

Gradio setup

From design to code.

MLoth



What is Gradio?

Gradio is an open-source Python package that allows you to quickly build a demo or web application for your machine learning model, API, or any arbitrary Python function.

You can then share a link to your demo or web application in just a few seconds using Gradio's built-in sharing features.

No JavaScript, CSS, or web hosting experience needed!

— From [the Gradio website](#)



Gradio Examples

Month
Select Months

May

Day
Select Day

9

Weather element
Choose weather element

temperature(°C) wind(m/s) humidity(%) air_pressure(hPa)

Location
Choose location

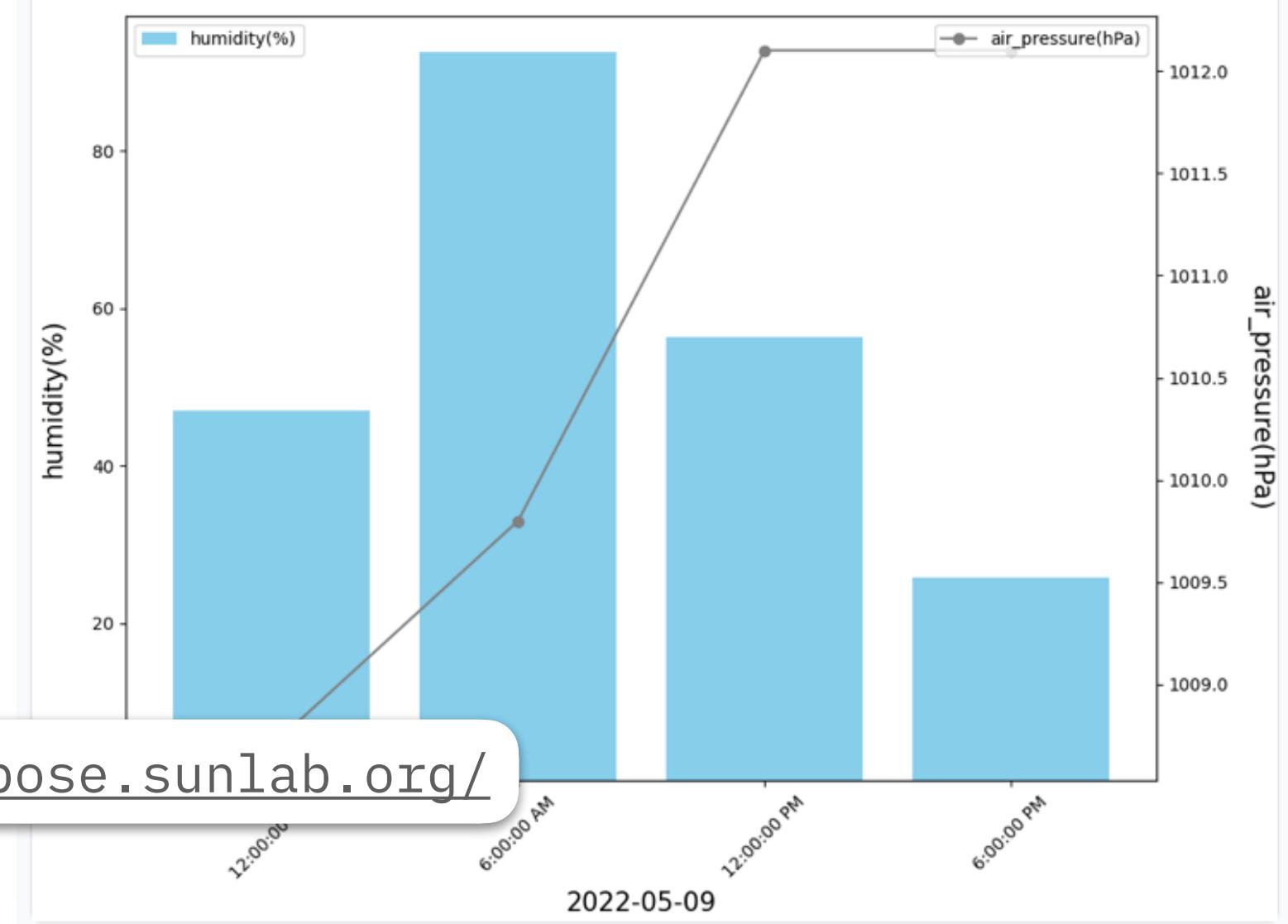
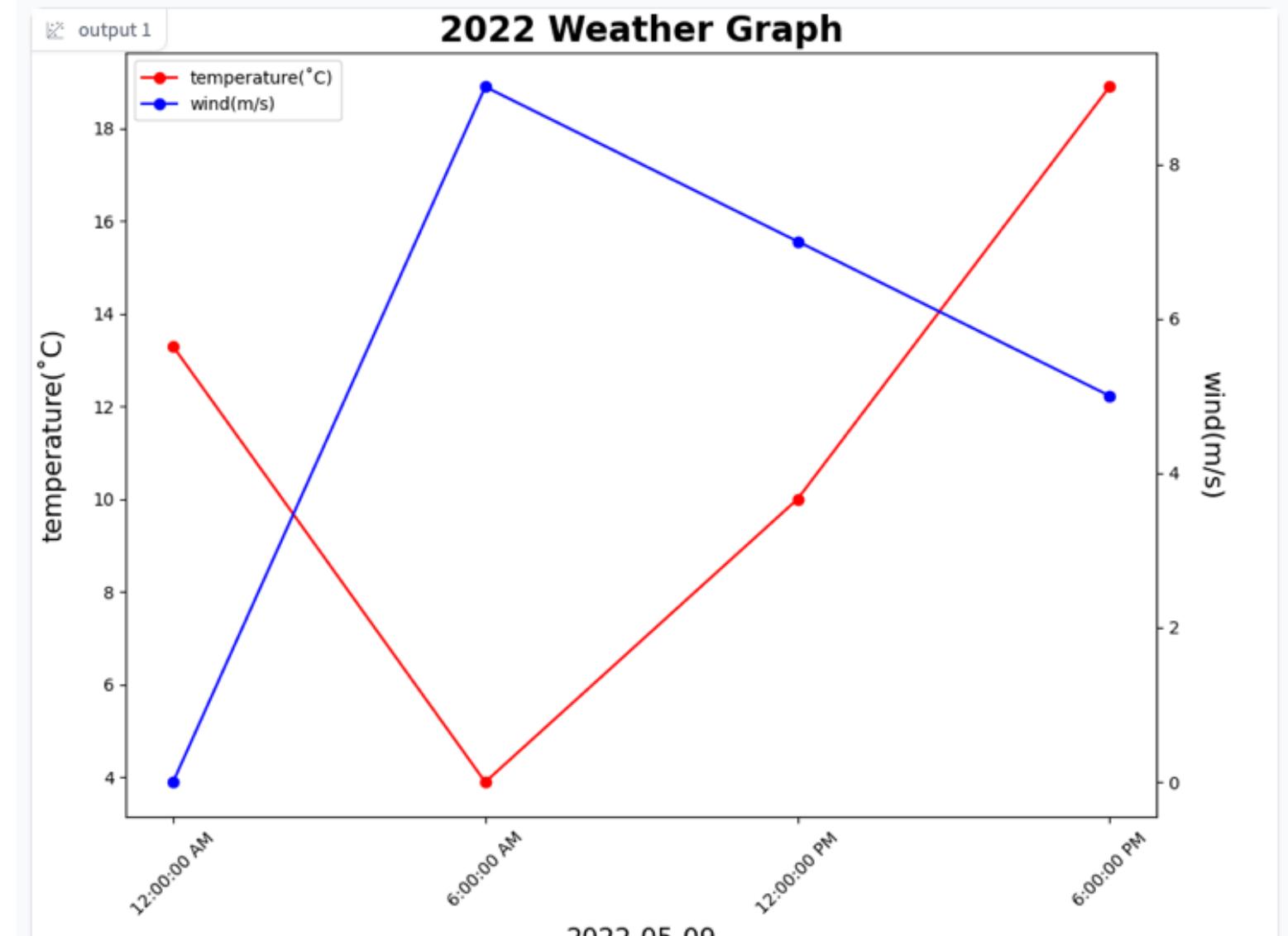
Washington Seoul

precipitation?

