

Module 1

Prompt Engineering

Hardeep Johar

Senior Lecturer in the Discipline of
Industrial Engineering and Operations Research

The Basic of LLMs

Large Language Models - 1

- A Large Language Model (LLM) is an AI system
- Usually built using large (terrabites) of text data
 - Wikipedia, news stories, books, chat groups,
- LLMs are built using transformer models (proprietary configurations)
- LLMs build response sequences (text) using next word prediction
- LLMs are “foundational models”
 - They are general purpose
 - Can answer questions about almost anything
 - Can be tailored to specific purposes (fine tuning)
- LLMs use a prompt-response (chat) format to interact with users

Large Language Models - 2

- LLMs are multi-modal
 - Text
 - Sound (text to sound generation)
 - Images (text to image generation)
 - Video (text to video generation)
- Open AI Sora
 - <https://openai.com/index/sora/>

ChatBots

- Chatbots are applications that interface with humans in natural language and communicate in a human-human style
- Chatbots have been around for while
- And have been frustrating humans for a while!
 - Tend to be scripted
 - The AI part is usually in understanding the user input
 - Output is usually canned
- Alexa, Siri are more competent examples

Generative AI ChatBots

- Generate text based on tokens and their context
- Keep track of in-conversation context
- Let's see what's under the hood!
- Visit <https://platform.openai.com/tokenizer> and follow the steps in the next few slides

<https://platform.openai.com/tokenizer>

Type “Jack loves Jill” in the box

GPT-3.5 & GPT-4

GPT-3 (Legacy)

Jack loves Jill

Clear

Show example

Tokens

Characters

3

15

Jack loves Jill

Text

Token IDs

Click on the TokenIds tab in the lower text box

GPT-3.5 & GPT-4

GPT-3 (Legacy)

Jack loves Jill

Clear

Show example

Tokens

3

Characters

15

[33731, 16180, 48311]

Text

Token IDs

Text to Tokens

- This should look familiar to us by now!
- The text is converted into a series of tokens
- As we've seen before, these are just numbers
- Here, 3 words have been translated into 3 tokens. Roughly:
 - Jack 33731
 - loves 16180
 - Jill 48311

Now flip Jack and Jill in the upper text box

GPT-3.5 & GPT-4 GPT-3 (Legacy)

Jill loves Jack

Clear

Show example

Tokens

4

Characters

15

[41, 484, 16180, 7762]

Text

Token IDs

Context Matters! - 1

- The token IDs for Jack and Jill have changed
- We now have 4 tokens:
 - Jill: 2 tokens 41 and 484
 - Jack: 7762 (changed from 33731
 - loves: 16180 (unchanged)
- Roughly:
 - loves means the same as before (somebody loves somebody)
 - Jill has moved from object to subject (Jack loves her to she loves Jack)
 - Jack has moved from subject to object
- Tokens contain context information!

Context Matters! - 2

- Let's take this a little further
- Type "My aunt Jill loves Jack"
- What should you see?
 - Note that the subject is now "My aunt"
 - Jill and Jack are both objects and use tokens in the object context
 - Loves is still loves, because amore è sempre amore!

My aunt Jill loves Jack

GPT-3.5 & GPT-4 GPT-3 (Legacy)

My aunt | Jill loves Jack

Clear

Show example

Tokens

5

Characters

23

[5159, 51166, 48311, 16180, 7762]

Text

Token IDs

Compare the Three!

Text	Jack	loves	Jill
Token	33731	16180	48311

Text	Jill		loves	Jack
Token	41	484	16180	7762

Text	My	aunt	Jill	loves	Jack
Token	5159	51166	48311	16180	7762

What Changes?

- Jill and Jack have both taken their object token values
- Loves is still loves
- Try inserting Jack and Jill in longer sentences. Does the token value change?

Can you figure out what the change means?

Tokens

- As we saw in the previous module, tokens are the smallest unit of text analysis
- AI chatbots generally charge users by the number of tokens
- For example, “Jill loves Jack” translates into 4 tokens and will cost you just a little bit more than “Jack loves Jill”
- The number of tokens is also a limiting usage factor:
 - GPT3 limits the user to 4096 tokens
 - GPT4 limits the user to 8192 tokens
 - GPT4.5 limits the user to 32768 tokens
- **Your prompt and the response must be within the limiting usage amount**
 - GPT3.5: 4096 tokens
 - GPT4: 8192 tokens

Prompt Engineering

Prompts vs Questions

- A prompt is a statement that asks for a response
 - Prompts are conversational
 - Prompts typically contain information that helps the responder structure their response
 - E.g., “Tell me about the time you hiked to the Everest Base Camp focusing more on the changing scenery as you climbed up”
 - **A prompt is the primary means of interacting with a chat bot**
- A question is a statement that directly asks for information
 - How high is the Everest Base Camp?
 - A question could be a prompt if the context is already known to the responder
 - A prompt can be phrased as a question

Prompt Engineering - Definition and Goal

- “Prompt engineering is the process of structuring text that can be interpreted and understood by a generative AI model”
* source: https://en.wikipedia.org/wiki/Prompt_engineering
- The goal of prompt engineering is to use natural language to “converse” with a large language model to get some useful output
 - Answers to questions
 - Natural language text
 - Programming code
 - Images
 - Voice
 - Videos

Prompt Engineering - Rules

Simple rules!

- Ask the right question!
- Ask the right question the right way!
- Carefully set up the context!
- Be clear about the structure of the response!

Prompt Engineering

Prompts can be:

- Zero shot:
 - Direct questions or requests
 - Usually used when seeking a fact or statistic
- One shot:
 - Question or request accompanied by one example
 - The example is often used to illustrate what sort of response is expected or to set the context
- Few shots:
 - Provide many examples to the LLM and then ask a question or make a request
 - Many examples are often necessary when asking the bot to make new associations

Prompt Engineering: Zero Shot Example 1



You


Write the first sentence of a romance novel.



ChatGPT

Amidst the bustling streets of Paris, where the scent of freshly baked croissants mingled with the melodies of street musicians, their eyes met across the crowded cafe, and in that fleeting moment, time seemed to stand still.

Prompt Engineering: Zero Shot Example 2

 ChatGPT

56



You

What is $7*8$?

Prompt Engineering: Zero Shot Example 3



You

Who has the high jump record in the Olympics?



ChatGPT

As of my last update, the men's high jump world record of 2.45 meters (8 feet 0.46 inches) is held by Javier Sotomayor of Cuba, set on July 27, 1993. For women, the world record of 2.09 meters (6 feet 10.28 inches) is held by Stefka Kostadinova of Bulgaria, set on August 30, 1987. These records may have changed since then, so it's a good idea to check for the most recent updates.

Prompt Engineering: Zero Shot - 4

- Context flows from one prompt to another



You

long jump?



ChatGPT

As of my last update, the Olympic record for the men's long jump is 8.90 meters (29 feet 2½ inches), set by Bob Beamon of the United States at the 1968 Mexico City Olympics. For women, the Olympic record is 7.40 meters (24 feet 3½ inches), achieved by Jackie Joyner-Kersey of the United States at the 1988 Seoul Olympics. These records may have changed since then, so it's advisable to verify with the latest Olympic results.

Prompt Engineering: Zero Shot - 5

- You can instruct the bot on how to format the output



You

Just give me the output in the
format name:distance:year



ChatGPT

Certainly! Here's the output in the requested
format:

Bob Beamon:8.90 meters:1968

Jackie Joyner-Kersey:7.40 meters:1988

Prompt Engineering: Context Setting

- Setting the context is very important since a chatbot can't read your mind
- Be as precise as possible

Prompt Engineering: Context Setting Example

- Example:



You

Hi. I'm going to ask you questions about the Olympics. Could you make sure your answers are specifically related to the olympics and nothing else?

- Then ask the question



You

Tell me something about nazism in 3 sentences.



ChatGPT

During the 1936 Berlin Olympics, hosted by Nazi Germany, Hitler intended to showcase Aryan racial superiority and the supposed strength of the Nazi regime. The Olympics became a propaganda tool for the Nazis, who sought to use the event to promote their ideology of racial supremacy. Despite concerns and controversies surrounding Nazi policies, the Games proceeded, with Jesse Owens, an African American athlete, notably winning four gold medals, challenging Hitler's racial theories and providing a powerful counter-narrative to Nazi propaganda.

Prompt Engineering: One Shot Examples



You

Write the first sentence of a romance novel using the first sentence of *Pride and Prejudice* as an example.



ChatGPT

In the bustling city of London, where whispers of intrigue floated through the air like petals on a summer breeze, it was an accepted truth that a debutante of impeccable pedigree must be in search of a suitable match for the upcoming season.



You

If $6\#2$ is 12, what is $7\#2$?



ChatGPT

The notation " $6\#2$ " is not a standard mathematical operation. However, if we infer that " $\#$ " represents multiplication, then " $6\#2$ " would be equivalent to 6 multiplied by 2, which equals 12. Following the same logic, " $7\#2$ " would also be equivalent to 7 multiplied by 2, resulting in 14.

