The Interface

Gradio can wrap almost any Python function with an easy to use interface. That function could be anything from a simple tax calculator to a pretrained model.

The core `Interface` class is initialized with three parameters:

- `fn`: the function to wrap
- `inputs`: the input component type(s)
- `outputs`: the output component type(s)

With these three arguments, we can quickly create interfaces and `launch()` them. But what if you want to change how the UI components look or behave?

Customizable Components

What if we wanted to customize the input text field - for example, we wanted it to be larger and have a text hint? If we use the actual input class for `Textbox` instead of using the string shortcut, we have access to much more customizability. To see a list of all the components we support and how you can customize them, check out the Docs

```
import gradio as gr

def greet(name):
  return "Hello " + name + "!"

iface = gr.Interface(
  fn=greet,
  inputs=gr.inputs.Textbox(lines=2, placeholder="Name Here..."),
  outputs="text")
iface.launch()
```

The Interface

Gradio can wrap almost any Python function with an easy to use interface. That function could be anything from a simple tax calculator to a pretrained model.

The core `Interface` class is initialized with three parameters:

- `fn`: the function to wrap
- `inputs`: the input component type(s)
- `outputs`: the output component type(s)

With these three arguments, we can quickly create interfaces and `launch()` them. But what if you want to change how the UI components look or behave?

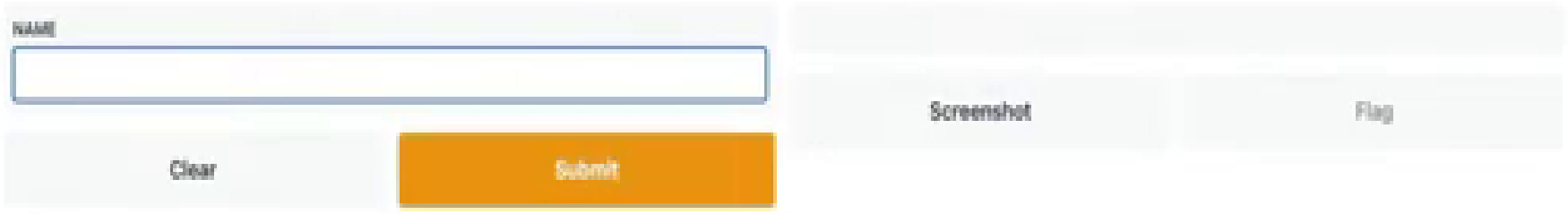
Customizable Components

What if we wanted to customize the input text field - for example, we wanted it to be larger and have a text hint? If we use the actual input class for `Textbox` instead of using the string shortcut, we have access to much more customizability. To see a list of all the components we support and how you can customize them, check out the Docs

```
import gradio as gr

def greet(name):
  return "Hello " + name + "!"

iface = gr.Interface(
  fn=greet,
  inputs=gr.inputs.Textbox(lines=2, placeholder="Name Here..."),
  outputs="text")
iface.launch()
```

```
import gradio as gr

def greet(name):
  return "Hello " + name + "!!"

iface = gr.Interface(fn=greet, inputs="text", outputs="text")
iface.launch()
```

3. The interface below will appear automatically within the Python notebook, or pop in a browser on http://localhost:7860 if running from a script.



The Interface

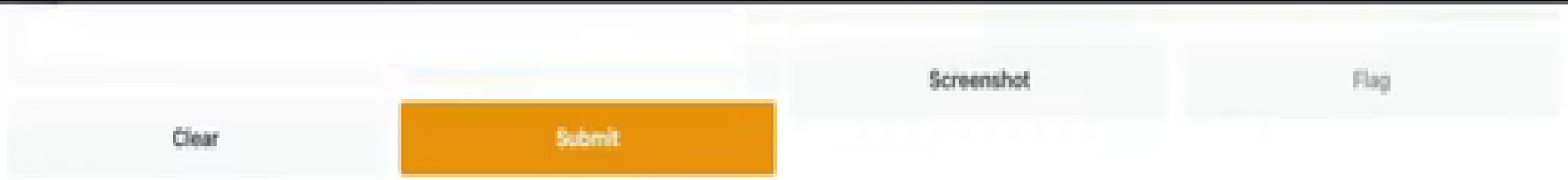
Gradio can wrap almost any Python function with an easy to use interface. That function could be anything from a simple tax calculator to a pretrained model.

The core `Interface` class is initialized with three parameters:

- `fn`: the function to wrap
- `inputs`: the input component type(s)
- `outputs`: the output component type(s)

With these three arguments, we can quickly create interfaces and `launch()` them. But what if you want to change how the UI components look or behave?

