Intelligent Systems

Quiz Study Guide

Decks 1 to 4 (Sep 1 to Sep 24, 2026). Built only from the professor's slides.

How to use this guide

- **Her slides are the only source.** Every definition uses her wording. When her wording differs from what you find online, write hers on the quiz. The orange **Answer her way** boxes mark those spots.
- **Learning outcomes come first** in each section. They are her own list of what you must be able to do, so they predict the questions.
- **Image slides are written out as text.** The timeline, the pipeline diagrams and the code screenshots now appear as text you can memorize.
- **Every code output in this guide was run in Python**, so the answer keys are exact.
- **Videos are not covered.** The slides only link to them.

Two-day plan

When	What to do
Day 1, evening	Read Decks 1 and 2. Cover the right column of each table and recite it. Do both practice sets without looking at the keys.
Day 2, morning	Read Decks 3 and 4. Trace every code block by hand before reading the output. Do the practice sets.
Day 2, evening	Do the Python tracing drill. Then read only the orange boxes and the Memorize page. Redo any question you missed.
Quiz morning	Memorize page only. Ten minutes.

What to weight most

- **Deck 4 (ML basics)** is the most recent and the most definition-heavy. Expect several questions.
- **Deck 2 (Data)** has the most lists and classifications: data types, graphs, structure, study types. These make easy multiple choice questions.
- **Deck 3 (Python)** has code with no output shown on the slide (slicing, slide 31). That is a ready-made quiz question.
- **Deck 1 (Intro)**: know the McCarthy definition, the two AI winters, the three eras and the ethical dilemmas.
- Her first-day slide sets the **first exam for 20.10.2026**, so this guide also serves as exam prep.

Deck 1

Introduction to Artificial Intelligence

Sep 1 and Sep 3, 2026

Learning outcomes (her slide, word for word)

- Define the concept of artificial intelligence
- Describe the main events that occurred before the rise of artificial intelligence

Definition of AI (slide 19)

John McCarthy (2007):

> *"It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable."*

Key phrases to recognize: **science and engineering**, **intelligent machines**, **intelligent computer programs**, **not confined to biologically observable methods**.

Course topics (slide 14)

1. Introduction to artificial intelligence 2. Machine learning 3. Supervised learning and unsupervised learning 4. Neural networks

AI history timeline (slide 20, from the image)

Period	Events above the line	Events below the line
1900–1950	1900 Triode vacuum tube · 1920 First ever robot · 1938 Digital logic · 1948 Cybernetics	**1950 Turing Test**: "A computer would deserve to be called intelligent if it could deceive a human into believing it was human"
1950–1973	**1956 AI term born**: the **Dartmouth Proposal**, the first paper where the AI definition is used (John McCarthy)	1957 First Perceptron · 1957 First AI program · 1965 First industrial robot · 1965 Eliza chatbot
1st Winter	Between 1973 and 1980	
1980–1987	1980 WABOT-2 · 1982 Hopfield ANN · 1986 Parallel processing · 1989 Convolutional NN	1986 Backpropagation: multi-layer networks could learn non-linear mappings
2nd Winter	Between 1987 and 1993	
1993–2000	**1997 Deep Blue** defeats chess world champion Garry Kasparov	1994 Robocup · 1997 LSTM ANN · 1999 Sony AIBO · 2000 Kismet robot
2000–2012	2003–2005 DARPA Challenges · 2006 Deep Belief ANN · 2008 IBM Blue Brain · 2010 Apple Siri	**2012 AlexNet**: deep learning architecture that outperforms ImageNet
2012–2017	**2017 Transformers** ("Attention is all you need")	2013 DeepMind · 2014 Generative AN · 2015 TensorFlow · **2016 AlphaGo**
2017–2020	2017 WaveNet · 2019 GPT-1 · 2020 BERT LM · 2020 GPT-3	2020 DALL-E: generating images from descriptions
2020–2025	**2022 ChatGPT**: fastest growing consumer app in history at 2 months	2020 Tesla self drive · 2021 Regulation and ethics · 2024 Sora LVM · 2024 EU AI Act

Period	Events above the line	Events below the line
2025–...	2025 DeepSeek overtakes OpenAI in 2 weeks	AGI, ASI, singularity, superhumans and beyond

Other milestones with their own slides: **Kasparov vs Deep Blue (1997)**, **AlphaGo (2016)**, **AlphaFold (2018)**.

The two AI winters (slide 21)

First AI Winter (1970s)	Second AI Winter (late 1980s–1990s)
• Caused by unmet expectations from early AI systems. • Limited computing power hindered progress. • Funding cuts due to slow advancements.	• Decline of expert systems, which were expensive, inflexible, and hard to maintain. • High costs and low scalability of AI applications. • Reduced investment and skepticism in AI research.