Prompt Engineering: Few Shots Example



You

If $6\#2$ is 10, what is $7\#2$?

One shot is not enough to answer the question.



ChatGPT

If $6\#2$ equals 10, then it appears that " $\#$ " represents a mathematical operation other than addition, subtraction, multiplication, or division, since none of those operations result in 10 when applied to 6 and 2. Without further context or information about the specific operation denoted by " $\#$ ", it's challenging to determine what it represents. Could you please clarify or provide additional details about the operation?



You

If $6\#2$ is 10; $6\#3$ is 16; $6\#4$ is 22; and $7\#3$ is 19 then what is $7\#4$?



ChatGPT

- Given the provided information, let's analyze the pattern: $6\#2 = 10$; $6\#3 = 16$; $6\#4 = 22$; $7\#3 = 19$
- Observing the pattern, it seems that the result is obtained by adding a constant number to the previous result, with the constant increasing by 6 each time.
- From $6\#2$ to $6\#3$, the constant added is 6
- Then, from $6\#3$ to $6\#4$, the constant added is 6.
- However, from $6\#4$ to $7\#3$, the constant added is 7 as the first operand has increased by 1
- So, to find $7\#4$, we should add 7 to $7\#3$:
- $7\#4 = 19 + 7 = 26$, therefore, $7\#4 = 26$.

Prompt Engineering: Concerns - 1

- Hallucinations
 - LLMs are generative. I.e., they generate a response by “guessing” each correct word in sequence
 - Unfortunately, that means that they can go off-track - one wrong guess and each subsequent guess is then in the wrong direction
- Copyright
 - Unclear whether the output of an LLM can be copyrighted
 - Unclear whether the data used for training by LLMs were covered by copyright
 - Numerous court cases in progress!

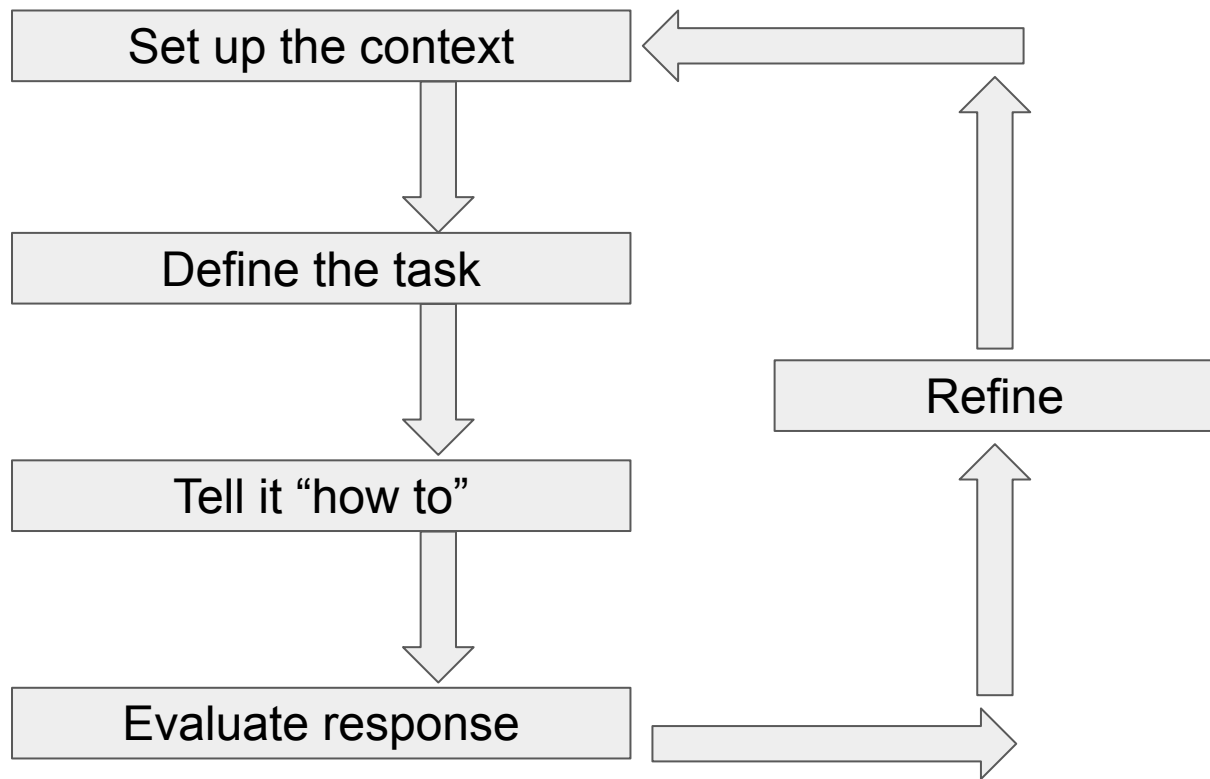
Prompt Engineering: Concerns - 2

- Security
 - LLMs typically use all inputs for enhancing their models
 - This effectively means that everything you type into an LLM is “kind of” absorbed into its network and is potentially available as information to all users
 - You can ask chatGPT not to use your data for training purposes
 - Click on the model selection icon
 - Scroll down and enable the temporary chat option
 - Note: OpenAI keeps changing the way they do this!
 - Be wary of sensitive corporate information or personal information!

Prompt Engineering: General Concepts

- Set the context carefully
- Be precise
- Specify the format of the output
 - Number of lines/words
 - Structure of output
- Start simple and then add more conditions/restrictions/directions

Prompt Engineering: Process



Examples

- Basic Q&A
- Writing an email to a friend
- Writing an email to a manager
- Summarizing a document
- Language translation
- Writing code (Python/Fortran)
 - Write code
 - Create unit tests
 - Fix bad code
- Database design and queries
- Image generation
- Video generation

Example: An Email to Your Manager

- **Context**

My parents are celebrating their 50th wedding anniversary next week. They live in California and I live in New York and I plan to fly out on Sunday and return the following Saturday so that I can be there on this important milestone. At work, I manage a small team of four people and we are in the middle of an important project with a deadline one month away.

- **Task**

Could you please write an email to my manager asking her if I could get a week's leave?

- **How to**

My manager's name is Helen and she is the Chief Projects Officer Please address the email to her

Example: An Email to Your Manager - Prompt



You

My parents are celebrating their 50th wedding anniversary next week. They live in California and I live in New York and I plan to fly out on Sunday and return the following Saturday so that I can be there on this important milestone. At work, I manage a small team of four people and we are in the middle of an important project with a deadline one month away.

Could you please write an email to my manager asking her if I could get a week's leave?

My manager's name is Helen and she is the Chief Projects Officer
Please address the email to her.

Example: Response Evaluation and Refinement



ChatGPT

Dear Helen,

I trust this email finds you well.

I am reaching out to formally request a week's leave from May 19th to May 25th. The reason for this request is to attend my parents' 50th wedding anniversary celebration in California.

As you are aware, my parents live in California, while I reside in New York. Given the significance of this milestone in their lives, I believe it is essential for me to be present to celebrate with them. Therefore, I plan to fly out on Sunday, May 19th, and return the following Saturday, May 25th.

I understand the importance of my role within the team, particularly considering our ongoing project with a deadline approaching one month away. Please be assured that I have taken proactive steps to minimize any disruption caused by my absence. I have briefed my team on the current status of the project and have made arrangements for them to handle any urgent matters that may arise during my absence. Furthermore, I will be available via email and phone should any critical issues arise that require my immediate attention.

I am fully committed to ensuring that our project remains on track, and I am more than willing to provide any necessary support or assistance to my team before my departure and upon my return.

If there are any specific procedures or documentation required for this leave request, please do not hesitate to let me know, and I will ensure that everything is completed promptly.

Thank you for considering my request. Your understanding and support in this matter are greatly appreciated.

Looking forward to your response.

Warm regards,

[Your Name]

Example: Response Evaluation and Refinement - 1

Refine the email:



You

Add that I've completely briefed Qing, my Assistant Project Manager, and that I have full faith in her capabilities. Also, I only need Helen's email approval. No other procedures are necessary.

Example: Response Evaluation and Refinement - 2



ChatGPT

Dear Helen,

I trust this email finds you well.

I am reaching out to formally request a week's leave from May 19th to May 25th. The reason for this request is to attend my parents' 50th wedding anniversary celebration in California.

As you are aware, my parents live in California, while I reside in New York. Given the significance of this milestone in their lives, I believe it is essential for me to be present to celebrate with them. Therefore, I plan to fly out on Sunday, May 19th, and return the following Saturday, May 25th.

I understand the importance of my role within the team, particularly considering our ongoing project with a deadline approaching one month away. Please be assured that I have taken proactive steps to minimize any disruption caused by my absence. **I have completely briefed Qing, my Assistant Project Manager, on the current status of the project and have full faith in her capabilities to handle any tasks that may arise during my absence.** Furthermore, I will be available via email and phone should any critical issues arise that require my immediate attention.

I am fully committed to ensuring that our project remains on track, and I am more than willing to provide any necessary support or assistance **to Qing and the rest of the team** before my departure and upon my return.