Clear Submit

output 0

location	date	time	temperature(°C)	wind(m/s)	humidity(%)
Washington	2022-05-09	12:00:00 AM	13.3	0	46.9
Washington	2022-05-09	6:00:00 AM	3.9	9	92.5
Washington	2022-05-09	12:00:00 PM	10	7	56.3
Washington	2022-05-09	6:00:00 PM	18.9	5	25.8

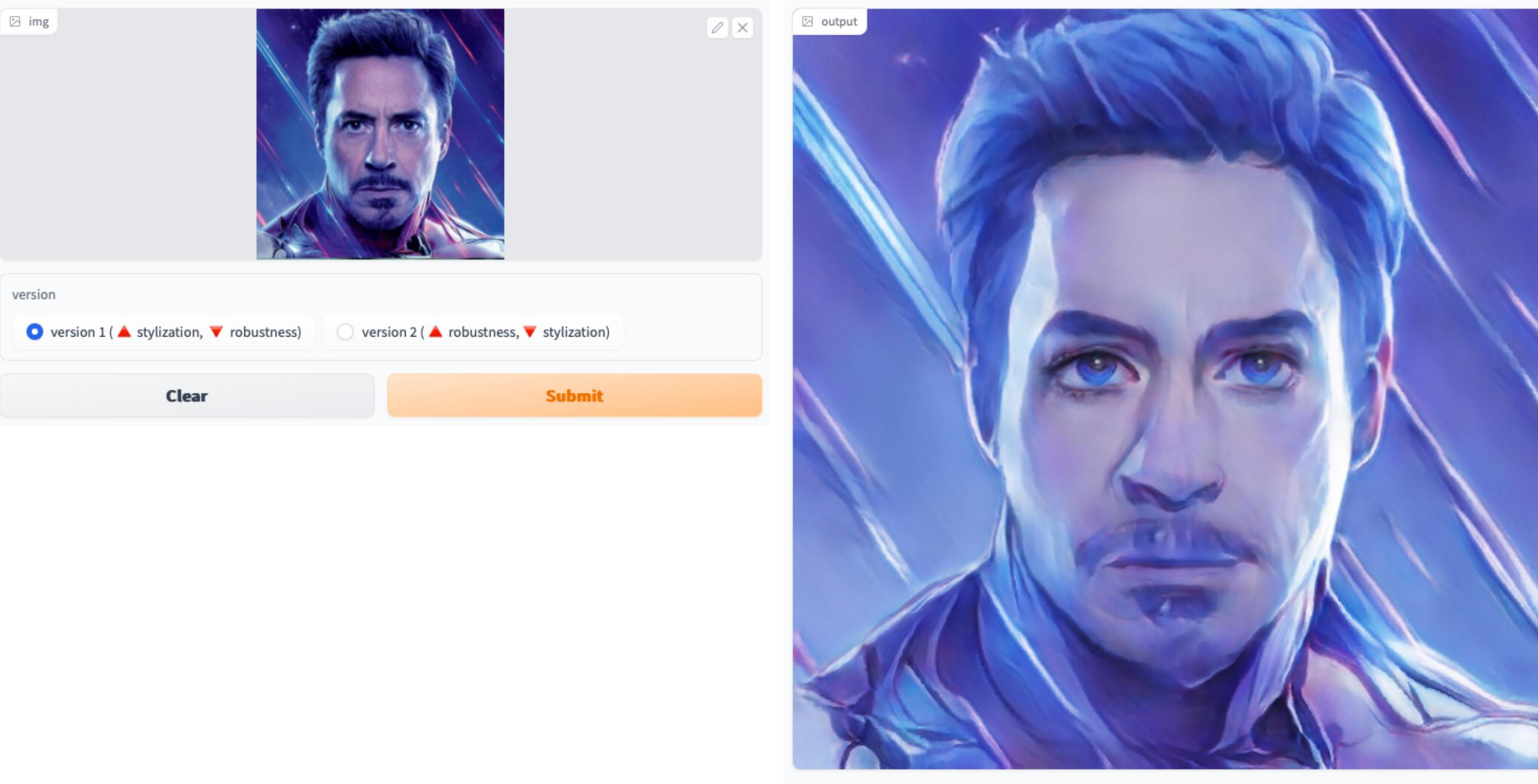


<http://deeppurpose.sunlab.org/>

AnimeGANv2

Gradio Demo for AnimeGanv2 Face Portrait. To use it, simply upload your image, or click one of the examples to load them. Read more at the links below. Please use a cropped portrait picture for best results similar to the examples below.

Gradio Examples

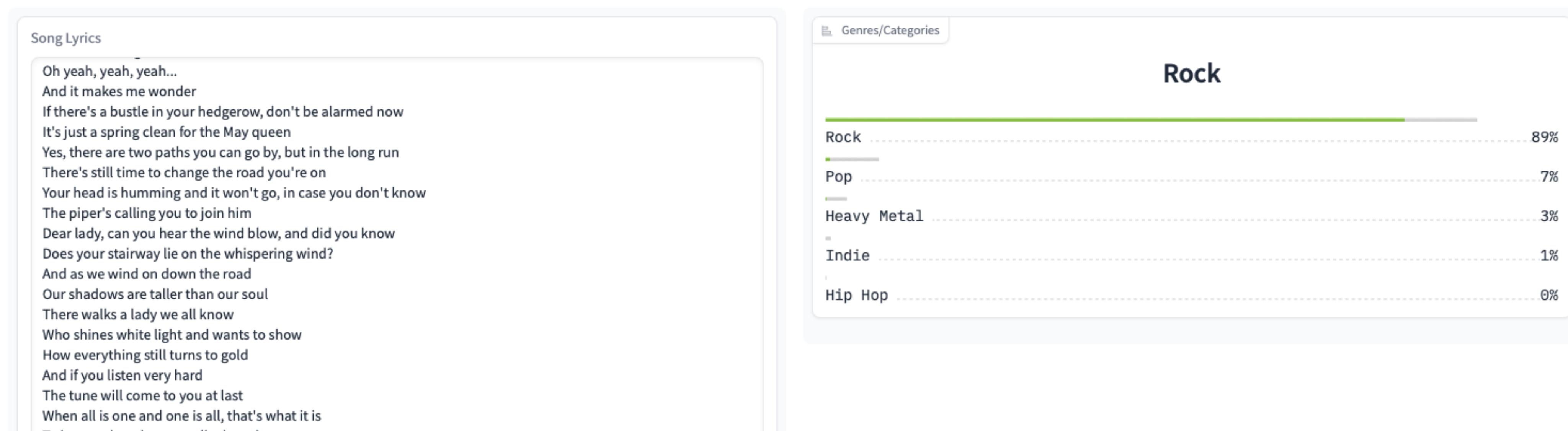


Examples	
img	version
	version 2 (▲ robustness, ▼ stylization)
	version 1 (▲ stylization, ▼ robustness)
	version 1 (▲ stylization, ▼ robustness)
	version 2 (▲ robustness, ▼ stylization)
	version 1 (▲ stylization, ▼ robustness)
	version 2 (▲ robustness, ▼ stylization)