Memory hook: Winter 1 = **expectations, computing power, funding**. Winter 2 = **expert systems, cost and scalability, skepticism**.

Evolution of AI (slide 22)

1950–1980	1980–2010	2010–present
Artificial intelligence	**Machine learning**	**Deep learning**

Alan Turing (slides 23–24)

- Considered one of the fathers of computer science and artificial intelligence.
- Creator of the **Turing test (1950)**, a test that determines whether one is speaking to a machine or a person.
- Turing test setup: **Player A** is a computer, **Player B** is a human responder, **Player C** is the interrogator. The interrogator asks both players questions and tries to decide which is the computer.

Generative AI (slides 28–30)

- a.k.a **gen AI**.
- A type of artificial intelligence that can generate original content such as **images, text, music and video**.
- It relies on **deep learning models** to create gen AI applications.
- Examples: **StyleGAN, ChatGPT-3, GPT-4, DALL-E**.
- Example sites: thispersondoesnotexist.com and thesecatsdonotexist.com (faces and cats that are not real).
- Use cases chart (2024 vs 2025): in 2025 the top use is **therapy/companionship**, followed by organizing my life and finding purpose.

Use cases (slides 35–36)

Slide 35	Slide 36
Crime prediction Traffic accident prediction Forest fire detection Online exam supervision (proctoring)	Text autocomplete CV scanning Dynamic pricing

Ethical dilemmas of AI (slide 38)

- **Biases** (gender, age, race)
- **Decision making**
- **Privacy**
- **Property, intellectual rights**
- **Environmental impact**

Her examples

Example	What it shows
Microsoft Tay (slide 39)	A Twitter chatbot that learned from users and became racist and xenophobic. Bias learned from data.
Google Translate (slide 40)	Turkish “o bir doktor” is gender-neutral. Before: translated only as “he is a doctor”. After: shows both “she is a doctor” and “he is a doctor”. Gender bias.
Ethics of AI art (slide 41)	Property and intellectual rights.
Deepfake video (slide 42)	A generated video of a real politician next to the reference. Implication of AI.
Keyboard sound attack (slide 43)	Deep learning guesses passwords from the sound of the keys (Harrison et al., 2023). Privacy implication.

⚠ **Answer her way**

• Her definition is **McCarthy (2007)**. The year on the slide is 2007, not 1956. 1956 is when the AI term was born (Dartmouth).

• Turing test year is **1950**.

• The three eras are exact: **1950–1980 AI, 1980–2010 ML, 2010–present DL**.

• She lists **five** ethical dilemmas. Biases has three examples: gender, age, race.

Practice questions

1. Who gave the AI definition on her slide, and in what year?
2. Complete: “AI does not have to confine itself to methods that are _____.”
3. Name the three causes of the First AI Winter.
4. What kind of system declined during the Second AI Winter and why?
5. Match each period to its era: 1950–1980, 1980–2010, 2010–present.
6. What does the Turing test determine, and when was it created?
7. In what year was the term AI born, and at what event?
8. What does generative AI rely on? Name two examples from her slide.
9. List the five ethical dilemmas of AI.
10. Which ethical dilemma does the Google Translate “o bir doktor” example show?
11. Which event did Deep Blue win in 1997?
12. Is CV scanning a use case on her slides? Is forest fire detection?

Answer key

1. John McCarthy, 2007.
2. biologically observable
3. Unmet expectations from early AI systems, limited computing power, funding cuts due to slow advancements.
4. Expert systems. They were expensive, inflexible, and hard to maintain.

5. 1950–1980 Artificial intelligence, 1980–2010 Machine learning, 2010–present Deep learning.
6. Whether one is speaking to a machine or a person. 1950.
7. 1956, the Dartmouth Proposal.
8. Deep learning models. Any two of StyleGAN, ChatGPT-3, GPT-4, DALL-E.
9. Biases, decision making, privacy, property and intellectual rights, environmental impact.
10. Bias (gender).
11. It defeated chess world champion Garry Kasparov.
12. Yes to both.

Deck 2

AI Concepts, Data and Data Collection

Sep 8 and Sep 10, 2026

Learning outcomes (her slide, word for word)

- Describe the difference between the concepts of Weak AI and Strong AI.
- Describe the statistical data types.
- Describe the data types according to their structure.
- Describe the different methods to collect data.

The Chinese room argument (slides 3–5)

- An experiment proposed by **John Searle**.
- A person who does not know Chinese is in a room. Outside, a Chinese person sends him messages. The person in the room uses an **instruction manual** to respond.
- The Chinese person gets the impression he is talking to someone who speaks Chinese, but this does not mean the person knows it.
- **The point:** if a machine proves to be intelligent, for example by passing the Turing test, this does not mean that it is “intelligent” or behaves like humans as if it had a “mind”.