Regarding the leave approval process, I understand that only your email approval is necessary, and no other procedures are required. If you could kindly confirm your approval via email at your earliest convenience, I would greatly appreciate it.

Thank you for considering my request. Your understanding and support in this matter are greatly appreciated.


Looking forward to your response.

Warm regards,

[Your Name]

Example: Response Evaluation and Refinement - 3

Refine the email:

 **You**

Helen doesn't like formal emails. Could you make this more informal?

Example: Response Evaluation and Refinement - 4



ChatGPT

Hey Helen,

Hope you're doing well!

Just wanted to give you a heads up that I'm planning to take some time off from May 19th to May 25th. My folks are celebrating their 50th wedding anniversary in California, and I really want to be there for it. They're out in sunny Cali, and I'm here in the Big Apple, so it's a bit of a hike, but totally worth it for such a big occasion.

I've already chatted with Qing, and she's totally on board with covering for me while I'm away. She's got everything under control, and I have total confidence in her. Of course, I'll be reachable by email or phone if anything urgent pops up, but I'm doing my best to make sure everything's smooth sailing while I'm gone.

I know we've got that big project coming up, but don't worry, I've made sure everything's set up for success before I take off. Plus, I'll be back before you know it, ready to dive back in and keep things rolling.

If it's cool with you, I just need a quick thumbs-up via email to finalize things on my end. No fancy paperwork or anything like that required!

Thanks a bunch for understanding, Helen. Really appreciate it!

Catch you later,

[Your Name]

Example: Response Evaluation and Refinement - 5

Refine the email:



You

Not that informal! Helen is not a hipster!

Example: Response Evaluation and Refinement - 6



ChatGPT

Hi Helen,

I hope you're doing well.

I wanted to touch base about taking some time off from May 19th to May 25th. My parents are celebrating their 50th wedding anniversary in California, and it's a big deal for our family. I'll be flying out on Sunday, May 19th, and returning the following Saturday, May 25th.

I've already spoken with Qing, and she's aware of my plans. I trust her to handle things while I'm away, and I'm confident that the team will be in good hands. Of course, I'll be reachable by email or phone if anything urgent comes up.

I understand we have that big project on the horizon, but I've made sure to tie up loose ends and ensure everything is in order before I leave. I'll be back in action before you know it, ready to jump back into things.

If it's alright with you, I just need a quick confirmation via email to finalize my plans. No need for any formalities or paperwork.

Thanks for your understanding, Helen. I really appreciate it.

Best regards,

[Your Name]

Example: Response Evaluation and Refinement - 7

Perfect!

Example: Summarizing Text



You

Summarize the contents of the page at this URL:

[https://en.wikipedia.org/wiki/Short_\(finance\)](https://en.wikipedia.org/wiki/Short_(finance))



ChatGPT

The Wikipedia page on "Short (finance)" provides an overview of the concept of short selling in finance. It explains that short selling involves selling an asset (such as a stock or commodity) that the seller does not own, with the intention of buying it back at a later time at a lower price. The page discusses the mechanics of short selling, including borrowing the asset from a broker, selling it on the market, and then buying it back to return it to the lender. It also covers the risks and potential rewards associated with short selling, as well as its role in financial markets. Additionally, the page explores various strategies and regulations related to short selling.

If you need a longer summary - ask!



You

Could you produce a 1000 word summary?

Example: Summarizing Diagrams

ChatGPT can understand diagrams!



You

Summarize the contents of the page at this url

https://en.wikipedia.org/wiki/Venn_diagram#/media/File:Venn_diagram_of_legs_and_flying.svg



ChatGPT

The Wikipedia page features a Venn diagram titled "Venn diagram of legs and flying." The diagram consists of three overlapping circles representing different groups: "has legs," "can fly," and "is a bird." The intersections of these circles illustrate relationships between the groups, such as birds having both legs and the ability to fly.

Example: Designing a Database



You

Can you suggest a set of SQL tables for an inventory control database, each with whatever attributes you think necessary. Please label each table as either an entity or a relationship.

Response Continues in ChatGPT...

Example: Writing SQL Queries



You

Using this set of tables, write an SQL query that lists all suppliers, customers and orders for a product (assume productName = "X")

Response Continues in ChatGPT...

Example: Writing Code - 1



You

Write Python Django code for creating these tables.



ChatGPT

Sure, here's how you can define these tables using Django models:

```
from django.db import models
class Order(models.Model):
    customer = models.ForeignKey(Customer, on_delete=models.CASCADE)
    order_date = models.DateField()
    shipped_date = models.DateField()
    ship_via = models.CharField(max_length=100)
    freight = models.DecimalField(max_digits=10, decimal_places=2)
```

... Many more class definitions ...

Make sure you have Django installed in your Python environment (pip install django) and that you have created a Django app where you will define these models. After defining these models, you can run python manage.py makemigrations followed by python manage.py migrate to create the corresponding database tables.

Example: Writing Code - 2



You

Write a function that returns a list of product ids for products where the quantity in stock minus the quantity on order is less than the reorder level (using the previously defined Django tables).



ChatGPT

```
from .models import Product, OrderDetail

def get_products_to_reorder():
    products_to_reorder = []

    # Retrieve all products
    products = Product.objects.all()

    for product in products:
        # Calculate the quantity available in stock
        quantity_in_stock = product.quantity_in_stock

        # Calculate the total quantity on order
        total_quantity_on_order =
OrderDetail.objects.filter(product=product).aggregate(total_quantity=models.Sum('quantity'))['total_quantity']
        if total_quantity_on_order is None:
            total_quantity_on_order = 0

        # Check if the quantity available in stock minus the quantity on order is less than the reorder level
        if quantity_in_stock - total_quantity_on_order < product.reorder_level:
            products_to_reorder.append(product.id)

    return products_to_reorder
```

Example: Debugging Code



You

Could you tell me if there are any errors in this code?

```
def next_pres(name):  
    names = ["Washington", "Adams", "Jefferson", "Madison", "Munro", "Quincy  
Adams"]  
    for i in range(len(names)):  
        if names[i] == name:  
            return names[i+1]  
    return name + " not found in the list"
```

Example: Debugging Code Response - 1

- ChatGPT finds the error easily




Yes, there is a potential error in the code. If the input name is the last name in the list of presidents, the code will attempt to access `names[i + 1]`, where `i` is the index of the last name in the list. This will result in an `IndexError` because there is no element at index `i + 1` in the list when `i` is the index of the last element.

To handle this scenario gracefully, you should add a condition to check if the input name is the last name in the list before attempting to access `names[i + 1]`. If it is the last name, you can return a message indicating that there are no more names after it.

Example: Debugging Code Response - 2

- And provides the corrected code

 ChatGPT

```
def next_pres(name):  
    names = ["Washington", "Adams", "Jefferson", "Madison", "Monroe",  
    "Quincy Adams"]  
    for i in range(len(names)):  
        if names[i] == name:  
            if i < len(names) - 1:  
                return names[i + 1]  
            else:  
                return "No president found after " + name  
    return name + " not found in the list"
```

Note that chatGPT has figured out that we're dealing with presidents!

Example: Documentation



You

Thank you. Could you add detailed documentation to the code?

```
def next_pres(name):  
    """  
    Finds the next U.S. president in the list of presidents after the given name.  
  
    Parameters:  
    name (str): The name of a U.S. president to find the next president after.  
  
    Returns:  
    str: The name of the next U.S. president after the given name.  
        If the given name is not found in the list of presidents, returns a message indicating that the name was not  
found.  
        If the given name is the last president in the list, returns a message indicating that there are no more  
presidents after it.  
    """  
  
    # List of U.S. presidents  
    names = ["Washington", "Adams", "Jefferson", "Madison", "Monroe", "Quincy Adams"]  
  
    # Iterate through the list of names  
    for i in range(len(names)):  
        # Check if the current name matches the input name  
        if names[i] == name:  
            # If the input name is found, check if it is the last name in the list  
            if i < len(names) - 1:  
                # If not the last name, return the next name in the list  
                return names[i + 1]  
            else:  
                # If the last name, return a message indicating there are no more names after it  
                return "No president found after " + name  
  
    # If the input name is not found in the list, return a message indicating it was not found  
    return name + "not found in the list"
```


ChatGPT Plus

- Read and interpret images and pdf files
- Read audio files
- Use your device camera to examine and interpret your surroundings
- Generate images with OpenAI's Dall-E tool

Example: Load and Interpret Images

Task: Load an image and ask ChatGPT to identify objects in it



You



What bird is in this image?



ChatGPT Plus

The bird in the image is a Hoatzin (*Opisthocomus hoazin*). Hoatzins are distinctive birds found in the swamps, riverine forests, and mangroves of the Amazon and the Orinoco Delta in South America, including Colombia. They are notable for their unique appearance, with a spiky crest on their head, blue facial skin, and reddish-brown plumage. The Hoatzin is also known for its unusual digestive system, which ferments vegetation in its crop, similar to how ruminants digest food.