<https://huggingface.co/spaces/akhaliq/AnimeGANv2>

Gradio Examples

Song Genre Predictor



≡ Examples

There's a lady who's sure all that glitters is gold And she's buying a stairway to heaven When she gets there she knows, if the stores are all closed With a word, she can get what she came for Ooh, ooh, and she's buying a stairway to heaven There's a sign on the wall, but she wants to be sure 'Cause you know sometimes words have two meanings In a tree by the brook, there's a songbird who sings Sometimes all of our thoughts are misgiven You know There's a feeling I get when I look to the west And my spirit is crying for leaving In my thoughts, I have seen rings of smoke through the trees And the voices of those who stand looking That's you And it's whispered that soon if we all call the tune Then the piper will lead us to reason And a new day will dawn for those who stand long And the forests will echo with laughter Remember laughter? Oh yeah, yeah, yeah... And it makes me wonder If there's a bustle in your hedgerow, don't be alarmed now It's just a spring clean for the May queen Yes, there are two paths you can go by, but in the long run There's still time to change the road you're on Your head is humming and it won't go, in case you don't know The piper's calling you to join him Dear lady, can you hear the wind blow, and did you know Does your stairway lie on the whispering wind? And as we wind on down the road Our shadows are taller than our soul There walks a lady we all know Who shines white light and wants to show How everything still turns to gold And if you listen very hard The tune will come to you at last When all is one and one is all, that's what it is To be a rock and not to roll, oh yeah And she's buying a stairway to heaven

I'm tired of being what you want me to be Feeling so faithless, lost under the surface Don't know what you're expecting of me Put under the pressure of walking in your shoes Every step that I take is another mistake to you (Caught in the undertow, just caught in the undertow) I've become so numb I can't feel you there Become so tired So much more aware I'm becoming this All I want to do Is be more like me And be less like you Can't you see that you're smothering me Holding too tightly, afraid to lose control? 'Cause everything that you thought I would be Has fallen apart right in front of you Every step that I take is another mistake to you (Caught in the undertow, just caught in the undertow) And every second I waste is more than I can take I've become so numb I can't feel you there Become so tired So much more aware I'm becoming this All I want to do Is be more like me And be less like you And I know I may end up failing too But I know You were just like me with someone disappointed in you I've become so numb I can't feel you there Become so tired So much more aware I'm becoming this All I want to do Is be more like me And be less like you I've become so numb I can't feel you there (I'm tired of being what you want me to be) there (I'm tired of being what you want me to be)

<https://huggingface.co/spaces/bharat-raghunathan/song-lyrics-classifier>

End of passion play crumpling away I'm your source of self-destruction veins that pump with fear sucking darkness clear Leading on your death's construction Taste me, you will see More is all you need Dedicated to How I'm killing you Come crawling faster (faster) Obey your master (master) Your life burns faster (faster) Obey your Master Master Master of puppets, I'm pulling your strings Twisting your mind and smashing your dreams Blinded by me, you can't see a thing Just call my name 'cause I'll hear you scream Master Master, just call my name 'cause I'll hear you scream Master Master Needlwork the way Never you betray Life of death becoming clearer Pain monopoly Ritual misery Chop your breakfast on a mirror Taste me you will see More

Gradio

Core concepts

- **Interface:** the main app itself. This is where components 'live'.
- **Input:** some data the users inputs, types, whatever.
 - Think of text, the camera, audio, a sensor, etc.
 - Can be one or multiple.
- **Output:** data your Python code generates
 - The results of an AI-model, external API data, etc.
 - Can also be one or multiple.

Input

In Gradio

- An input refers to the type of data that a user provides to a machine learning model, function, or application.
- It defines **the way users interact with the system.**
- This will usually be something from the user (text, voice, image, files, etc.).

The image shows a Gradio user interface for a "activity" task. At the top, there is a checkbox labeled "Is it before noon". Below this, the word "activity" is followed by a dropdown menu icon. Under the activity section, there are three checkboxes: "juice", "beverage", and "snack". Further down, there is another checkbox labeled "nap". At the bottom of the form are two buttons: "Clear" (gray) and "Submit" (orange).

Output

In Gradio

- The type of data that **a function** returns and is displayed in the interface.
- It determines **how the processed result is shown** to the user.
- But keep in mind:
 - Eg.: if the function returns **True** this **can mean a lot of things** depending on the context. Design and show the most fitting output.
 - $1 \rightarrow \text{True} \rightarrow \checkmark \rightarrow \text{👍} \rightarrow \text{🎉} // \text{ You are in the 1st percentile!}$
 - $1 \rightarrow \text{True} \rightarrow \text{✗} \rightarrow \text{👎} \rightarrow \text{🦠} // \text{ You are ill...}$

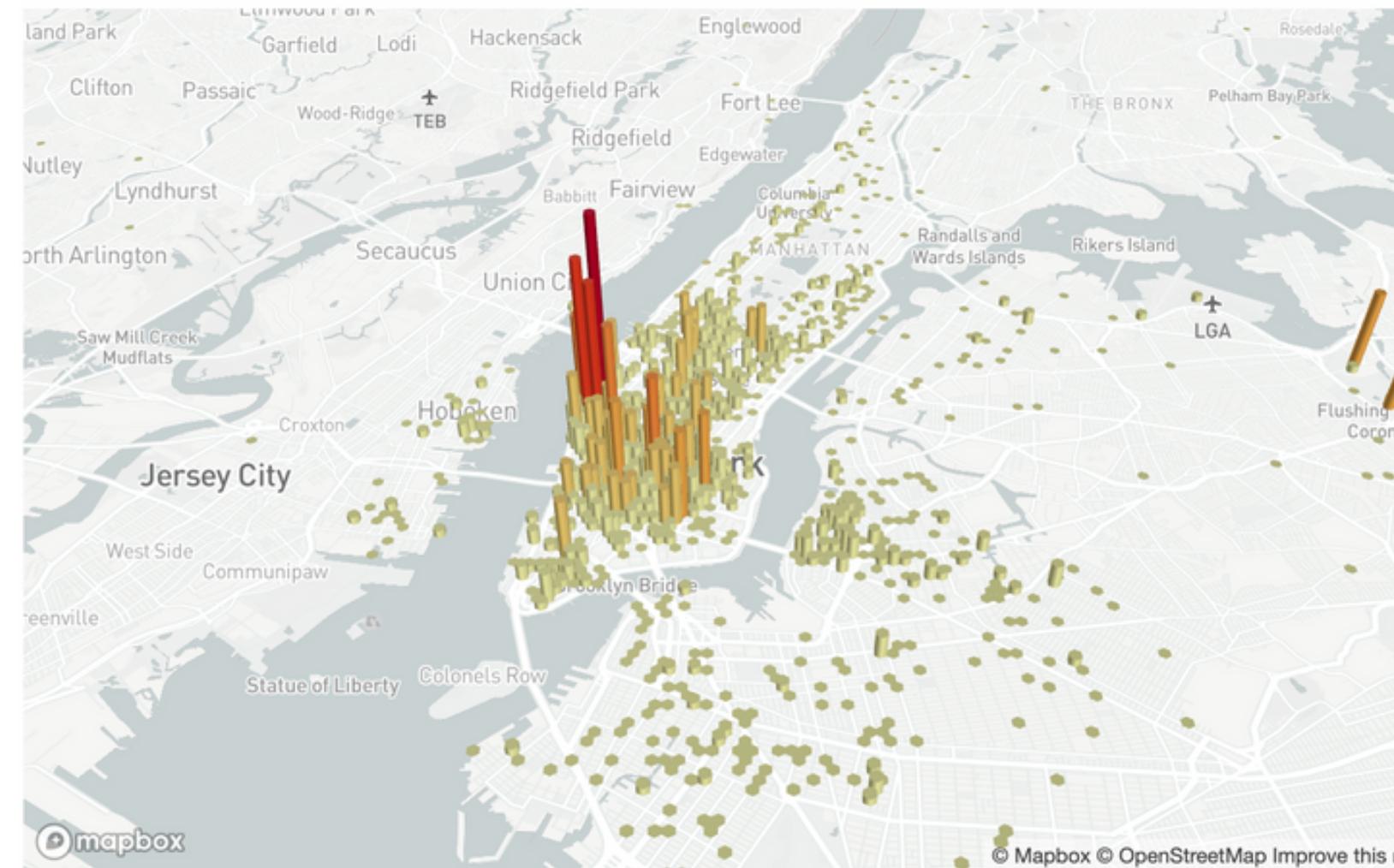
Output Can be much more...