Customizable Components



The Interface

Gradio can wrap almost any Python function with an easy to use interface. That function could be anything from a simple tax calculator to a pretrained model.

The core `Interface` class is initialized with three parameters:

- `fn`: the function to wrap
- `inputs`: the input component type(s)
- `outputs`: the output component type(s)

With these three arguments, we can quickly create interfaces and `launch()` them. But what if you want to change how the UI components look or behave?

Customizable Components

What if we wanted to customize the input text field - for example, we wanted it to be larger and have a text hint? If we use the actual input class for `Textbox` instead of using the string shortcut, we have access to much more customizability. To see a list of all the components we support and how you can customize them, check out the Docs

```
import gradio as gr

def greet(name):
  return "Hello " + name + "!"

iface = gr.Interface(
  fn=greet,
  inputs=gr.inputs.Textbox(lines=2, placeholder="Name Here..."),
```

1. Install Gradio from pip.

```
pip install gradio
```

2. Run the code below as a Python script or in a Python notebook (or in a colab notebook).

```
import gradio as gr

def greet(name):
  return "Hello " + name + "!!"

iface = gr.Interface(fn=greet, inputs="text", outputs="text")
iface.launch()
```

3. The interface below will appear automatically within the Python notebook, or pop in a browser on http://localhost:7860 if running from a script.

The Interface

Gradio can wrap almost any Python function with an easy to use interface. That function could be anything from a simple tax calculator to a pretrained model.

The core `Interface` class is initialized with three parameters:

The Interface

Gradio can wrap almost any Python function with an easy to use interface. That function could be anything from a simple tax calculator to a pretrained model.

The core `Interface` class is initialized with three parameters:

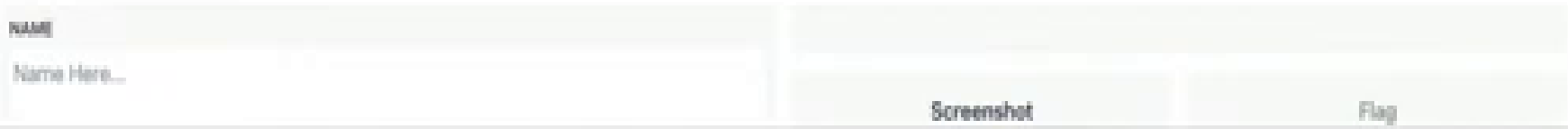
- `fn`: the function to wrap
- `inputs`: the input component type(s)
- `outputs`: the output component type(s)

With these three arguments, we can quickly create interfaces and `launch()` them. But what if you want to change how the UI components look or behave?

Customizable Components

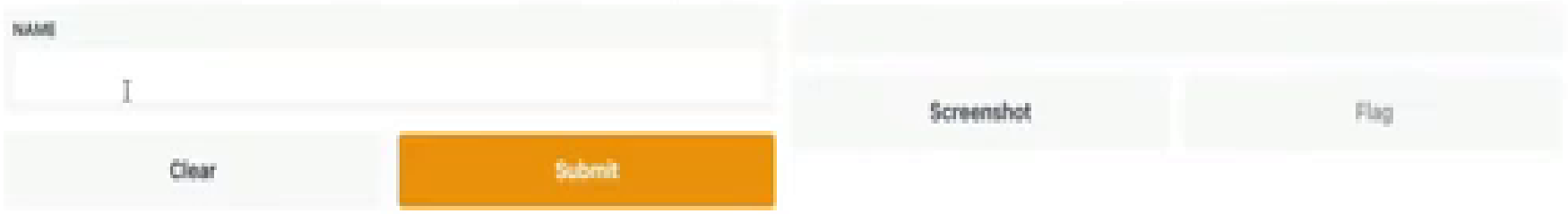
What if we wanted to customize the input text field - for example, we wanted it to be larger and have a text hint? If we use the actual input class for `Textbox` instead of using the string shortcut, we have access to much more customizability. To see a list of all the components we support and how you can customize them, check out the **Docs**

```
import gradio as gr

def greet(name):
  return "Hello " + name + "!"

iface = gr.Interface(
  fn=greet,
  inputs=gr.inputs.Textbox(lines=2, placeholder="Name Here..."),
  outputs="text")
iface.launch()
```



```
import gradio as gr

def greet(name):
  return "Hello " + name + "!!"

iface = gr.Interface(fn=greet, inputs="text", outputs="text")
iface.launch()
```

3. The interface below will appear automatically within the Python notebook, or pop in a browser on http://localhost:7860 if running from a script.



The Interface

Gradio can wrap almost any Python function with an easy to use interface. That function could be anything from a simple tax calculator to a pretrained model.