Example: Image Generation



You

Generate an image that contains this Hoatzin in Times Square



ChatGPT Plus



Here is the image of a Hoatzin bird in Times Square. The contrast between the bird's natural beauty and the vibrant urban setting creates a striking visual.

Example: Data Analysis

LLMs can run fairly advanced data analysis techniques

- As an example, we'll run a regression on housing data
- The data (sample):

Longitude	Latitude	Median Age	Total Rooms	Total Bedrooms	Population	Households	Median Income	MedianHomeValue
-122.23	37.88	41	880	129	322	126	8.3252	452600
-122.22	37.86	21	7099	1106	2401	1138	8.3014	358500
-122.24	37.85	52	1467	190	496	177	7.2574	352100
-122.25	37.85	52	1274	235	558	219	5.6431	341300
-122.25	37.85	52	1627	280	565	259	3.8462	342200

Example: Data Analysis - prompt



You

Read the data in the csv file and run a regression that uses MedianHomeValue as the dependent variable and Median Income, Population, and Households as independent variable. Report the r-square, the intercept and the beta coefficients.



ChatGPT Plus

Here are the results of the regression analysis:

- **R-squared:** 0.5125
- **Intercept:** 41,226.46
- **Beta Coefficients:**
 - Median Income: 41,557.80
 - Population: -45.84
 - Households: 140.36

These results indicate that about 51.25% of the variance in Median Home Value can be explained by the Median Income, Population, and Households.

Example: Data Analysis - visualization

- The results are quite good. More than 50% of the variance in home prices is explained by median income, population, and the number of households in a block
- We can ask the LLM to help us interpret the results
- Asking ChatGPT to “visualize the results with plots” results in three plots
 - Plot 1: Observed vs Predicted median home values
 - Plot 2: Residuals vs Predicted median home values
 - Plot 3: Distribution of residuals
- ChatGPT also explains how to interpret each of these plots (and you can ask for further explanation if necessary)

Example: Data Analysis - plot 1

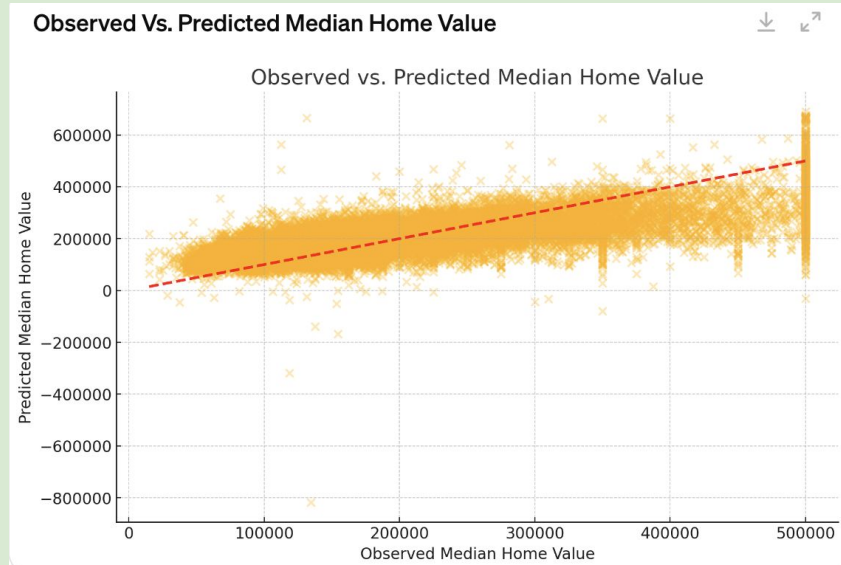


You

Visualize the regression results with plots.



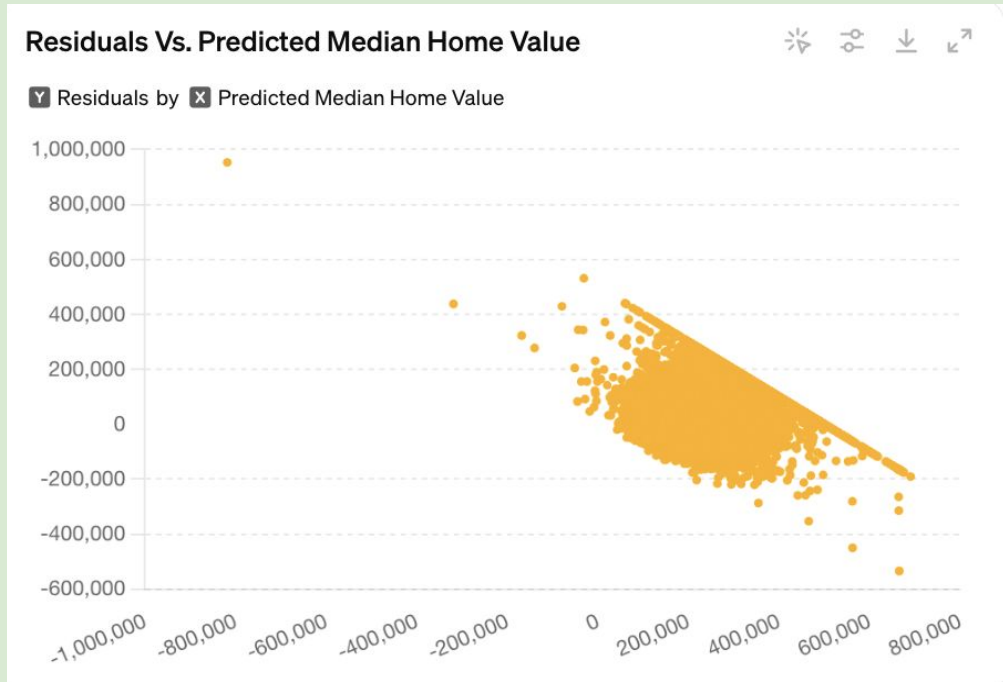
ChatGPT Plus



Observed vs. Predicted Median Home Value:

This scatter plot shows the relationship between the observed and predicted values. The red dashed line represents the ideal scenario where the observed values equal the predicted values.

Example: Data Analysis - plot 2

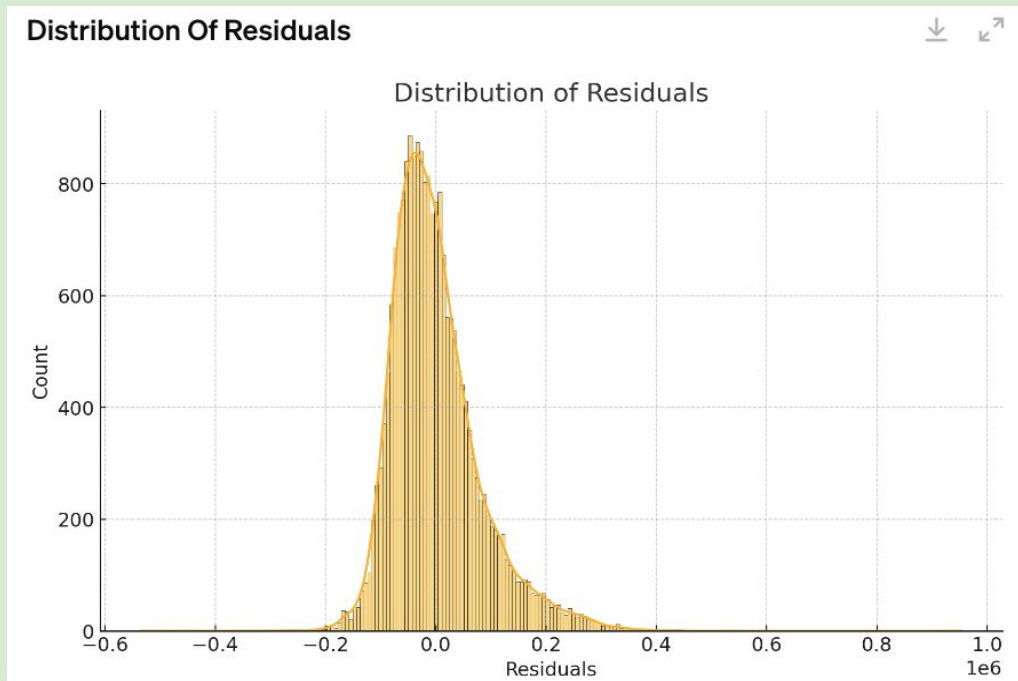


Residuals vs. Predicted Median Home Value:

This scatter plot displays the residuals (the difference between the observed and predicted values) against the predicted values. The red dashed line at zero helps identify any patterns in the residuals.

Example: Data Analysis - plot 3

 ChatGPT Plus



Distribution of Residuals:

This histogram, along with the kernel density estimate (KDE) curve, shows the distribution of the residuals. Ideally, the residuals should be normally distributed around zero.

Example: Data Analysis - explained

- We can see that the model fits the data but not perfectly
 - In the first plot, look at the angle of the orange dots (the actuals) vs the angle of the red line (the predictions)
 - In the second plot, ideally, the residuals should be clustered around zero, ours are skewed to the right
 - In the third plot, we should see a normal distribution. Our distribution is somewhat normal but is skewed toward the right (long tail)
- We can ask ChatGPT to help us understand these graphs

Example: Data Analysis - understand plots



You

Could you use the first plot (observed vs predicted median home values) to give me a sense for how good my results are? In about 100 words.