NYC Uber Ridesharing Data

Select hour of pickup

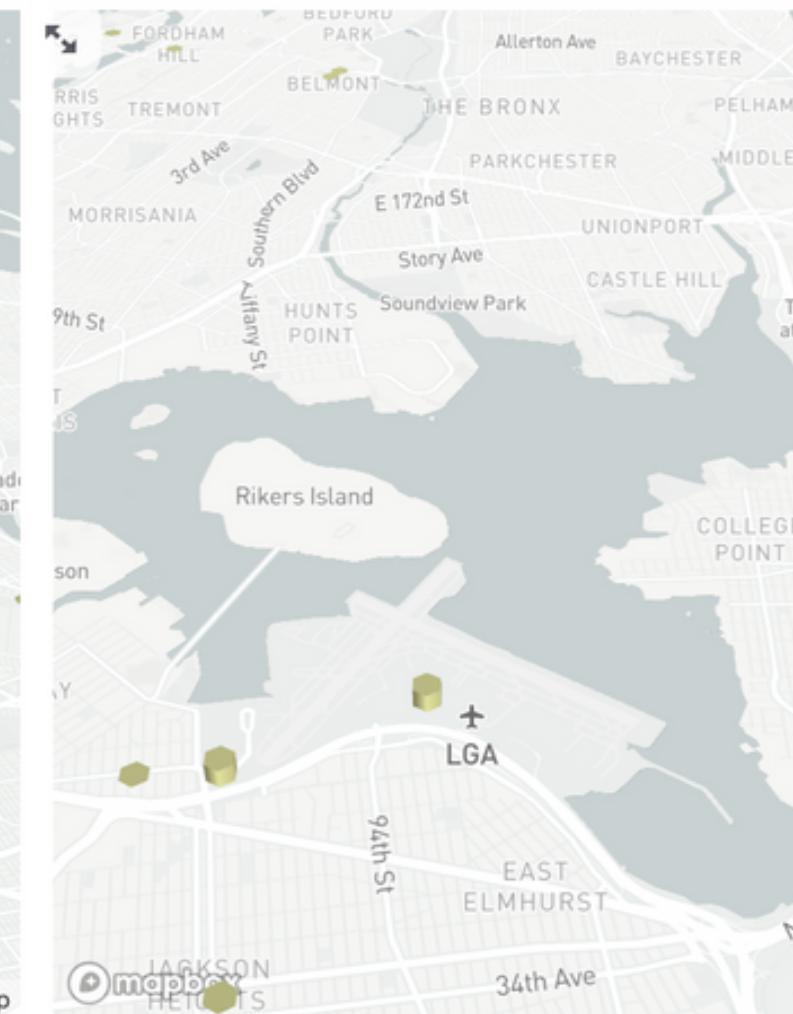


All New York City from 0:00 and 1:00

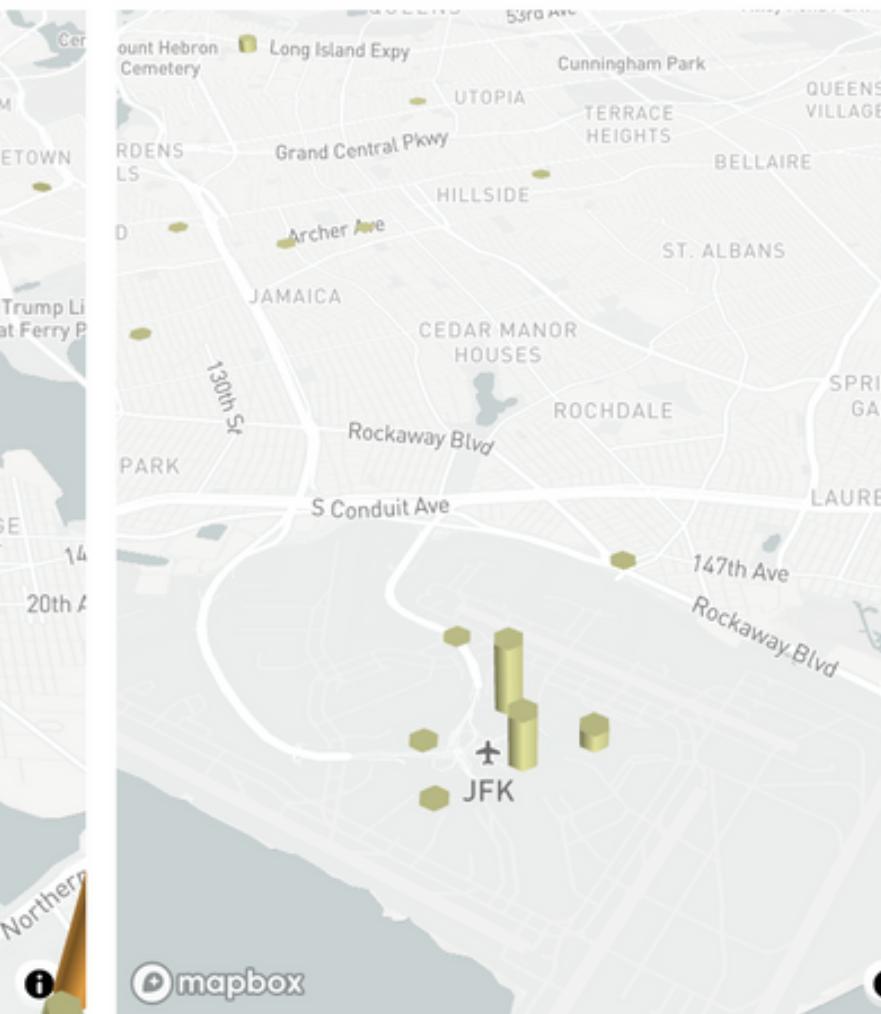


Examining how Uber pickups vary over time in New York City's and at its major regional airports. By sliding the slider on the left you can view different slices of time and explore different transportation trends.

