AI Bootcamp

AI Applications with Gradio

Module 21 Day 3

Class Objectives

By the end of class, you will be able to:

1 Become familiar with the Gradio and its different components.

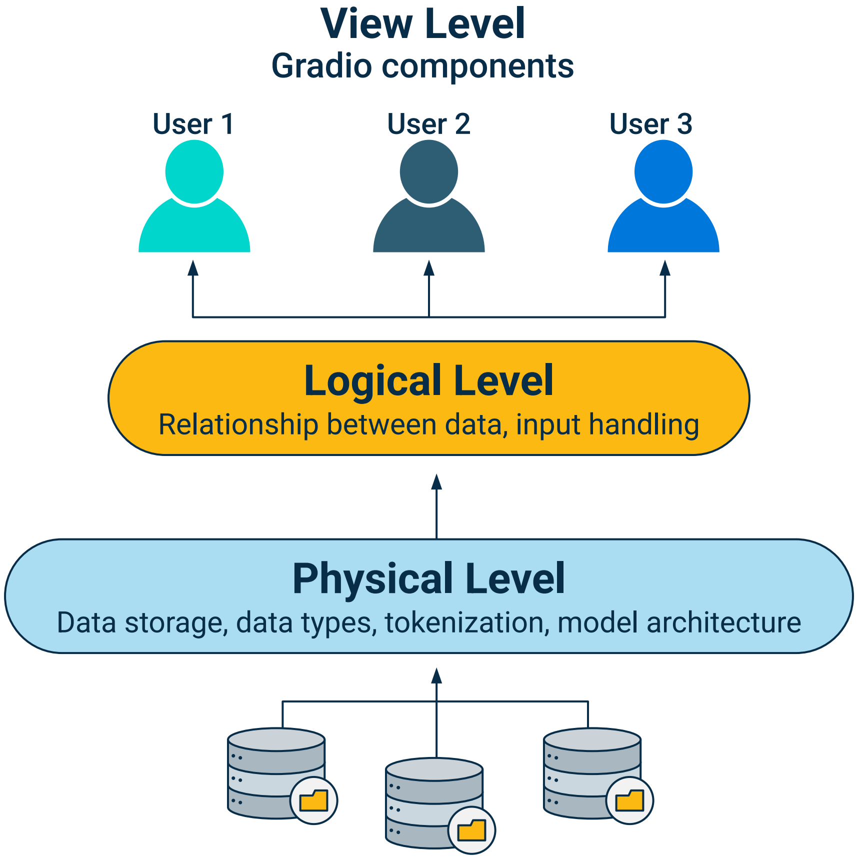
2 Create Python applications using Gradio.

3 Create NLP-applications using Gradio and Hugging Face transformers.

Instructor **Demonstration**

Introduction to Gradio Interfaces

Levels of abstraction



View Level

Gradio components

User 1

User 2

User 3

Logical Level

Relationship between data, input handling

Physical Level

Data storage, data types, tokenization, model architecture

Activity:

Pizza Order with Gradio

In this activity, you will write a function that creates a pizza order that calculates the total cost of a pizza order based on the size and includes up to three toppings. Then, you will use Gradio to launch the application.

Suggested Time:
15 Minutes

Time’s up!

Let’s review

Questions?

Instructor **Demonstration**

Text Summarization with Gradio

Activity:

Text Summarization with Gradio

In this activity, you will use Gradio to create an application that creates the most likely summary and more diverse summary of text of your choosing.

Suggested Time:

10 Minutes

Time’s up!

Let’s review

Questions?

Instructor **Demonstration**

Other Gradio Components

Activity:

Question and Answering Text Box with Gradio

In this activity, you will refactor the code from the Hugging Face Question and Answering solution and create two textbox components.

Suggested Time:
15 Minutes

Time’s up!

Let’s review

Questions?

Break

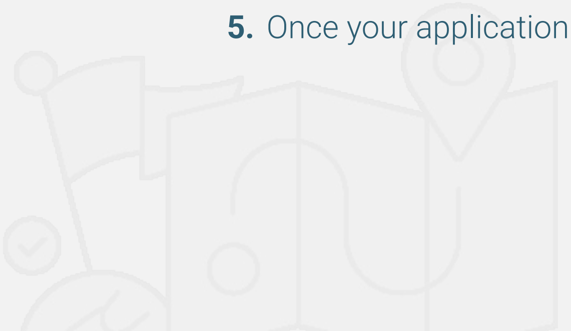
15 mins

Activity:

Build an Application with Gradio

In your groups:

1. Identify one task you want your application to perform by completing the following sentence: Our application will allow users to ______________________________.
2. Decide what kind of program you need to code to execute this task. You can code an AI model or a Python application.
3. After importing any libraries or data you may need, define a function to execute the main functionality of your application.
4. Add a Gradio Interface that allows users to provide their input and view output. Add labels and components that make the application intuitive for users to interact with.
5. Once your application is functioning, generate a public URL to the application.



Suggested Time:
30 Minutes

Time’s up!

Let’s review

Build an Application with Gradio

Peer feedback questions

1. Which features/components made the application intuitive to use?

2. Which existing features/components could be improved to create a more intuitive interface?

3. Are there any extra features/components you think a user might find valuable when using this application?

Questions?

Question 1:

Gradio Interfaces can only be used to create applications for NLP models?

1. True
2. False

Question 1: Answers

Gradio Interfaces can only be used to create applications for NLP models?

1 True — This is incorrect. Gradio can be used to create web-based demos for any machine learning models.

2 False — **Correct**. Gradio can be used to create web-based demos for any machine learning models.

Question 2:

Consider the following code snippet:

```
slider = gr.Slider(50, 200, value= 100, label="Number",
info="Choose between a 50 and 200 word summary.")
```

Which of the following values will be the default value for the slider?

1 50

2 100

3 200

4 None of the above

Question 2: Answers

Consider the following code snippet:

```
slider = gr.Slider(50, 200, value= 100, label="Number",
info="Choose between a 50 and 200 word summary.")
```

Which of the following values will be the default value for the slider?

1	50	Incorrect
2	100	**Correct**
3	200	Incorrect
4	None of the above	Incorrect

Question 3:

Consider the following code snippet:

```
slider = gr.Slider(50, 200, value=50, label="Number", info="Choose summary length.")
app = gr.Interface(fn=summarize, inputs=["text", slider,], outputs="text")
```

Which type of components will be part of the Gradio Interface? Select all that apply.

1 Text component for string input

2 Number component for numerical input

3 Checkbox component for boolean input

4 Slider component to select a number from a range

Question 3: Answers

Consider the following code snippet:

```
slider = gr.Slider(50, 200, value=50, label="Number", info="Choose summary length.")
app = gr.Interface(fn=summarize, inputs=["text", slider,], outputs="text")
```

Which type of components will be part of the Gradio Interface? Select all that apply.

1	Text component for string input	**Correct**
2	Number component for numerical input	Incorrect
3	Checkbox component for boolean input	Incorrect
4	Slider component to select a number from a range	**Correct**

Let’s **recap**

Recap

After today's lesson you are able to:

1. Be familiar with Gradio and its different components.
2. Create Python applications using Gradio.
3. Create NLP applications using Gradio and Hugging Face transformers.

Challenge

For this week's challenge you will refactor code from an SMS text classification solution into a function that constructs a linear Support Vector Classification (SVC) model. Once the model is created and trained, you will create a Gradio app to host the application, enabling users to test text messages.

Questions?

The End