ChatGPT Plus

The observed vs. predicted median home values plot shows a moderate fit, with an R-squared value of 0.5125, indicating that the model explains about 51.25% of the variance.

While many points align around the ideal $y=x$ line, there's noticeable scatter, suggesting variability in the residuals. This indicates that the model captures some but not all of the data's variance.

To improve the model, consider adding more relevant features, exploring non-linear relationships, or using advanced regression techniques.

Overall, the model provides a reasonable but not perfect prediction of median home values.

Example: Data Analysis - follow up 1



You

Could you rerun the regression after standardizing the variables and then tell me how important each of the features is?



ChatGPT Plus

- This proves too complicated for the LLM
- So, chatGPT gives a list of alternative methods for standardizing the data and calculating feature importance

Advanced Data Analysis

- Since ChatGPT Plus could not standardize the data for us, we need to write some code to do the standardization
- A new file, `standardized_data.csv` contains the standardized data
- Let's see what ChatGPT can do with this data

Modeling with ChatGPT: Example - 1



You

The first row of this file contains column names. Could you split the data into 70% training and 30% testing, run a regression on the training and report the r-square and rmse on the testing data? The dependent variable is MedianHomeValue.



ChatGPT

The linear regression model on the training data resulted in an R^2 (coefficient of determination) of approximately 0.536 on the testing data, indicating that about 53.6% of the variance in the median home value is explained by the model. The root mean square error (RMSE) is about \$78,006, which quantifies the average error in predictions of the median home value.

Modeling with ChatGPT: Example - 2



You

Is the R-square value considered good?

Response Continues in Next Slide...



ChatGPT

The acceptability of an R^2 value, such as the 0.536 obtained in your regression analysis, depends heavily on the context and the complexity of the data:

1. **Context and Domain:** In some fields, such as physical sciences, an R^2 value less than 0.9 might be considered poor because the relationships between variables are often well-understood and highly predictable. In contrast, in social sciences or economics, an R^2 value of 0.5 might be considered quite good due to the inherent variability and complexity of the data.
2. **Purpose of the Model:** If the model is used for explanatory purposes, to understand which factors are significant predictors of the outcome, a moderate R^2 might still be very useful. However, if the model is intended for predictive purposes, where accurate forecasts are crucial (such as in real estate pricing), a higher R^2 would generally be preferable.
3. **Comparison to Baseline:** Sometimes, it's helpful to compare the R^2 value to that of a simpler model, such as one that predicts the mean for all observations. If the R^2 value significantly improves upon such a baseline, it indicates that the model is capturing useful patterns in the data.
4. **Variability of the Dependent Variable:** The more variable the dependent variable (in your case, Median Home Value), the harder it might be to achieve a high R^2 . Considering the variability of home prices due to numerous influencing factors (location, size, market conditions, etc.), an R^2 of 0.536 might be more meaningful.

In summary, while an R^2 value of 0.536 isn't typically considered high, it might still be considered adequate depending on the specific objectives of your analysis and the variability inherent in your data. To enhance this, exploring more complex modeling techniques or refining the feature set could be beneficial, particularly if higher predictive accuracy is desired.

Modeling with ChatGPT: Example - 3



You

What would you suggest I do next. For context, I'm trying to predict home values in California.

Response Continues in Next Slide...



ChatGPT

Given that you're focusing on predicting home values in California, here are some strategies to potentially improve your model's accuracy and robustness:

1. Expand Your Feature Set:

- Geographical Features: Since location is a critical determinant of home prices, consider including more granular location data, such as zip codes, neighborhood indicators, or proximity to key amenities (schools, parks, commercial centers).
- Economic Indicators: Local employment rates, median household income, and economic growth indicators can provide insight into the purchasing power and demand trends in different areas.
- Physical Characteristics: Detailed features about the home, such as its age, condition, style, presence of a garden or pool, and recent renovations, could significantly impact value.

2. Advanced Modeling Techniques:

- Non-linear Models: Explore models that can capture non-linear relationships, such as decision trees, random forests, or gradient boosting machines. These can often handle complex interactions between features better than linear models.
- Neural Networks: If you have a sufficiently large dataset, deep learning models might be able to capture intricate patterns that other models miss.
- Regularization Techniques: Models like Ridge or Lasso regression can help manage multicollinearity and feature selection automatically, potentially improving model performance.



ChatGPT

3. Data Quality and Engineering:

- Feature Engineering: Try transforming your features, such as by creating polynomial features, interaction terms, or aggregating certain features to see if they provide more predictive power.
- Scaling and Normalization: Ensure your data is properly scaled or normalized, as this can significantly impact the performance of many machine learning models, especially those that are distance-based.

Domain-Specific Research:

- Market Trends: Stay updated on real estate market trends in California, such as changes in housing laws, tax implications, and major economic events, which could influence home values and should be considered in the model if possible.

Consultation with Experts:

- Consider consulting with real estate experts or economists who specialize in the California market to get insights into factors that might be unique to the region and not readily apparent from the data alone.

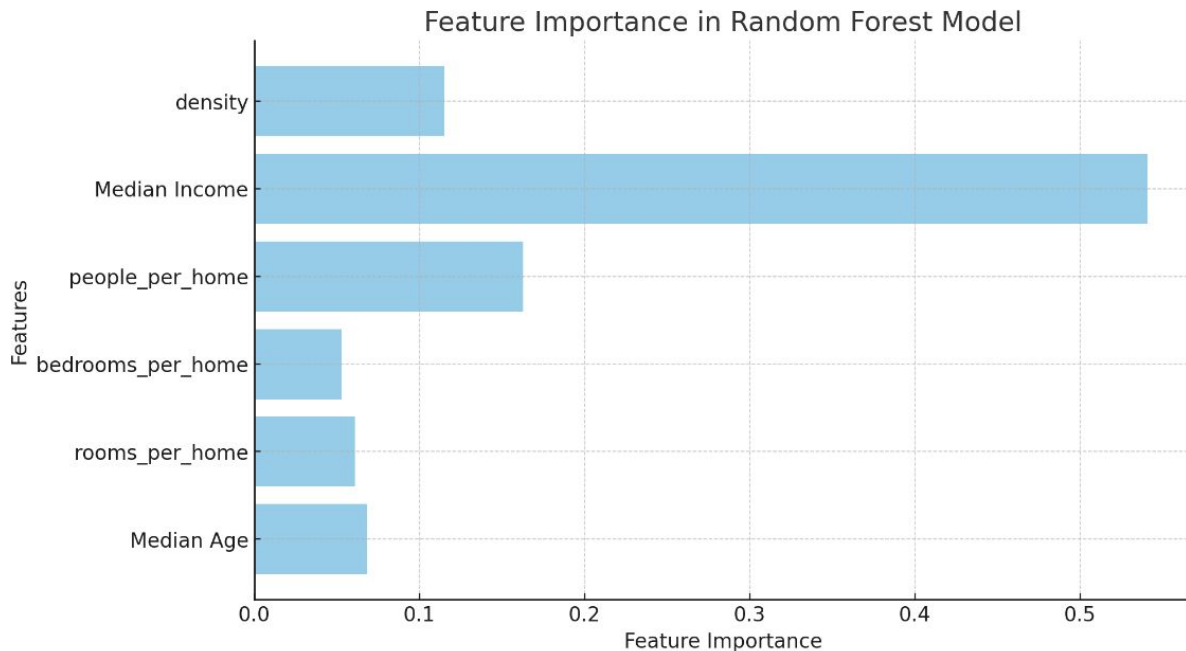
Modeling with ChatGPT: Example - 4



You

Could you use the same data split, 70% training and 30% testing, and using a random forest model draw a graph that shows feature importance?

Response Continues in Next Slide...



The bar graph above illustrates the importance of each feature in the Random Forest model used to predict home values in California. Features with higher bars have a greater impact on the model's predictions, indicating that these features are more influential in determining home values. This visualization can help you focus on the most relevant features when further refining your model or when collecting additional data. [\[>-\]](#)

Modeling with ChatGPT: Example - 5



You

Could you create a new data column, call it `value_category`. Populate it as follows: 1) calculate the median of the `MedianHomeValue` column. 2) If a `MedianHomeValue` is greater than the calculated median, the `value_category` value should be 1. Otherwise it should be zero.



ChatGPT

The new column **`value_category`** has been successfully created and populated based on the median of the **`MedianHomeValue`**. In this new column, homes with a value greater than the median have been marked with a **1**, and those at or below the median are marked as **0**. The data now includes this category, which can be used for further analysis or classification tasks.

Modeling with ChatGPT: Example - 6



You

Run a random forest classifier with `value_category` as the dependent variable and the same 70% training and 30% testing split. Report the accuracy, precision, recall, and AUC on the testing sample. Please drop the `MedianHomeValue` column for this analysis.

Response Continues in Next Slide...