The core `Interface` class is initialized with three parameters:

- `fn`: the function to wrap
- `inputs`: the input component type(s)
- `outputs`: the output component type(s)

With these three arguments, we can quickly create interfaces and `launch()` them. But what if you want to change how the UI components look or behave?

- `inputs`: the input component type(s)
- `outputs`: the output component type(s)

With these three arguments, we can quickly create interfaces and `launch()` them. But what if you want to change how the UI components look or behave?

Customizable Components

What if we wanted to customize the input text field - for example, we wanted it to be larger and have a text hint? If we use the actual input class for Textbox instead of using the string shortcut, we have access to much more customizability. To see a list of all the components we support and how you can customize them, check out the Docs

```
import gradio as gr

def greet(name):
  return "Hello " + name + "!"

iface = gr.Interface(
  fn=greet,
  inputs=gr.inputs.Textbox(lines=2, placeholder="Name Here..."),
  outputs="text")
iface.launch()
```

Multiple Inputs and Outputs

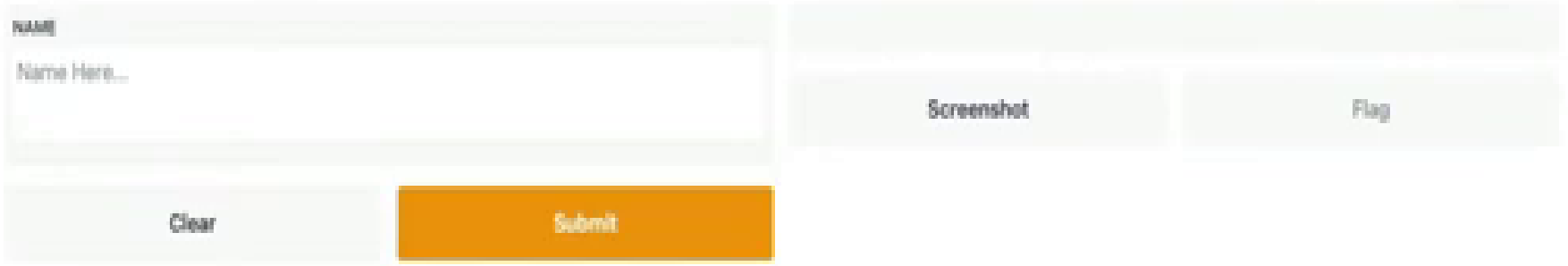
- `inputs`: the input component type(s)
- `outputs`: the output component type(s)

With these three arguments, we can quickly create interfaces and `launch()` them. But what if you want to change how the UI components look or behave?

Customizable Components

What if we wanted to customize the input text field - for example, we wanted it to be larger and have a text hint? If we use the actual input class for `Textbox` instead of using the string shortcut, we have access to much more customizability. To see a list of all the components we support and how you can customize them, check out the Docs

```
import gradio as gr

def greet(name):
  return "Hello " + name + "!"

iface = gr.Interface(
  fn=greet,
  inputs=gr.inputs.Textbox(lines=2, placeholder="Name Here..."),
  outputs="text")
iface.launch()
```



Multiple Inputs and Outputs

- outputs: the output component type(s)

With these three arguments, we can quickly create interfaces and `launch()` them. But what if you want to change how the UI components look or behave?

Customizable Components

What if we wanted to customize the input text field - for example, we wanted it to be larger and have a text hint? If we use the actual input class for `Textbox` instead of using the string shortcut, we have access to much more customizability. To see a list of all the components we support and how you can customize them, check out the **Docs**

```
import gradio as gr

def greet(name):
  return "Hello " + name + "!"

iface = gr.Interface(
  fn=greet,
  inputs=gr.inputs.Textbox(lines=2, placeholder="Name Here..."),
  outputs="text")
iface.launch()
```

Multiple Inputs and Outputs

Let's say we had a much more complex function, with multiple inputs and outputs. In the example below, we have a function that takes a string, boolean, and number, and returns a string

00:02:09

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
interface = gr.Interface(fn = hello_world, inputs = 'text', outputs = "text")
```

```
interface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://31668.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

NAME

Gradio

OUTPUT 0.00s

Hello.... ...Gradio !!!!