Weak AI vs Strong AI (slides 6–7)

Weak AI	Strong AI
Focus on performing **specific tasks**.	Ability to perform actions in the same way as a human being would do. It can solve problems and **learn new skills on its own**.
Examples: **chatbots, spam filters in emails, Google Maps, smart assistants like Siri and Alexa**.	Use of the learned knowledge in **various tasks**, and **adaptation to the environment** in case of unforeseen events.

Philosophy of AI (slide 8)

A discipline that seeks to understand the nature of AI and its implications in the **legal, social and ethical** aspects. It asks:

- Can a machine act intelligently? Can it solve a problem that a person would solve by thinking?
- Are human intelligence and machine intelligence the same thing? Is the human brain essentially a computer?
- Can a machine have a mind, mental states and consciousness like humans? Can it sense how things are?

AI, Machine Learning and Data Science (slides 10–13)

Her diagram (Braincube, slide 11) draws nested circles: **Data Science** is the outer circle, **Artificial Intelligence** inside it, and **Machine Learning** at the center. The labels say:

- **Data Science**: collection, preparation, and analysis of data. Leverages AI/ML, research, industry expertise, and statistics to make business decisions.
- **Artificial Intelligence**: technology for machines to understand, interpret, learn, and make “intelligent” decisions. Includes machine learning among many other fields.
- **Machine Learning**: algorithms that help machines improve through supervised, unsupervised, and reinforcement learning. **Subset of AI and Data Science tool**.

Machine learning (slide 12)	Data Science (slide 13)
• A subfield of AI that **focuses on statistics**. • The computer solves problems by learning from certain data and then using that experience to solve the problem in new situations. • Systems improve the way they perform a specific task as they gain experience or accumulate data.	• A **multidisciplinary** approach that combines several fields, such as machine learning, statistics, and computer science (algorithms, applications). • Focuses on analyzing **large volumes of data** to extract information, understand patterns, and subsequently make decisions.

Data vs information (slides 14–16)

Data	Information
From Latin *datum*, “something given”.	From Latin *informatio*, “to shape the mind, instruct, teach”.
Symbolic representation of a situation or scenario that **lacks meaning**.	**Processed data** that is interpreted and leads to a message.
Raw facts and figures.	Data that has been processed (in a context) to give it meaning.

Information is the result of data and context. Context refers to the unit of measurement in which the data is found. Her formula (slide 16): **Information = Data + Context + Meaning**.

Example	Data	Context
1.50 meters	1.50	meters
Black hair	black	hair
6 dogs	6	dogs

Quote on slide 23 (Mark Twain): “Data is like garbage. You'd better know what you are going to do with it before you collect it.”

Statistical data types (slides 17–21)

Type	a.k.a	Definition	Subtypes
Quantitative	numerical	Numerical values can be analyzed by obtaining statistics such as averages and percentiles.	**Discrete**: integer numbers (15, 40) **Continuous**: float numbers (3.15, 23.2)
Qualitative	categorical	Attributes that can take a numerical value but **lack mathematical meaning**.	**Nominal**: cannot be ordered (female, male) **Ordinal**: can be ordered (low, medium, high)

Her examples: **quantitative** = weight, height, number of students in a classroom. **Qualitative** = gender, ethnicity, hometown.

Graphs according to the data type (slide 22)

Data type	Graph
Nominal	**Pie chart**
Ordinal	**Bar graph**
Discrete	**Bar graph**
Continuous	**Histogram**

Data types according to the structure (slides 24–27)

Type	Description	Sources / examples
Structured	Sorted data. **Easy to handle** given their structure (data organized in **rows and columns**).	Surveys, forms, Excel sheets, databases
Unstructured	Data lack a structure. **Requires more effort to handle** given the complexity of the format.	Social networks, videos, comments, call transcripts
Semi-structured	Presents a certain structure since it employs **labels** to organize data. Eases the creation of hierarchies and groups. **Easier to store than unstructured** data.	Emails, JSON files, XML files

Data according to the collection method (slide 28)

Observational	Experimental
Data is collected by observing a subject in a **natural setting without altering the environment or any variable**. Example: people's preferences about a product or service, people's behavior in a specific place.	Data collected under **controlled conditions**. Example: the **A/B test** to evaluate a webpage design.