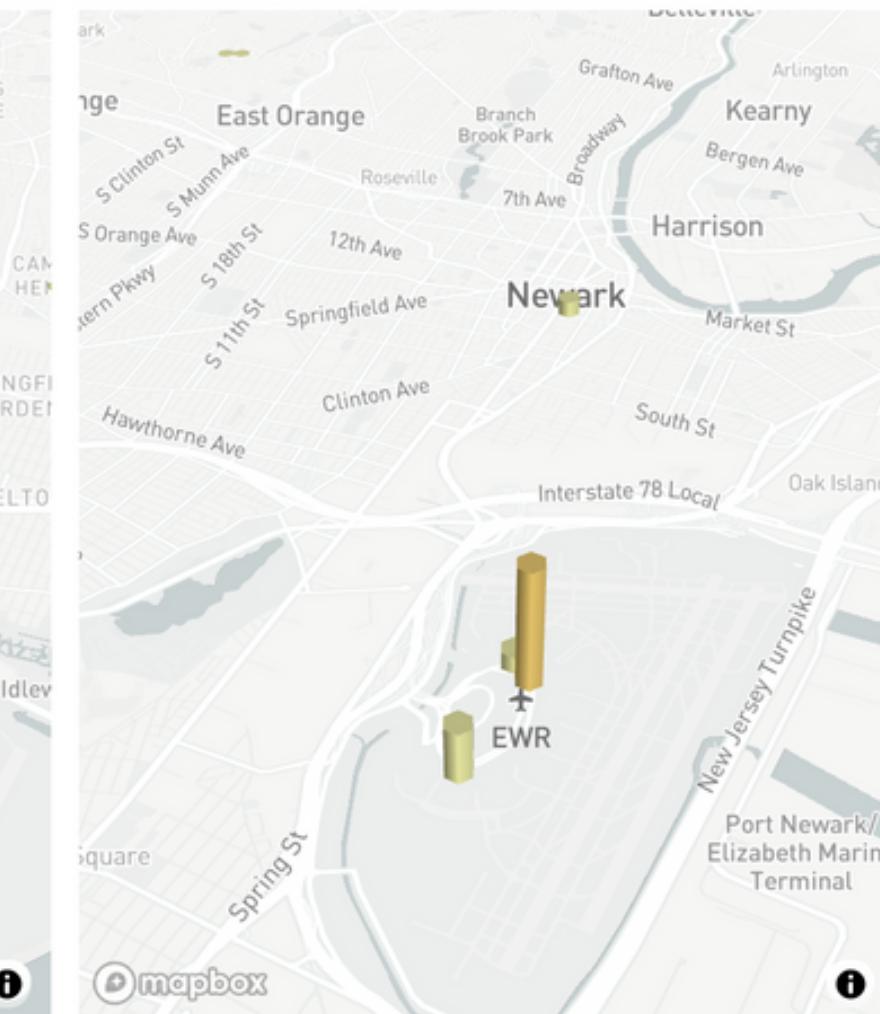
La Guardia Airport



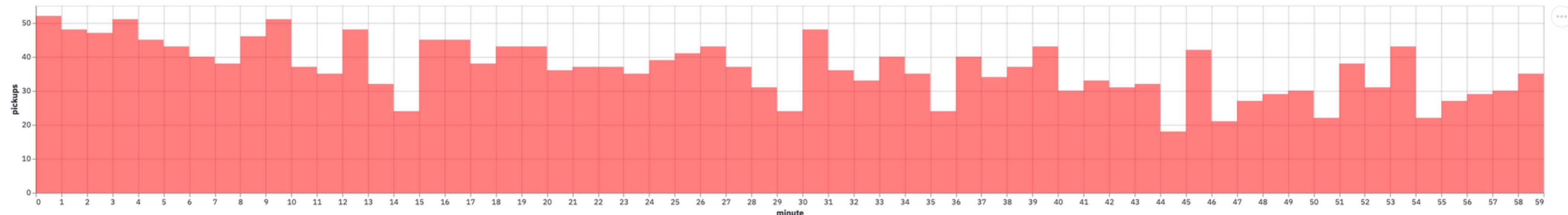
JFK Airport



Newark Airport



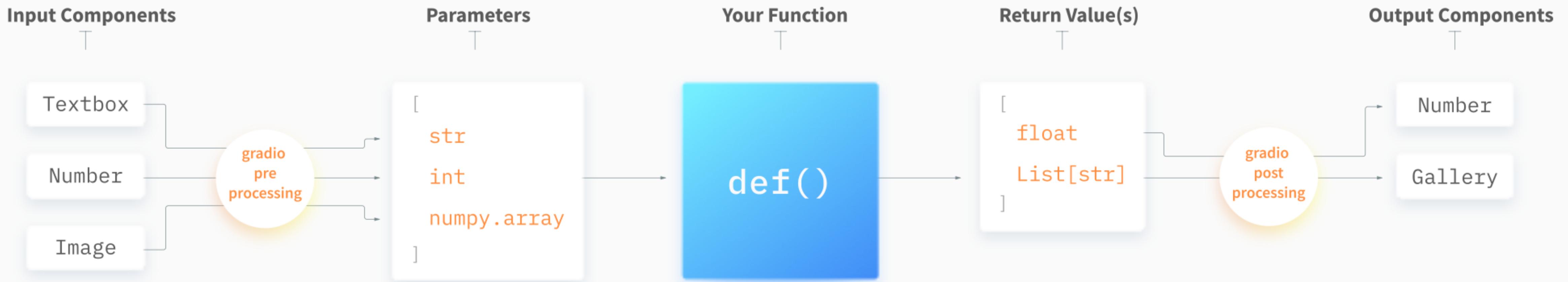
Breakdown of rides per minute between 0:00 and 1:00



The interface class

In Gradio

- From the interface function, we define how the input(s) are being processed to become the desired output(s).



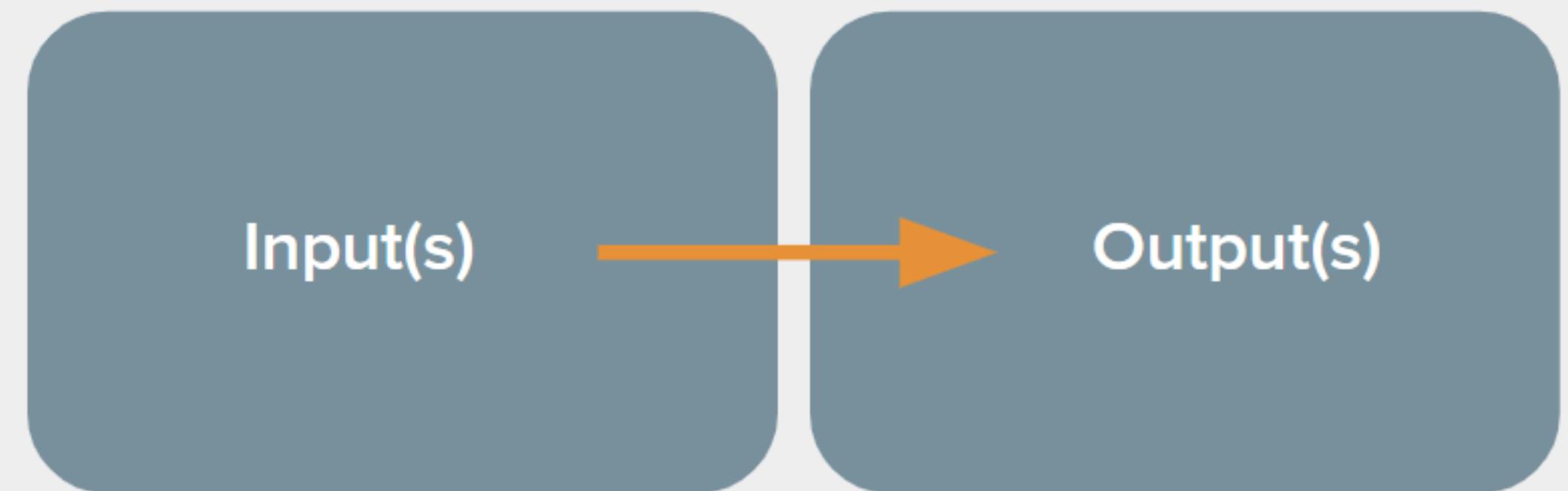
The interface class

In Gradio

The three arguments of this function:

1. **fn**: the function to wrap a user interface (UI) around
2. **inputs**: which Gradio component(s) to use for the input. The number of components should match the number of arguments in your function.
3. **outputs**: which Gradio component(s) to use for the output. The number of components should match the number of return values from your function.