ChatGPT

The Random Forest classifier model yielded the following performance metrics on the testing sample:

- Accuracy: 85.89%
- Precision: 87.12%
- Recall: 84.50%
- AUC (Area Under the Curve): 93.66%

These results indicate a strong performance of the model in classifying homes into categories based on whether their values are above or below the median. The high AUC score suggests that the model does a good job at ranking predictions rather than merely classifying, meaning it is effective at distinguishing between the two categories. Precision and recall are also reasonably balanced, suggesting good performance both in terms of how many selected items are relevant (precision) and how many relevant items are selected (recall).

Modeling with ChatGPT: Example - 7



You

Could you draw the ROC curve?

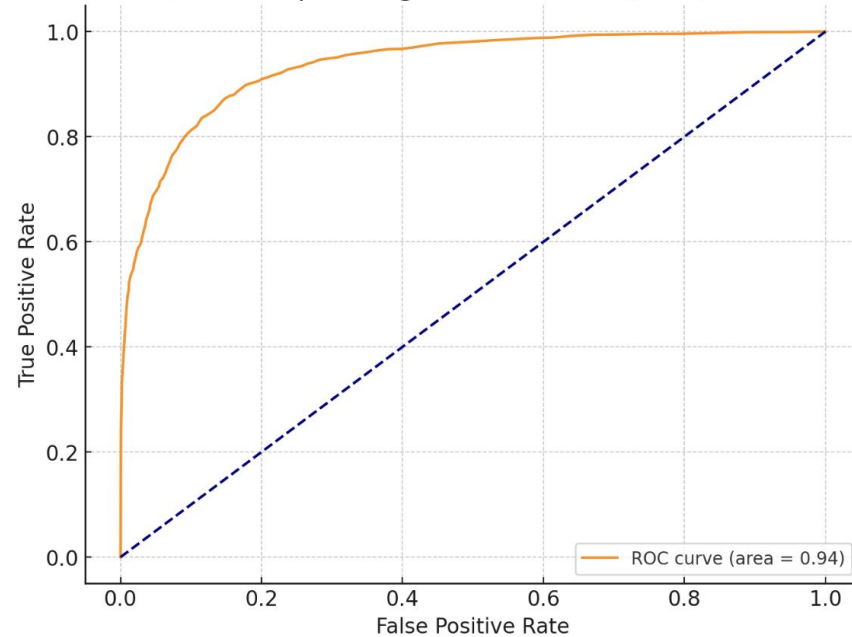
Response Continues in Next Slide...



ChatGPT



Receiver Operating Characteristic (ROC) Curve



Here's the Receiver Operating Characteristic (ROC) curve for the Random Forest classifier model. The curve demonstrates the trade-off between the true positive rate (TPR) and false positive rate (FPR) at various threshold settings. The area under the curve (AUC) of approximately 0.94 indicates a high level of model performance, with a strong ability to differentiate between the two categories of home values. [-]

Specialized GPTs

- A large number of specialized GPTs, created by users, exist for specialized tasks
- And many more are being constantly created
- Example repository: <https://www.gpt.ai/> (caveat emptor!)

Example: Starting a New Brewery - 1

- You're starting a new craft brewery and need help coming up with marketing strategies, a name, a logo, etc. for your beer
- You decide to use the specialized models on gpt.ai to help you launch the business

Example: Starting a New Brewery - 2

- gpt.ai has numerous marketing resources
 - ChatGPT Marketing Ideas Bot: for generating a marketing strategy
 - Elegant logo creator: to create a brand logo for your beer

Marketing Ideas Bot



You

I've created a craft beer in my garage. The beer is meant for IPA lovers and is fruity with a medium malt flavor and has the aroma of pine, berries. A mountain motif IPA! I expect the price point to be on the high side. Can you create a marketing plan for me?



ChatGPT

- Returns a marketing plan
- Comes up with name suggestions
- Ideas for
 - Target audience
 - Story
 - Website
 - branding/packaging
 - Experiential marketing
 - And much more ...

Marketing Ideas Bot: Output Examples

ChatGPT

Name: Choose a name that evokes the mountain theme and the beer's unique flavors. Examples: "Peak Pine IPA," "Berry Summit IPA," or "**Mountain Ridge IPA.**"

ChatGPT

Logo and Label Design: Design a logo and label that are visually striking and **reflect the mountain theme. Use imagery of mountains, pine trees, and berries. The design should convey a sense of adventure and quality.**

Elegant Logo Bot - 1



You

I've created a craft beer in my garage. The beer is meant for IPA lovers and is fruity with a medium malt flavor and has the aroma of pine, berries. A mountain motif IPA! I expect the price point to be on the high side. I've chosen the name "Mountain Ridge IPA". Can you design a logo that reflects the mountain theme. Use imagery of mountains, pine trees, and berries. The design should convey a sense of adventure and quality.



ChatGPT



Here is the logo design for "Mountain Ridge IPA" featuring mountains, pine trees, and berries. The design is minimalistic and elegant, conveying a sense of adventure and quality. If you would like any adjustments or further refinements, please let me know.

Elegant Logo Bot - 2



You

I'd like something that will stand out on a dark beer bottle.



ChatGPT



Here is the updated logo design for "Mountain Ridge IPA" that will stand out on a dark beer bottle. The contrasting colors and bold typography ensure visibility and convey a sense of adventure and quality. If you need any further adjustments, please let me know.

Bottom Line! - 1

- You can do a lot of things with large language models
 - Get factual information
 - Ask it to reason for you
 - Write email, essays, marketing plans, for you
 - Do statistical and mathematical calculations
 - And much more...

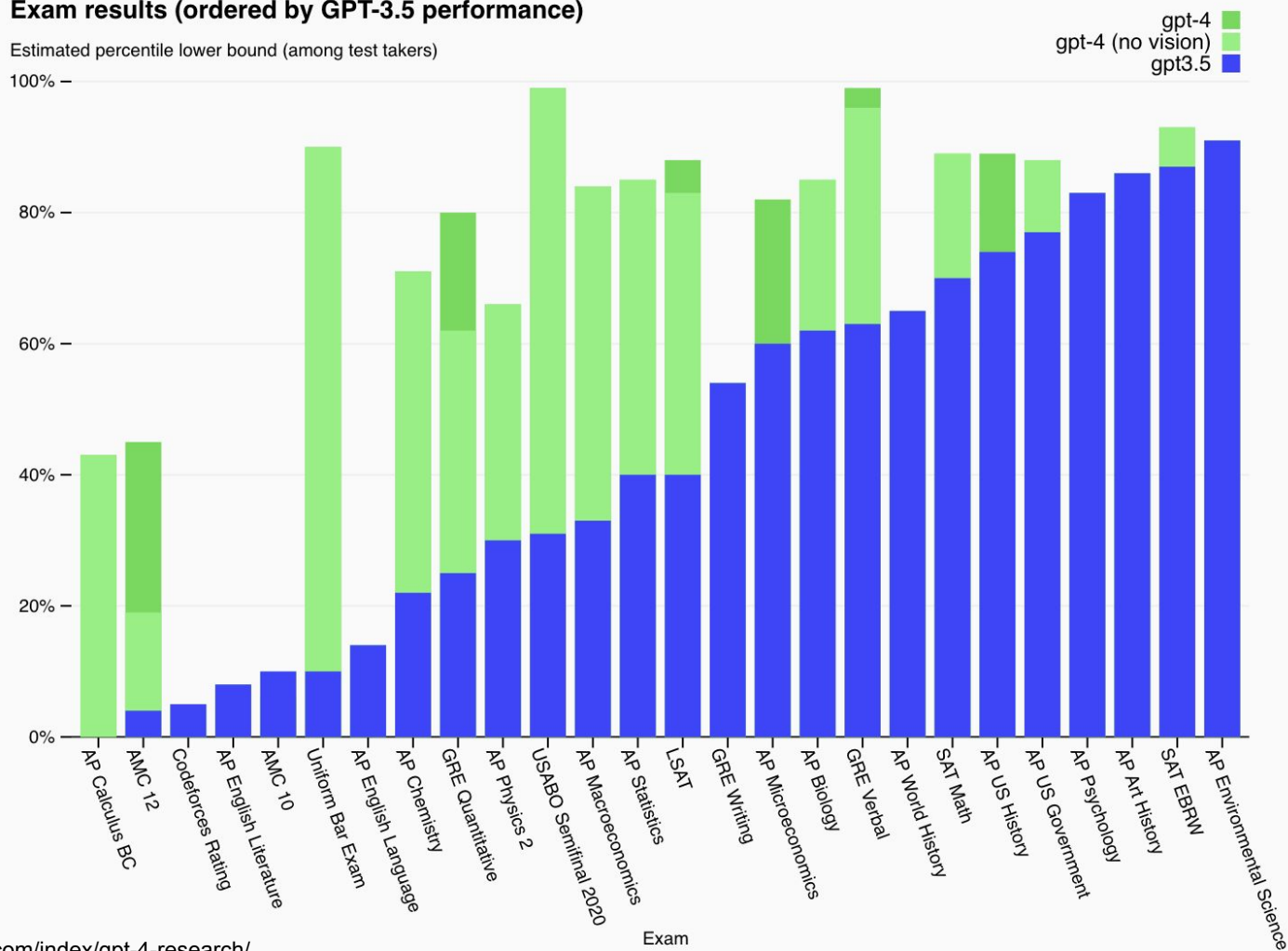
Bottom Line! - 2

- Large language models are getting better and will keep improving

See performance graph on next slide

Exam results (ordered by GPT-3.5 performance)