Types of observational studies (slides 29–33)

Study	Her definition	Her example
Cross-sectional	Observe **different groups** in a **specific period of time**.	The predominance of an illness in a population.
Case-control	Investigate the **causes** of a disease or condition by comparing two groups: one with the disease (**case**) and one without it (**control**). Identifies contributing factors.	High-fat diet and colon cancer, comparing dietary habits of people with and without cancer.
Cohort	Observe a group of people (the **cohort**) over time. The cohort is split into groups that share characteristics or exposures (behaviours, environmental factors). Groups are tracked and eventually **compared**.	Long-term effects of diet, lifestyle, and medication use on women's health.
Longitudinal	Observe **the same group** for a **longer time (e.g., years)**. Useful to find changes over time and identify trends.	Lifestyles of similar individuals to uncover factors of successful ageing.

Sources for searching datasets (slide 34)

Kaggle · Google Dataset Search · Data.Gov · Datahub.io · UCI Machine Learning Repository

⚠ **Answer her way**

• **Nominal goes with a pie chart.** Ordinal and discrete go with a bar graph. Continuous goes with a histogram. Use exactly this mapping.

• Qualitative data “can take a numerical value but lacks mathematical meaning”. A zip code or jersey number is qualitative.

• Discrete = **integer**, continuous = **float**. That is how she defines them.

• Semi-structured data is **easier to store than unstructured**. Structured data is **easy to handle**.

• Context is “the unit of measurement in which the data is found”. In “black hair”, **hair** is the context and **black** is the data.

• Cross-sectional = different groups, one moment. Longitudinal = same group, years.

• The A/B test is her example of **experimental** data, not observational.

Practice questions

1. Who proposed the Chinese room argument, and what does it say about the Turing test?
2. Classify each as weak AI or strong AI: spam filter, Siri, a system that learns new skills on its own and adapts to unforeseen events.
3. What does machine learning focus on, according to her slide?
4. Name the fields Data Science combines.
5. Data comes from the Latin word ____ meaning ____.
6. In “6 dogs”, which part is the data and which is the context?
7. Classify: number of students in a classroom, height, hometown, shirt size (S, M, L), eye colour.
8. Which graph does she assign to continuous data? To nominal data?
9. A satisfaction rating of low, medium, high. Which data type and which graph?
10. Classify by structure: a JSON file, a call transcript, an Excel sheet, an email.
11. Why is semi-structured data easier to organize than unstructured data?
12. An A/B test on a webpage design is which collection method?
13. Comparing people with lung disease against people without it to find causes. Which study type?
14. Following the same group of people for 20 years to find trends. Which study type?
15. Measuring how common an illness is in a population at one point in time. Which study type?
16. Name three dataset sources from her slide.

Answer key

1. John Searle. Passing the Turing test does not mean a machine is “intelligent” or has a “mind”.
2. Weak, weak, strong.
3. Statistics. It is a subfield of AI.
4. Machine learning, statistics, and computer science (algorithms, applications).
5. *datum*, “something given”.
6. 6 is the data, dogs is the context.
7. Discrete, continuous, nominal, ordinal, nominal.
8. Histogram. Pie chart.
9. Ordinal. Bar graph.
10. Semi-structured, unstructured, structured, semi-structured.
11. It uses labels to organize data, which eases the creation of hierarchies and groups.
12. Experimental (controlled conditions).
13. Case-control.
14. Longitudinal.
15. Cross-sectional.

16. Any three of Kaggle, Google Dataset Search, Data.Gov, Datahub.io, UCI ML Repository.

Deck 3

Jupyter, Google Colab and Python Basics

Sep 15 and Sep 17, 2026

Learning outcomes (her slide, word for word)

- Describe the concept of Jupyter Notebook
- Describe the different types of cells available on Google Collab
- Create a notebook on Google Collab with Python

Jupyter Notebook (slide 3)

- A **client-server application** that facilitates editing and running notebooks (code) through a **web browser**.
- Code runs using a **kernel**, which functions as a **computational engine** to perform the operations. There is a kernel for various languages. For Python it is **IPython**.

Google Colaboratory (slide 4)

- a.k.a **Colab** or **Google Colab**.
- A tool created by **Google in 2017** for the development of notebooks.
- A **hosted Jupyter Notebook service** that does not require any prior configuration for use.

Types of cells in Colab (slide 6)

Cell type	What to know
Code cell	Shows a pair of brackets **[ ]** on the left. When executed it shows a **numeric value that indicates the order of execution** of the cell with respect to the others, e.g. [2].
Section header cell	Double click in the cell to edit the text.
Text cell	Written in Markdown. Inserted with “+ Text” in the toolbar or from the Insert menu.
Scratch code cell	Fourth type on her list.

Run cells from the **Runtime** menu (Run all, Run before, Run the focused cell, Run selection, Run cell and below, Interrupt execution, Restart session). Delete a cell with the trash icon on the cell or from the Edit menu.

Markdown (slide 7)

- Created by **John Gruber in 2004**.
- A **lightweight markup language** used to add formatting elements, links, and images to plain text files.
- Headings: `# Heading level 1`, `## Heading level 2`, `### Heading level 3`. More # signs mean a smaller heading.

