial portions of the software.

Suitability for Generative AI: Suitable for developers who want to allow maximum freedom for users, with minimal restrictions.

2. GNU General Public License (GPL):

Permissiveness: Less permissive compared to MIT.

Main Features: Requires that modified versions of the licensed software must also be open source. Known for its copyleft requirement.

Requirements: If you distribute or modify the software, it must remain under the GPL license.

Suitability for Generative AI: Suitable for projects where maintaining open-source status for all derivative works is important.

Open-source Licensing

3. Apache License 2.0:

Permissiveness: More permissive than GPL but less than MIT.

Main Features: Allows for commercial use, modification, distribution, patent use, and private use. It also provides an express grant of patent rights from contributors to users.

Requirements: Changes made to the licensed software must be documented. Includes a provision that prevents the trademark from being used without permission.

Suitability for Generative AI: Good for projects that need patent protection and are comfortable with allowing modifications without the requirement for those modifications to be open source.

4. BSD License:

Permissiveness: Similar to the MIT License in its permissiveness.

Main Features: Similar to MIT, but with clauses regarding the use of the name of the project or its contributors. Generally, it prohibits the use of the name of the project or its contributors for endorsement purposes without prior consent.

Requirements: Redistribution and use in source and binary forms must retain the copyright notice and disclaimer.

Suitability for Generative AI: Suitable for projects that are fine with broad usage but want to control the endorsement or promotion using their project’s name.

Open-source Licensing

5. Creative Commons (CC):

Permissiveness: Varies depending on the specific Creative Commons license used.

Main Features: Originally designed for creative works rather than software, but sometimes used in generative AI. Allows for a range of permissions and restrictions, including commercial use, modifications, and sharing under the same license.

Requirements: Depends on the specific CC license, but often includes attribution and may include share-alike or non-commercial clauses.

Suitability for Generative AI: Can be suitable for generative AI content (like models and datasets), especially when a balance of sharing and restriction is desired.

Deloitte.