`gr.Interface(fn, inputs, outputs)`



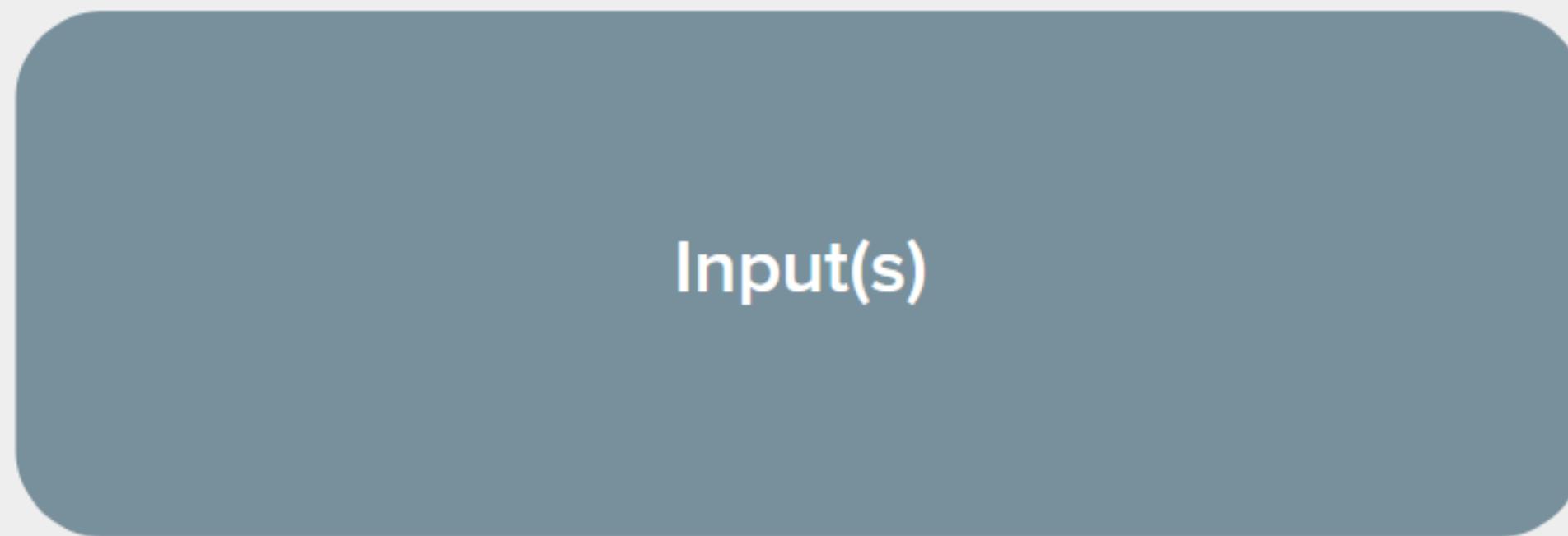
Standard Interface

`gr.Interface(fn, None, outputs)`



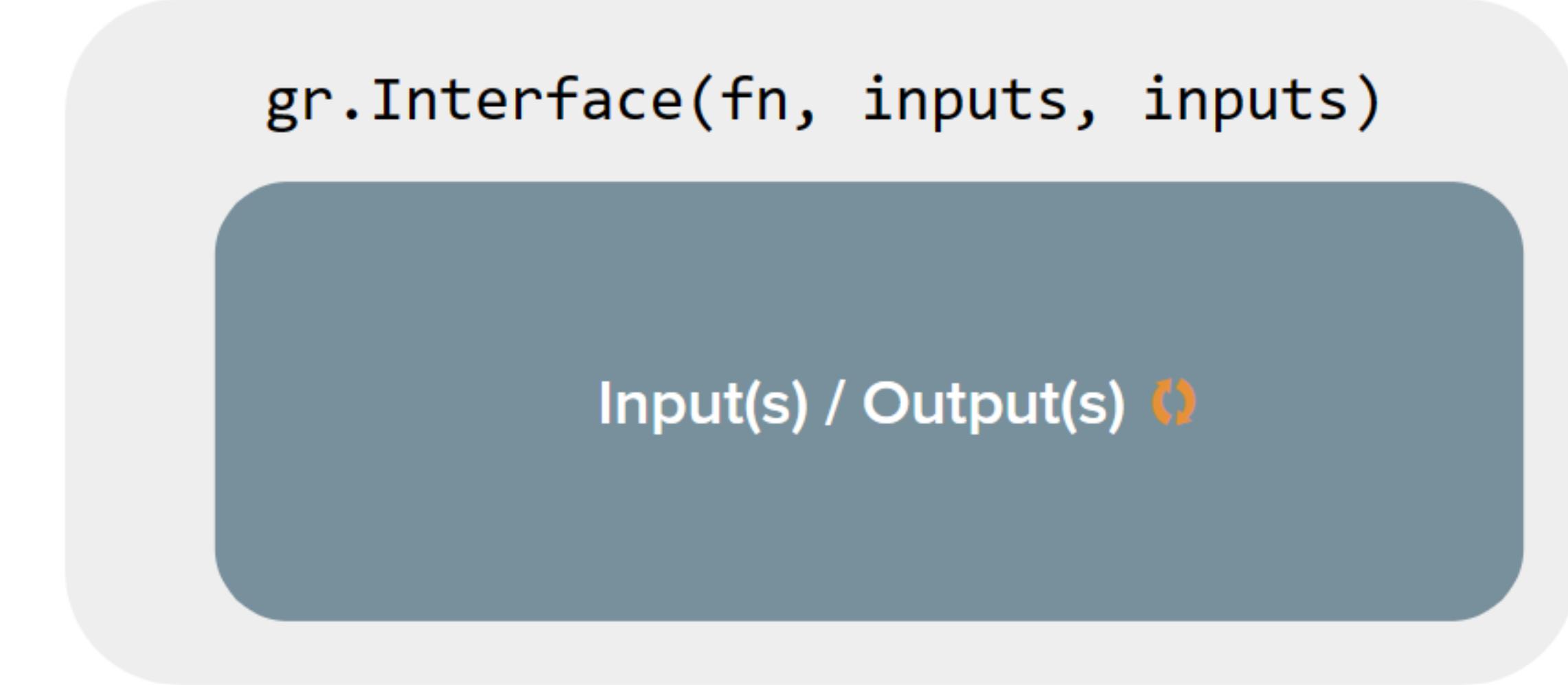
Output-Only Interface (e.g. GAN)

`gr.Interface(fn, inputs, None)`



Input-Only Interface (e.g. Data Collector)

`gr.Interface(fn, inputs, inputs)`



Unified Interface (e.g. Text Generator)



Useful components.



Input Components (User-provided data)

Components

- **gr.Textbox()** → Accepts text input (single/multiline).
- **gr.Number()** → Accepts numerical input.
- **gr.Slider()** → Provides a numeric slider within a range.
- **gr.Checkbox()** → Boolean toggle (True/False).
- **gr.CheckboxGroup()** → Multiple checkbox selections.
- **gr.Radio()** → Single choice from a list.
- **gr.Dropdown()** → Dropdown menu for selecting one or multiple options.
- **gr.File()** → Allows file uploads.
- **gr.Image()** → Accepts images (uploads or drawings).
- **gr.Video()** → Accepts video uploads.
- **gr.Audio()** → Accepts audio files or recordings.
- **gr.ColorPicker()** → Lets users pick a color.
- **gr.DataFrame()** → Editable table for structured input.

✍️ Input Components (User-provided data)

Components

- There are some nuances that really matter.
- Make sure to pick the most fitting component!

What burger do you want?

- Hamburger
- Chicken burger
- Veggie burger
- Mushroom burger
- Beef burger
- Bacon burger
- Bunless burger
- Ostrich burger

What burger do you want?