Python (slide 15)

- Created by **Guido van Rossum** in the **late 80s**.
- **High-level, interpreted, cross-platform** programming language.
- Code blocks are defined using **indentation** and do not require braces `{}`.
- Current version on her slide: **3.14.7 (August 2026)**.

Code from her slides

print() and sum of two numbers (slides 16–17)

print() receives strings in quotation marks, **single or double**.

```
print("Hello world")

num1 = 10
num2 = 20
res = num1 + num2
print(res)
```

```
Output:
Hello world
30
```

Three formats to show values with print() (slide 18)

```
print("The sum of num1 and num2 is: ", res)     # comma
print(f"The sum of num1 and num2 is: {res}")    # f-string
print("The sum of num1 and num2 is "+ str(res)) # + and str()
```

```
Output:
The sum of num1 and num2 is:  20
The sum of num1 and num2 is: 20
The sum of num1 and num2 is 20
```

The comma adds a space between items, so the first line has **two** spaces before 20. With +, you must convert the number with **str()**. Note: her slide shows 20 here even though res was 30 on slide 17. She changed the values between screenshots.

range() (slide 19)

- A **built-in function** to generate a sequence of numbers.
- By default, the numbers start from **0** up to the number indicated with steps of **1**.
- Syntax: `range(start, end, step)`. **start** and **step** are optional. The end number is **not included**.

```
for num in range(0,13,2):
    print(num)
```

```
Output:
0
2
4
6
8
10
12
```

if-elif-else (slide 21)

```
num = 2
if num <0:
    print("The number is negative")
elif num > 10:
    print("The number is greater than 10")
else:
    print("The number is greater than 0 and less than 10")
```

```
Output:
The number is greater than 0 and less than 10
```

Functions (slides 22–24)

```
def showMessage():
    print("Invoking the method showMessage")

showMessage()
```

```
Output:
Invoking the method showMessage
```

```
def calculateOperation(num1,num2,op="+"): # a function with a default parameter
    if op == "-":
        res = num1-num2
    elif op == "*":
        res = num1*num2
    elif op == "/":
        res = num1/num2
    else:
        res = num1+ num2
    print("The result is: ", res)

calculateOperation(3,2)       # without specifying the operator
calculateOperation(3,2,"*")   # specifying the * operator
```

```
Output:
The result is:  5
The result is:  6
```

op="+" is a **default parameter**. If you leave it out, the function uses "+".

Collections in Python (slide 25)

Basic collections: **List**, **Tuple**, **Dictionary**.

	List	Tuple	Dictionary
Syntax	`[elem1, elem2]` or `list()`	`(element1, element2)`	`{key1:value1, ...}`
Ordered?	**Ordered**	Yes (index access)	**No order** (her slide)
Modifiable?	**Modifiable**	**Immutable**: cannot be modified once created	**Mutable**
Repeats?	Supports repeated values	Yes	**Keys are unique**
Stores	Multiple values in a variable	Elements of different types and compound elements (lists, tuples, etc.)	Numbers, text strings, lists, and other dictionaries, as key-value pairs
Access	Indexes starting from **0**	Indexes starting from 0	By key, e.g. `.get('id')`

```
lst = [1,3,5,7]   # Create a list
len(lst)
```

```
Output:
4
```

```
tpl = (2,4,5,6)
tpl
type(tpl)   # The type function is used to verify the data type of a variable
```

```
Output:
(2, 4, 5, 6)
tuple
```

```
dic = {}   # Empty dictionary
print(dic)
dic1={'user1':'abohr', 'id':100}
dic2={'user1':'bjinx', 'id':200}
print(dic1)
dic1.get('id')   # Get values according to the key
dic3 = {'customer1':dic1,'customer2':dic2}   # dictionary using other dictionaries
dic3
```

```
Output:
{}
{'user1': 'abohr', 'id': 100}
100
{'customer1': {'user1': 'abohr', 'id': 100},
 'customer2': {'user1': 'bjinx', 'id': 200}}
```

Slicing (slides 30–31)

An operator to extract or access portions of sequences like **strings, lists, and tuples**.

Part	Optional?	Default
start: starting index of the slice	Yes	0
stop: the last index of the slice (not included)	Yes	len(sequence)
step: step value of the slice	Yes	1

Her slide writes the syntax as `[start:stop:end]` but the bullets name the third part **step**. Treat it as `[start:stop:step]`.