Estimated percentile lower bound (among test takers)



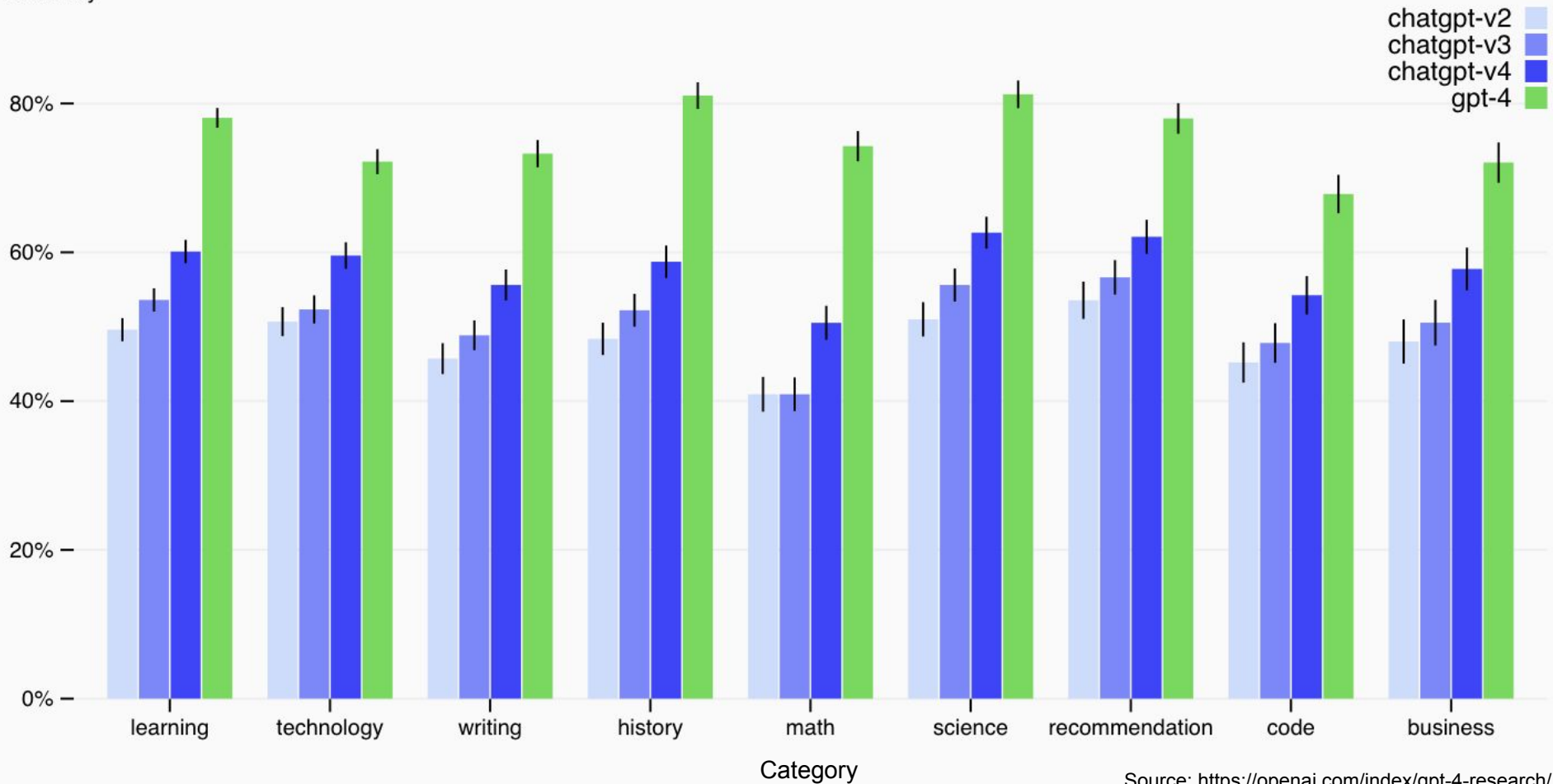
Bottom Line! - 3

- But they still make mistakes

See performance graph on next slide

Internal factual eval by category

Accuracy



Chain of Thought

Chain of Thought

Guides models to think in steps before giving the final answer.

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

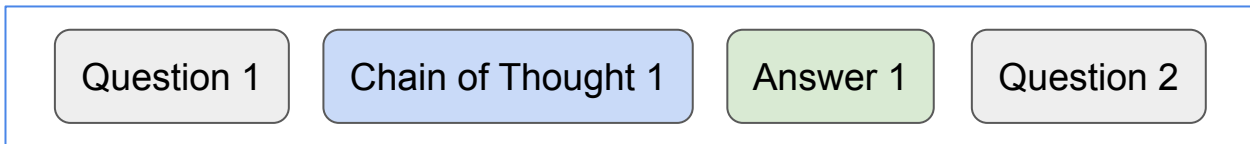
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

Source: [Wei et al. \(2022\)](#)

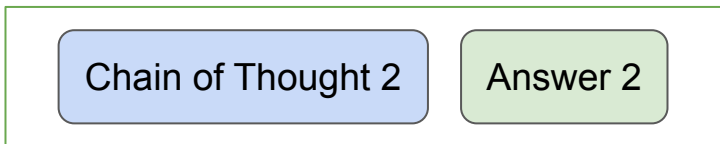
Chain of Thought

Generating a chain of thought—a series of intermediate reasoning steps—significantly improves the ability of LLMs to perform complex reasoning.

You:



LLM:



- Improves multi-step reasoning
- Works best in larger language models
- Makes reasoning more interpretable

Reasoning LLMs

Unlike standard LLMs, reasoning models generate a long internal chain of thought before producing an answer.

You:

Question

Reasoning LLM:

Chain of Thought

Answer

- Produce richer intermediate reasoning steps (Reinforcement Learning)
- Better at complex problem-solving, coding, scientific tasks
- Excel in multi-step planning and logic

Multi-turn Conversation

Turn 1

Question 1

CoT 1

Answer 1

Turn 2

Question 1

Answer 1

Question 2

CoT 2

Answer 2

Turn 3

Question 1

Answer 1

Question 2

Answer 2

Question 3

CoT 3

Answer 3

Prompting Advice

Reasoning models perform best with straightforward prompts.

Structure:

Goal

Context

Example:

Help me plan a 15 minute online course on reasoning LLMs such as DeepSeek R1 and OpenAI o1. My target audiences are technical professionals seeking to develop AI skills. Please include coding examples in python. Use the attached knowledge:



knowledge.txt

Document

Summarized CoT of o1 Model

Reasoned about LLMs course planning for 6 seconds ^

Charting the course

OK, let me see. Crafting a 15-minute online course on reasoning LLMs, targeting technical professionals, and integrating Python coding examples. Drawing from the "DeepSeek R1" and "OpenAI o1" knowledge.

Formulating the outline

I'm crafting a 5-step plan for a brief course on reasoning LLMs, interweaving OpenAI o1 and DeepSeek R1 Python examples, and ensuring diverse perspectives are covered.

Navigating code insights

Integrating Python examples and emphasizing best practices, I'm mapping out references and ensuring clarity with precise code details, while considering recent updates.

Complete CoT of R1 Model

🖼 Thought for 29 seconds ^

Alright, the user wants me to help plan a 15-minute online course on reasoning LLMs like DeepSeek R1 and OpenAI o1 for technical professionals looking to develop AI skills. They want coding examples in Python and to use the knowledge from the provided file. Let me start by understanding the key points from the knowledge file.

First, the file describes reasoning models like OpenAI's o1 and o3-mini, emphasizing their use of reinforcement learning and chain-of-thought reasoning. It mentions parameters like 'reasoning_effort' and how these models handle complex problem-solving, coding, and STEM tasks. There are code examples in JavaScript, Python, and bash, but the user wants Python examples. The DeepSeek R1 model is also discussed, highlighting its CoT reasoning accessible via API and specific parameters.

I need to structure a 15-minute course. Let's break it down:

1. **Introduction (2-3 minutes):** Briefly explain reasoning LLMs, their purpose, and key models (OpenAI o1, o3-mini, DeepSeek R1).

2. **Core Concepts (3-4 minutes):** Discuss chain-of-thought reasoning, parameters like 'reasoning_effort' and 'max_completion_tokens', and how these models differ from traditional LLMs.

3. **Coding Examples (6-7 minutes):** Provide Python examples for both OpenAI and DeepSeek models. Use cases could include code refactoring, problem-solving (like transposing a matrix), and handling reasoning tokens.

4. **Best Practices and Q&A (2-3 minutes):** Tips for prompting, managing context windows, and cost control. Address potential questions.