- Chicken burger

✖️ ✅

Pizza toppings

- Bacon
- Brie
- Extra cheese
- Feta
- Ham
- Mushrooms
- Onions
- Olives
- Peppadews
- Pepperoni
- Salami

✖️ ✅

✓ Remember me

✓ I agree to the terms of service

✖️ ✅

Your favorite meal:

- burger
- pizza
- noodles
- fries
- chips

✖️ ✅

Would you like to receive notifications?

- Yes
- No

Receive notifications

✖️ ✅



Output Components (Display results)

Components

- **gr.Textbox()** → Displays text output.
- **gr.Label()** → Displays labeled predictions or results.
- **gr.Number()** → Shows numeric output.
- **gr.Image()** → Displays images (e.g., processed images).
- **gr.Video()** → Displays videos.
- **gr.Audio()** → Plays back audio output.
- **gr.HTML()** → Renders custom HTML content.
- **gr.JSON()** → Displays structured JSON data.
- **gr.Plot()** → Renders matplotlib or plotly graphs.



Layout Components (Organizing UI)

Components

- **gr.Row()** → Places components side by side.
- **gr.Column()** → Stacks components vertically.
- **gr.Tab()** → Organizes components into different tabs.
- **gr.Accordion()** → Collapsible sections for better organization.
- These will naturally become relevant as the input or output grows in size or complexity.



Styling

Themes

Defaults or custom

- There are a couple of built-in themes.
- Themes can easily be changed or overridden.

```
import gradio as gr

def greet(name):
    return f"Hello, {name}!"

theme = gr.themes.Default(primary_hue="blue", secondary_hue="red")

iface = gr.Interface(fn=greet, inputs=gr.Textbox(), outputs=gr.Textbox(), theme=theme)
iface.launch()
```

Themes

Start with a theme

- Themes are available as python files and can be changed.
- There are three main colors:
 -  **primary_hue**: This is the color draws attention in your theme.
 -  **secondary_hue**: This is the color that is used for secondary elements in your theme.
 -  **neutral_hue**: This is the color that is used for text and other neutral elements in your theme.
- Styles can be further customized: <https://www.gradio.app/guides/theming-guide>.

Styling of a TextBox

What are the default options?

- When you look at the docs, the *pure design options* are limited.
- This is not per se a bad thing.
- **Don't overdesign** the basic building blocks of a UI.

Parameters
value: str Callable None
lines: int
max_lines: int
placeholder: str None
label: str None
info: str None
every: Timer float None
inputs: Component list[Component] set[Component] None
show_label: bool None
container: bool
scale: int None
min_width: int
interactive: bool None
visible: bool
elem_id: str None
autofocus: bool
autoscroll: bool
elem_classes: list[str] str None
render: bool
key: int str None
type: Literal['text', 'password', 'email']
text_align: Literal['left', 'right'] None
rtl: bool
show_copy_button: bool
max_length: int None
submit_btn: str bool None
stop_btn: str bool None



Deployment

Share=True

Easy peasy

```
demo.launch(share=True) # Share your demo with just 1 extra parameter 🚀
```

- This generates a public, shareable link that you can send to anybody!
- When you send this link, the user on the other side can try out the model in their browser.
- Because the processing happens on your device (as long as your device stays on), you don't have to worry about any packaging any dependencies.
- You can also share it on hugging space.
- Be careful what you deploy just randomly on the internet.



Getting started

To create a Python virtual environment (**venv**), follow these steps:

1. Open a Terminal or Command Prompt

- On **Windows**, use `cmd`, PowerShell, or Windows Terminal.
- On **macOS/Linux**, use the Terminal.



Optional

2. Navigate to Your Project Folder

Go to the directory where you want to create the virtual environment:

```
```sh
cd path/to/your/project
````
```

3. Create the Virtual Environment

Run the following command:

```
```sh
python -m venv venv
````
```

This creates a folder named `venv` containing the virtual environment.

4. Activate the Virtual Environment

- **Windows (Command Prompt)**:

```
```sh
venv\Scripts\activate
````
```

- **Windows (PowerShell)**:

```
```powershell
venv\Scripts\Activate.ps1
````
```

(If you get a security error, try: `Set-ExecutionPolicy Unrestricted -Scope Process`)

- **macOS/Linux**:

```
```sh
source venv/bin/activate
````
```



Make sure to work in a virtual environment.

5. Verify the Virtual Environment

Run:

```
```sh
python -V
````
```

It should show the Python version from the virtual environment.

6. Deactivate When Done

To exit the virtual environment, run:

```
```sh
deactivate
````
```

Install

The package itself

- First, install the package: `pip install --upgrade gradio`
- Next, the code is always the same structure:

```
import gradio as gr

def greet(name, intensity):
    return "Hello, " + name + "!" * int(intensity)

demo = gr.Interface(
    fn=greet,
    inputs=["text", "slider"],
    outputs=["text"],
)

demo.launch()
```

fn

Defined by the inputs and the outputs

```
import gradio as gr

def greet(name, is_morning, temperature):
    salutation = "Good morning" if is_morning else "Good evening"
    greeting = f"{salutation} {name}. It is {temperature} degrees today"
    celsius = (temperature - 32) * 5 / 9
    return greeting, round(celsius, 2)

demo = gr.Interface(
    fn=greet,
    inputs=["text", "checkbox", gr.Slider(0, 100)],
    outputs=["text", "number"],
)

demo.launch()
```



Alternative



My Favorite
Text Editor



Exercise



A Gradio app

Temperature conversion

Exercise 1

- We'll create a simple first app to convert temperature units.
 - Celsius to Fahrenheit: $(C \times 9/5) + 32$
 - Fahrenheit to Celsius: $(F - 32) \times 5/9$
- What should the inputs be?
- What should be the output(s)?

Bird classifier

Exercise 2

- For this exercise, we will connect to the FastAPI API that we created last week.
- Think what the API can do and how this can help here.
- What are the inputs, what are the outputs?
- Start with a simple app that allows the selection of species and other properties from the data.