Her example on slide 31 shows **no output**. This is the most likely code question on the quiz:

```
lst = [10,8,6,4,2]
lst2 = lst[:]   # a copy of lst
lst2[1] = 10
print(lst2[3:])
print(lst2[:3])
print(lst2[1:-1])
print(lst2[-1:1])
print(lst2)
```

```
Output:
[4, 2]
[10, 10, 6]
[10, 6, 4]
[]
[10, 10, 6, 4, 2]
```

- `lst[:]` makes a **copy**, so changing lst2 does not change lst. lst stays [10, 8, 6, 4, 2].
- `lst2[1:-1]` drops the first and the last element.
- `lst2[-1:1]` is **empty**: it starts at the last index and moves forward with step 1, so it never reaches index 1.

⚠ **Answer her way**

• Her slide says **“Dictionaries have no order.”** Modern Python keeps insertion order, but answer **no order** on her quiz.

• Colab has exactly **four** cell types: code, section header, text, scratch code.

• The kernel for Python is **IPython**. A kernel is the **computational engine**.

• Colab: created by Google in **2017**. Markdown: John Gruber, **2004**. Python: Guido van Rossum, **late 80s**.

• Tuples are **immutable**. Lists and dictionaries are **mutable**.

• Python uses **indentation**, not braces, to define code blocks.

Practice questions

1. What is Jupyter Notebook?
2. What is a kernel, and which kernel does Python use?
3. Who created Colab and when? What does it require to use?
4. List the four types of cells in Colab.
5. What does the number inside [ ] on a code cell mean?
6. Who created Markdown and when? What is it?
7. Name three characteristics of Python from her slide.
8. What does `range(5)` generate?
9. Which parameters of range(start, end, step) are optional?
10. Three properties of a list from her slide?
11. What is the main difference between a list and a tuple?
12. Three properties of dictionaries from her slide?
13. What does `calculateOperation(3,2)` print, and why?
14. Which function verifies the data type of a variable?

Answer key

1. A client-server application that facilitates editing and running notebooks (code) through a web browser.
2. A computational engine that performs the operations. IPython.
3. Google, 2017. It requires no prior configuration (a hosted Jupyter Notebook service).
4. Code cell, section header cell, text cell, scratch code cell.
5. The order of execution of that cell with respect to the others.
6. John Gruber, 2004. A lightweight markup language to add formatting elements, links, and images to plain text files.
7. High-level, interpreted, cross-platform. Blocks defined by indentation, no braces.
8. 0, 1, 2, 3, 4
9. start and step.
10. Ordered, modifiable, supports repeated values.
11. A tuple is immutable. It cannot be modified once created.
12. Mutable, have no order, keys are unique.
13. The result is: 5. op defaults to "+".
14. type()

Deck 4

Machine Learning Basics

Sep 22 and Sep 24, 2026

Learning outcomes (her slide, word for word)

- Distinguish the concept of algorithm and model in the context of machine learning
- Describe the different datasets employed to build a machine learning model
- List the machine learning techniques
- Define the concept of supervised learning
- Define the concept of unsupervised learning

Algorithm vs model (slides 3–5)

Algorithm	Model
Procedure that is performed on a set of data to **create a model**.	**Output of an algorithm** after it has been executed on a dataset.
Tasks they perform: **pattern recognition, learning from data**.	**Mathematical representation** of what has been learned using an algorithm.
Examples: **linear regression, logistic regression, neural networks**.	The generation of a model **depends on the data** that an algorithm uses to learn.

Slide 5: models are created from a set of data that allows patterns to be found and new predictions to be made. The models **must be evaluated with a different set of data** to assess their performance in new cases.

One-line version: **algorithm + data → model**.

Machine learning techniques (slide 6)

Column 1	Column 2
1. Regression	5. Anomaly detection
2. Classification	6. Sequence mining
3. Clustering	7. Dimension reduction
4. Associations	8. Recommendation systems

Memory hook: **R C C A** / **A S D R**. “Really Cool Cats Always / Avoid Silly Dogs Running.”

Splitting the dataset (slides 7–10)

All data splits into three sets: **Training** (the largest), **Validation**, **Test**.

Set	Question it answers (slide 7)	Her description
Training	Models learn the task	Data used to train a model. Data previously analyzed and “cleaned.” The quality of the model will depend on the quality of the data.

Set	Question it answers (slide 7)	Her description
Validation	Which model is the best?	Data with similar characteristics to the training dataset (distribution, patterns). Used to verify the **accuracy and error** of the model before testing it on the test dataset. Facilitates the tuning of the model's **hyperparameters**.
Test	How good is this model truly?	Data used to test the predictions of the generated model with **new cases**. Lets us know the accuracy of the model once it has been adjusted with the validation dataset.