Need to ensure the examples are concise and relevant. For OpenAI, use the provided Python examples from the file, maybe adapt the matrix transpose script. For DeepSeek, use their API example, ensuring the Python code is correct. Highlight differences in parameters and handling reasoning content.

Check if the user provided any specific API endpoints or authentication details, but since they didn't, keep examples generic with placeholders for API keys.

Make sure to mention the 'reasoning_effort' parameter, managing tokens, and possibly show how to handle responses that include reasoning content for DeepSeek.

Also, include a slide or section on when to use reasoning models vs traditional LLMs, based on the advice from the knowledge file about treating them like senior vs junior coworkers.

Finally, ensure the course is engaging, sticks to time, and provides actionable insights for technical professionals to start using these models.

Very long, we won't read it here.

The point is you can access the complete CoT.

Complete CoT also available via API.

Check notebook for more details.

Model Response Comparison - Intro

Course Outline (15 minutes)

1. Overview of Reasoning LLMs (2 minutes)

- Define what “reasoning” models are: Large Language Models trained with reinforcement learning to perform complex multi-step reasoning.
- Emphasize how DeepSeek R1 and OpenAI o1 produce a chain of thought internally before responding.
- Mention key benefits for coding, mathematics, and STEM research tasks.

2. Key Features & Differences Compared to Typical GPT Models (3 minutes)

- **Reasoning tokens:** Explain that these models generate extra “reasoning tokens” under the hood to plan answers.
- **Extended context:** They often require higher context length for reasoning; be prepared to increase max token usage.
- **Reasoning effort** (OpenAI o1) or CoT length (DeepSeek R1): Adjusting how “deeply” the model thinks can improve accuracy but use more tokens.

15-Minute Online Course: Mastering Reasoning LLMs (DeepSeek R1 & OpenAI o1)

Target Audience: Technical professionals seeking AI development skills

1. Introduction to Reasoning LLMs (2 min)

- **What are Reasoning LLMs?**
 - Models like OpenAI o1/o3-mini and DeepSeek R1 use **chain-of-thought (CoT) reasoning** to solve complex tasks (coding, math, planning).
 - Trained via reinforcement learning; prioritize multi-step problem-solving over simple text generation.
 - **Key Models:**
 - **OpenAI o1:** High accuracy for complex tasks.
 - **DeepSeek R1:** Open-source, provides explicit CoT output.
-

2. Core Concepts (3 min)

- **Chain-of-Thought (CoT):**
 - Models generate internal reasoning tokens (invisible) before final output.
 - Example: Solving a math problem → model first writes equations, then answers.
- **Parameters:**
 - `reasoning_effort` (OpenAI): Controls token allocation (`low` , `medium` , `high`).
 - `max_completion_tokens` (DeepSeek): Limits total tokens (reasoning + output).

Model Response Comparison - Coding

3. Example: Using OpenAI o1 via Python (3 minutes)

1. Installation:

```
bash

pip install openai
```

2. Sample Python Script:

```
python

from openai import OpenAI

client = OpenAI(api_key="YOUR_API_KEY") # Replace with your key

prompt = """
Write a bash script that takes a matrix [1,2],[3,4],[5,6]
and prints the transpose in the same format.
"""

response = client.chat.completions.create(
    model="o1", # or "o1-preview" if that's the preview version
    reasoning_effort="medium",
    messages=[
        {
            "role": "user",
            "content": prompt
        }
    ]
)

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

- `reasoning_effort`: Sets how many reasoning tokens the model uses (low, medium, high).
- Keep in mind the context window to allow enough space for reasoning tokens.

3. Coding Examples (7 min)

Example 1: OpenAI o1 for Code Refactoring

```
python

from openai import OpenAI
client = OpenAI()

response = client.chat.completions.create(
    model="o3-mini",
    reasoning_effort="high", # Allocate more tokens for complex reasoning
    messages=[
        {
            "role": "user",
            "content": """
Refactor this React component to style nonfiction books in red.
Return only the code, no markdown:

const books = [ { title: 'Dune', category: 'fiction' }, ... ];
export default function BookList() { ... }
            """
        }
    ]
)

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


Model Response Comparison - Coding

4. Example: Using DeepSeek R1 via Python (3 minutes)

1. Installation:

```
bash
pip3 install -U openai
```

(DeepSeek R1 requires the updated OpenAI-compatible SDK, as mentioned in the knowledge base.)

2. Sample Python Script:

```
python
from openai import OpenAI

# Replace with your DeepSeek API Key and base URL
client = OpenAI(api_key="<DeepSeek API Key>", base_url="https://api.deepseek.com")

messages = [
    {
        "role": "user",
        "content": "9.11 and 9.8, which is greater?"
    }
]

response = client.chat.completions.create(
    model="deepseek-reasoner",
    messages=messages
)

# Extract the chain of thought (reasoning_content) and the final content
reasoning_content = response.choices[0].message.reasoning_content
content = response.choices[0].message.content

print("Reasoning Content:\n", reasoning_content)
print("Final Answer:\n", content)
```

- DeepSeek's `reasoning_content` returns the internal chain of thought.
- The `content` field returns the final user-visible answer.

Example 2: DeepSeek R1 for Math Reasoning

```
python
from openai import OpenAI
client = OpenAI(base_url="https://api.deepseek.com", api_key="YOUR_KEY")

# Multi-turn conversation with CoT access
messages = [{"role": "user", "content": "9.11 vs 9.8: which is larger?"}]
response = client.chat.completions.create(model="deepseek-reasoner", messages=messages)

# Extract CoT and final answer
reasoning = response.choices[0].message.reasoning_content
answer = response.choices[0].message.content
print(f"Reasoning: {reasoning}\nAnswer: {answer}")
```

Example 3: Matrix Transpose with OpenAI o1

```
python
response = client.chat.completions.create(
    model="o3-mini",
    messages=[
        {
            "role": "user",
            "content": "Write a Python function to transpose a matrix given as a string like '[1,2],[3,4]'. Return only code."
        }
    ]
)
print(response.choices[0].message.content)
# Output: def transpose_matrix(s): ...
```

Model Response Comparison - Conclusion

5. Best Practices & Wrap-Up (4 minutes)

- **Prompting style:** Reasoning models do better with higher-level instructions than GPT-based models, but providing clarity can still help.
- **Managing cost:** Keep an eye on token usage (especially for deep chain-of-thought). Consider parameters like `max_completion_tokens`.
- **Real-world use cases:**
 - Complex code generation/refactoring.
 - Advanced math or scientific reasoning (e.g., step-by-step solutions).
 - Agentic multi-step workflows, where the model plans out tasks before executing.
- **Next Steps:** Encourage experimentation with different `reasoning_effort` or CoT settings to see how each model handles intricate tasks.

References

- DeepSeek R1 details (model downloads, usage, local deployment, open-source checkpoints): [DeepSeek Platform](#)
- OpenAI o1 usage and docs: [OpenAI Chat Completions](#)

This short course sets a foundation for understanding and experimenting with reasoning LLMs. Participants can further explore advanced prompting strategies and integrate these models into pipelines where deeper logic and planning are essential.

4. Best Practices & Q&A (3 min)

- **Prompting Tips:**
 - Treat reasoning models like a "senior coworker"—provide goals, not step-by-step instructions.
 - Use `reasoning_effort="high"` for STEM/coding tasks.
- **Cost Control:**
 - Monitor `usage.completion_tokens_details` (OpenAI) to track reasoning token costs.
 - Set `max_completion_tokens=4000` (DeepSeek) to avoid overages.
- **Q&A:**
 - *Q: When to use reasoning models vs GPT-4?*
 - *A:* Use reasoning LLMs for multi-step logic (coding/math); GPT-4 for creative writing.

Course Summary:

- Reasoning LLMs excel at structured problem-solving.
- Use Python APIs to integrate them into workflows (code refactoring, data analysis).
- Experiment with parameters like `reasoning_effort` to optimize performance.

Resources:

- [OpenAI Documentation](#)
- [DeepSeek R1 GitHub](#)

Compare with Chat Models

Here's a high level summary, see notebook for complete response.

- Some minor misalignment:
 - DeepSeek V3 also generated examples for o3-mini only but not o1
- Otherwise almost identical – our 15 minute example is too simple
- More practical use cases:
 - Add a feature to a software – involves editing multiple files in a directory simultaneously
 - Multi-step agentic planning – a reasoning model is used as “the planner,” producing a detailed, multi-step solution to a problem and then selecting and assigning the right chat model (“the doer”) for each step

Prompting Best Practice

- Keep prompts straightforward and minimal.
 - Reasoning models are like your senior co-workers.
 - Try zero shot before few shot.
- Avoid chain-of-thought prompts.
 - They are already trained to do so. Prompts such as “think step by step” is unnecessary.
- Be specific about your goal and output format.
 - Encourage the model to keep reasoning and iterating until it meets your goal.
 - Crucial especially for complex task.

Summary

- Reasoning LLMs are very capable new models.
- Generate internal CoT before final answer.
- Require minimal prompting but also benefit from specific output instruction.
- Check notebook for coding examples and API info.