A simple image classification

Exercise 3

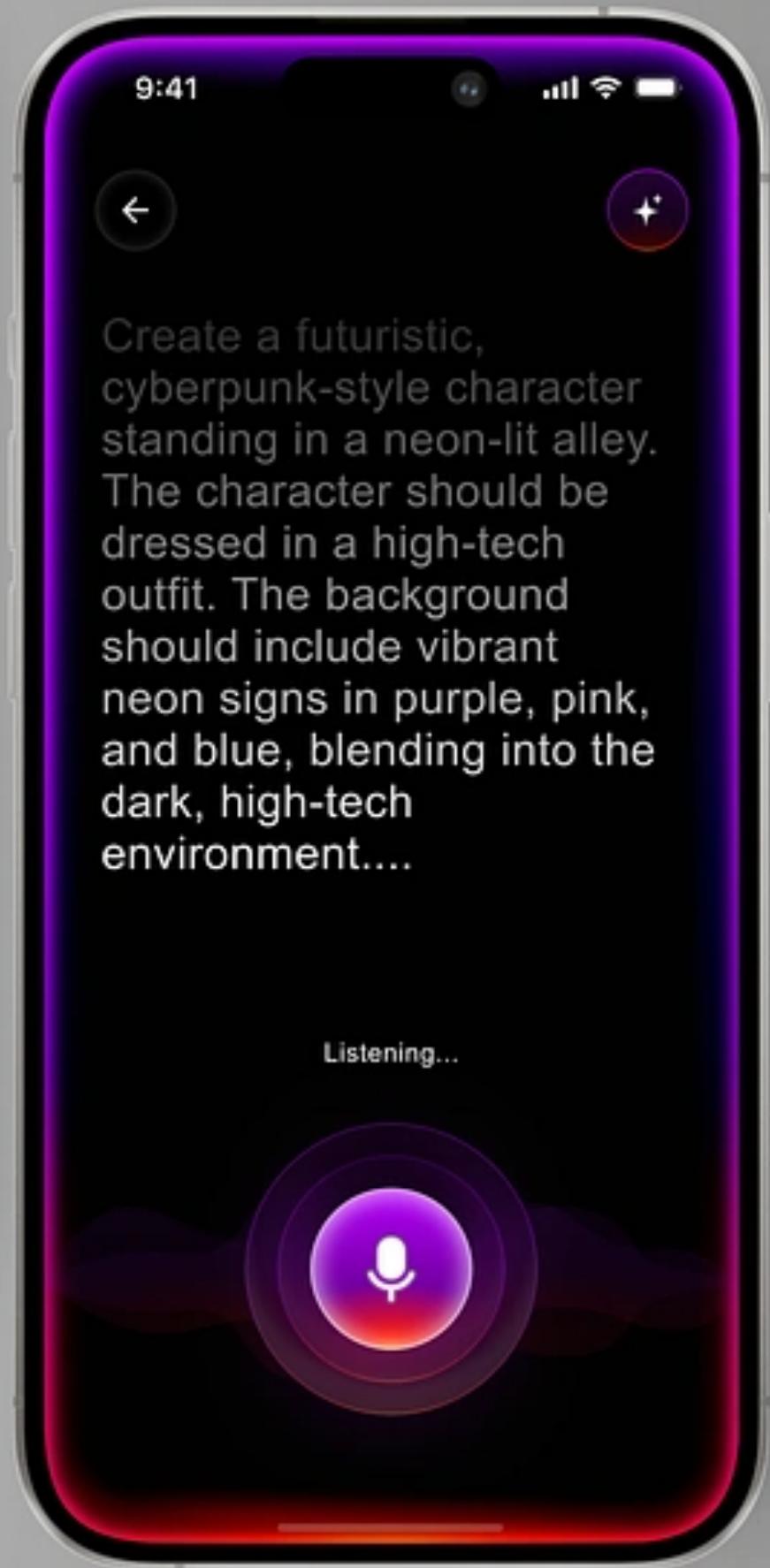
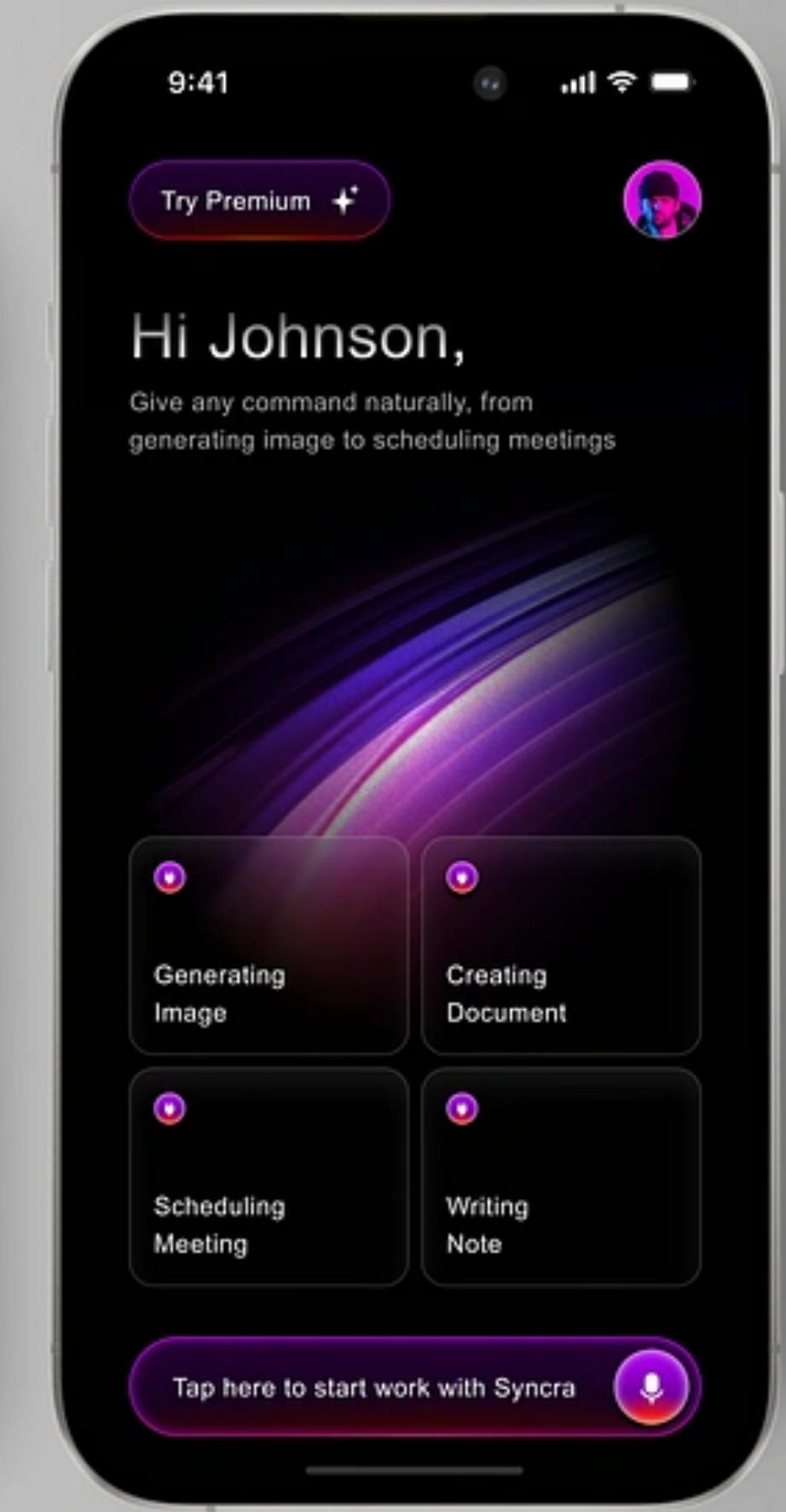
- Next, we will try to create a simple image based model.
- Grab a picture, and generate some output:
 1. Start by simply converting the image to grayscale.
 2. Next, output some details of the image (can be any info).
 3. Finally, try to recognize what is in the picture.



Optional



Can we make... ?



The screenshot shows the 'Talk Data to Me' feature within the Zoho CRM interface. On the left, there's a vertical sidebar with icons for Home, Recent, and Products. The main area features a large search bar at the top with placeholder text 'Ask AI a question or make a request...'. Below the search bar are several buttons for common tasks: 'Clean account fields', 'Clean contact fields', 'Create master 'People' list', 'Account Fit Score', 'Match leads to account', and a 'See prompt library' button. At the bottom of the search bar is a dropdown menu labeled 'Select source' and a character icon representing the AI agent. The overall design is clean and modern, with a light gray background.

Ask AI a question or make a request...

Select source

0/2000

Ask about:

- Clean account fields
- Clean contact fields
- Create master 'People' list
- Account Fit Score
- Match leads to account
- See prompt library

Talk Data to Me

Choose a prompt below or write your own to start chatting with Seam

Yes!

Euh... maybe.

- We can customize blocks and components with HTML & CSS.
- HTML & CSS are the building blocks that are used in professional websites and web apps.
- CSS contains styles that are being applied to HTML blocks:

```
with gr.Blocks(css=".gradio-container {background-color: red}") as demo:
```

- Gradio will mostly just give you an environment to show functionality and demo features.
- And beyond! Should you?

Recap

What is Gradio?

- How to install the package?
- What are the core concepts of a Gradio app?
- What are the main components?
- How to pick the right input components?
- How to change the style of a Gradio app?
- What is the interface class?
- How to get started with more complex layouts?



Done.