Steps to create a machine learning model (slide 11)

1. Manage Data	2. Train Models	3. Evaluate Models	4. Deploy Model
Collect Explore Cleanse Prepare Split	Pick learning task Engineer features Select algorithms	Score models Assess performance Compare alternatives	Apply model to fresh data Monitor outcomes Improve model

A loop arrow goes from Deploy back to Manage Data: **“Update with new data.”**

Steps to build and deploy a machine learning model (slide 12)

Nine steps in order (Kaestner and Kang, Carnegie Mellon):

#	Step	#	Step	#	Step
1	Model Requirements	4	Data Cleaning	7	Model Evaluation
2	Data Collection	5	Feature Engineering	8	Model Deployment
3	Data Labeling	6	Model Training	9	Model Monitoring

Types of machine learning (slides 13–19)

Supervised learning, **Unsupervised learning**, **Reinforcement learning**.

	Supervised	Unsupervised	Reinforcement
Data	**Labeled** training data set. Training data contains both **input and output (label)**.	**Unlabeled** data.	**Does not require data for training**.
How it learns	Finding a **function** that takes the input data and assigns an output label.	Groups data on its own. Can result in **less accurate predictions**.	Through the **experience** of an **agent** in its **environment**. The agent observes, responds with an **action**, and gets a **reward** depending on the action.
Used for	**Classification**: predicting class labels or categories. **Regression**: predicting continuous values.	**Group data into categories** and **detect anomalies**.	Objective: associate the observation with the action (**policy**) that leads to a reward.
Her diagram	Labeled pictures (bird, fish) → model training → model. Input a new picture → prediction “bird”.	Unlabeled pictures → model training → model → outcome: groups.	

⚠ **Answer her way**

- **Reinforcement learning “does not require data for training.”** Write that if asked.
- **Unsupervised learning “can result in less accurate predictions.”**
- Unsupervised is used to **group data into categories and detect anomalies**.
- Supervised has two types: **classification** (categories) and **regression** (continuous values).
- Linear regression, logistic regression and neural networks are **algorithms**, not models.
- Hyperparameter tuning belongs to the **validation** set, not the test set.
- She lists **eight** ML techniques. Know all eight.

Practice questions

1. Distinguish an algorithm from a model.
2. Is a neural network an algorithm or a model on her slides?
3. What does the generation of a model depend on?
4. Why must a model be evaluated on a different dataset?
5. List the eight ML techniques.
6. Which dataset is used to tune hyperparameters?
7. Match each set to its question: “Which model is the best?”, “How good is this model truly?”, “Models learn the task.”
8. Name the four stages of slide 11 in order.
9. Under which stage does “Split” appear? “Engineer features”?
10. Which step comes right after Data Labeling in the nine-step pipeline?
11. Define supervised learning.
12. Predicting house prices is which type of supervised learning? Predicting spam or not spam?
13. Define unsupervised learning and give its two uses.
14. In reinforcement learning, what is a policy?
15. Which type of learning does not require data for training?

Answer key

1. An algorithm is a procedure performed on data to create a model. A model is the output of an algorithm after it runs on a dataset, a mathematical representation of what was learned.
2. An algorithm.
3. The data that the algorithm uses to learn.
4. To assess its performance in new cases.
5. Regression, classification, clustering, associations, anomaly detection, sequence mining, dimension reduction, recommendation systems.
6. The validation dataset.
7. Validation, test, training.
8. Manage data, train models, evaluate models, deploy model.
9. Manage data. Train models.
10. Data Cleaning.
11. Learning with a labeled training dataset that contains input and output (label) data, finding a function that takes the input and assigns an output label.
12. Regression. Classification.
13. Training the model with unlabeled data, which can result in less accurate predictions. Used to group data into categories and detect anomalies.
14. The association between the observation and the action that leads to a reward.
15. Reinforcement learning.

Python Tracing Drill

Her style, small changes. Write the output on paper before you check the key. Every answer below was run in Python 3.

1	`for i in range(1,10,3):`
	`  print(i)`
		`1`
		`4`
		`7`

2	`print(list(range(2,8)))`	`[2, 3, 4, 5, 6, 7]`
3	`print(list(range(10,0,-3)))`	`[10, 7, 4, 1]`
4	`print(list(range(5,2)))`	`[]` `(start is already past end with step 1)`

5	`total = 0`
	`for n in range(1,5):`
	`  total = total + n`
	`print(total)`
		`10`
		`(1+2+3+4, 5 is not included)`

6	`num = 0`
	`if num <0:`
	`  print("negative")`
	`elif num > 10:`
	`  print("greater than 10")`
	`else:`
	`  print("between")`
		`between`

7	`num = 10` `(same if-elif-else as her slide 21)`	`The number is greater than 0 and less than 10` `(10 > 10 is False, so it falls into else)`
8	`calculateOperation(3,2,"/")`	`The result is:  1.5`
9	`calculateOperation(6,3,"/")`	`The result is:  2.0` `(/ always gives a float)`
10	`calculateOperation(3,2,"%")`	`The result is:  5` `("%" is not handled, so else adds)`

11	`lst = [1,3,5,7]`
	`print(lst[-1])`
	`print(lst[::2])`
	`print(lst[1:10])`
		`7`
		`[1, 5]`
		`[3, 5, 7]`
		`(stop past the end is fine)`

12	`s = "Intelligent"`
	`print(s[0:5])`
	`print(s[-3:])`
	`print(s[::-1])`
		`Intel`
		`ent`
		`tnegilletnI`

13	`tpl = (2,4,5,6)`
	`print(tpl[1:3])`
	`print(len(tpl))`
		`(4, 5)`
		`4`

14	`tpl = (1,2)`
	`tpl[0] = 5`
		`TypeError: 'tuple' object does not`
		`support item assignment`
		`(tuples are immutable)`

15	`dic1 = {'user1':'abohr', 'id':100}`
	`print(dic1.get('age'))`
	`print(dic1['user1'])`
		`None`
		`abohr`

16	`dic1['id'] = 150` `print(dic1)`	`{'user1': 'abohr', 'id': 150}` `(dictionaries are mutable)`

17	`d = {'a':1, 'a':2}` `print(d)`	`{'a': 2}` `(keys are unique, the last value wins)`

18	`x = [1,2,3]`
	`y = x`
	`y[0] = 99`
	`print(x)`
		`[99, 2, 3]`
		`(y = x is NOT a copy)`

19	`x = [1,2,3]`
	`y = x[:]`
	`y[0] = 99`
	`print(x)`
		`[1, 2, 3]`
		`(x[:] is a copy, like her slide 31)`

20	`res = 20`
	`print("Total:", res)`
	`print("Total:" + str(res))`
		`Total: 20`
		`Total:20`
		`(+ adds no space)`

Tip: cover the right column with a sheet of paper while you work.

Memorize Page

Every list and pairing she can ask you to recite.

Topic	Recite this
AI definition	McCarthy (2007): the science and engineering of making intelligent machines, especially intelligent computer programs
Key years	1950 Turing test · 1956 AI term (Dartmouth) · 1997 Deep Blue · 2012 AlexNet · 2016 AlphaGo · 2017 Transformers · 2018 AlphaFold · 2022 ChatGPT
3 eras	1950–1980 AI · 1980–2010 ML · 2010–present DL
Winter 1 (1970s)	Unmet expectations · limited computing power · funding cuts
Winter 2 (late 80s–90s)	Decline of expert systems · high costs, low scalability · reduced investment, skepticism
5 ethical dilemmas	Biases (gender, age, race) · decision making · privacy · property/IP · environmental impact
7 use cases	Crime prediction · traffic accidents · forest fires · proctoring · text autocomplete · CV scanning · dynamic pricing
Chinese room	John Searle · passing the Turing test ≠ having a mind
Weak vs strong AI	Weak: specific tasks (chatbots, spam filters, Google Maps, Siri, Alexa) · Strong: like a human, learns new skills on its own, adapts
Information	Data + Context + Meaning · context = unit of measurement
Data types	Quantitative: discrete (integer), continuous (float) · Qualitative: nominal (no order), ordinal (order)
Graphs	Nominal → pie · Ordinal → bar · Discrete → bar · Continuous → histogram
Structure	Structured (rows/columns) · Unstructured (videos, comments) · Semi-structured (labels: email, JSON, XML)
Collection	Observational (natural setting) · Experimental (controlled, A/B test)
4 observational studies	Cross-sectional · Case-control · Cohort · Longitudinal
5 dataset sources	Kaggle · Google Dataset Search · Data.Gov · Datahub.io · UCI ML Repository
Creators	Colab: Google 2017 · Markdown: John Gruber 2004 · Python: Guido van Rossum, late 80s
4 Colab cells	Code · Section header · Text · Scratch code
Collections	List: ordered, modifiable, repeats · Tuple: immutable · Dict: mutable, no order, unique keys
range / slicing	range(start, end, step), end excluded · [start:stop:step], defaults 0, len, 1
Algorithm vs model	Algorithm: procedure on data to create a model · Model: output of the algorithm
8 ML techniques	Regression · Classification · Clustering · Associations · Anomaly detection · Sequence mining · Dimension reduction · Recommendation systems
3 datasets	Training (learn) · Validation (which model is best, hyperparameters) · Test (how good truly)
4 stages	Manage data → Train models → Evaluate models → Deploy model → update with new data
9 steps	Requirements · Collection · Labeling · Cleaning · Feature eng. · Training · Evaluation · Deployment · Monitoring
3 ML types	Supervised: labeled (classification, regression) · Unsupervised: unlabeled, less accurate, groups and anomalies · Reinforcement: no data, agent, action, reward, policy