Clear Submit Screenshot Flag

2s completed at 4:47 PM

00:02:16

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk

Editing

[10]

view the api | built with

```
(<Flask 'gradio.networking'>,
 'http://127.0.0.1:7860/',
 'https://31668.gradio.app')
```

```
interface = gr.Interface(fn = hello_world, inputs = 'text', outputs = "text")
```

2s completed at 4:47 PM

00:02:26

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share A

+ Code + Text

RAM Disk Editing

[10]

view the api | built with

```
(<Flask 'gradio.networking'>,
 'http://127.0.0.1:7860/',
 'https://31668.gradio.app')
```

```
interface = gr.Interface(fn = hello_world,
                         inputs = 'text',
                         outputs = 'text')
```

2s completed at 4:47 PM

00:02:31

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

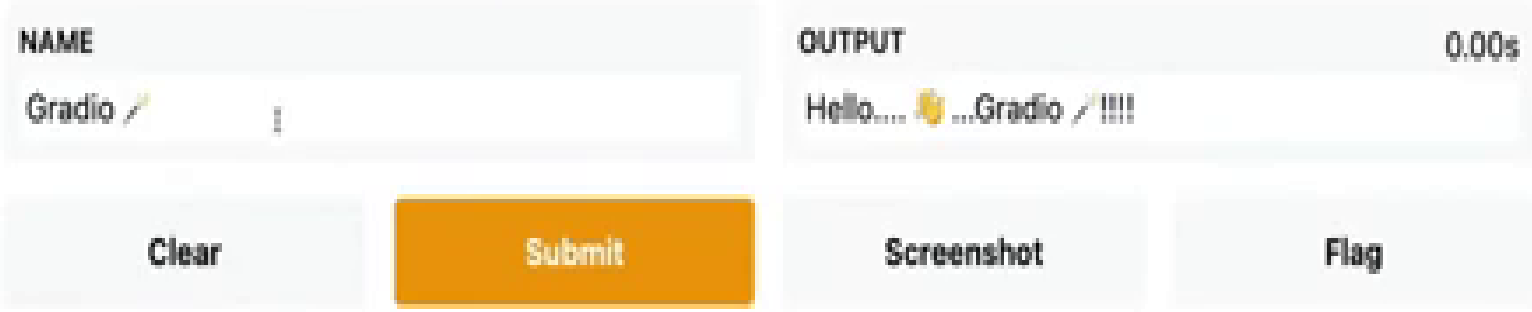
RAM Disk

Editing

```
[9] interface = gr.Interface(fn = hello_world, inputs = 'text', outputs = "text")
```

```
interface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://31668.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```



2s completed at 4:47 PM

00:02:39

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

[10]

```
(<Flask 'gradio.networking'>,
 'http://127.0.0.1:7860/',
 'https://31668.gradio.app')
```

```
interface = gr.Interface(fn = hello_wo
                         inputs = gr.i
                         outputs = "text")
```

inputs module
inspect
interface
interpretation
ip_address
Interface
get_input_instance
load_interface
get_output_instance
quantify_difference_in_label

2s completed at 4:47 PM

00:03:16

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[10]

view the api | built with

```
(<Flask 'gradio.networking'>,
 'http://127.0.0.1:7860/',
 'https://31668.gradio.app')
```

[11]
```
interface = gr.Interface(fn = hello_world,
                         inputs = gr.inputs.Textbox(lines = 5, placeholder="Enter your input here....."),
                         outputs = "text")
```

```
interface.launch()
```

```
<bound method Interface.launch of Gradio Interface for: hello_world
---------------------------------
inputs:
|-Textbox(label="None")
outputs:
|-Textbox(label="None")>
```

0s completed at 5:03 PM

00:03:21

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help Saving...

Comment Share

+ Code + Text

RAM Disk

Editing

```
[11] interface = gr.Interface(fn = hello_world,
                              inputs = gr.inputs.Textbox(lines = 5, placeholder="Enter your input here....."),
                              outputs = "text")
```

```
interface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://22229.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

Executing (3s) C... > laun... > url ... > hea... > reque... > reque... > sen... > sen... > urlop... > make requ... > getrespon... > beg... > read stat... > readin... > recv in... > rea...

00:03:24

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

+ Code + Text

RAM Disk Editing

NAME

Enter your input here.....

Screenshot Flag

Clear Submit

3s completed at 5:03 PM

00:03:28

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[11]  inputs = gr.inputs.Textbox(lines = 5, placeholder="Enter your input here....."),
      outputs = "text")
```

```
interface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://22229.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

NAME

Enter your input here.....

Screenshot Flag

Clear Submit

3s completed at 5:03 PM

00:03:31

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

+ Code + Text

RAM Disk Editing

https://1b9d.gradio.app)

```
interface = gr.Interface(fn = hello_world,
                         inputs = gr.inputs.Textbox(lines = 10, placeholder="Enter your input here....."),
                         outputs = "text")
```

```
interface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://22229.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

3s completed at 5:03 PM

00:03:34

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

+ Code + Text

RAM Disk Editing

```
!interface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
```

Executing (1s) Cell > launch() > setup_tunnel() > create_tunnel() > connect() > start_client() > wait() > wait()

00:03:35

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

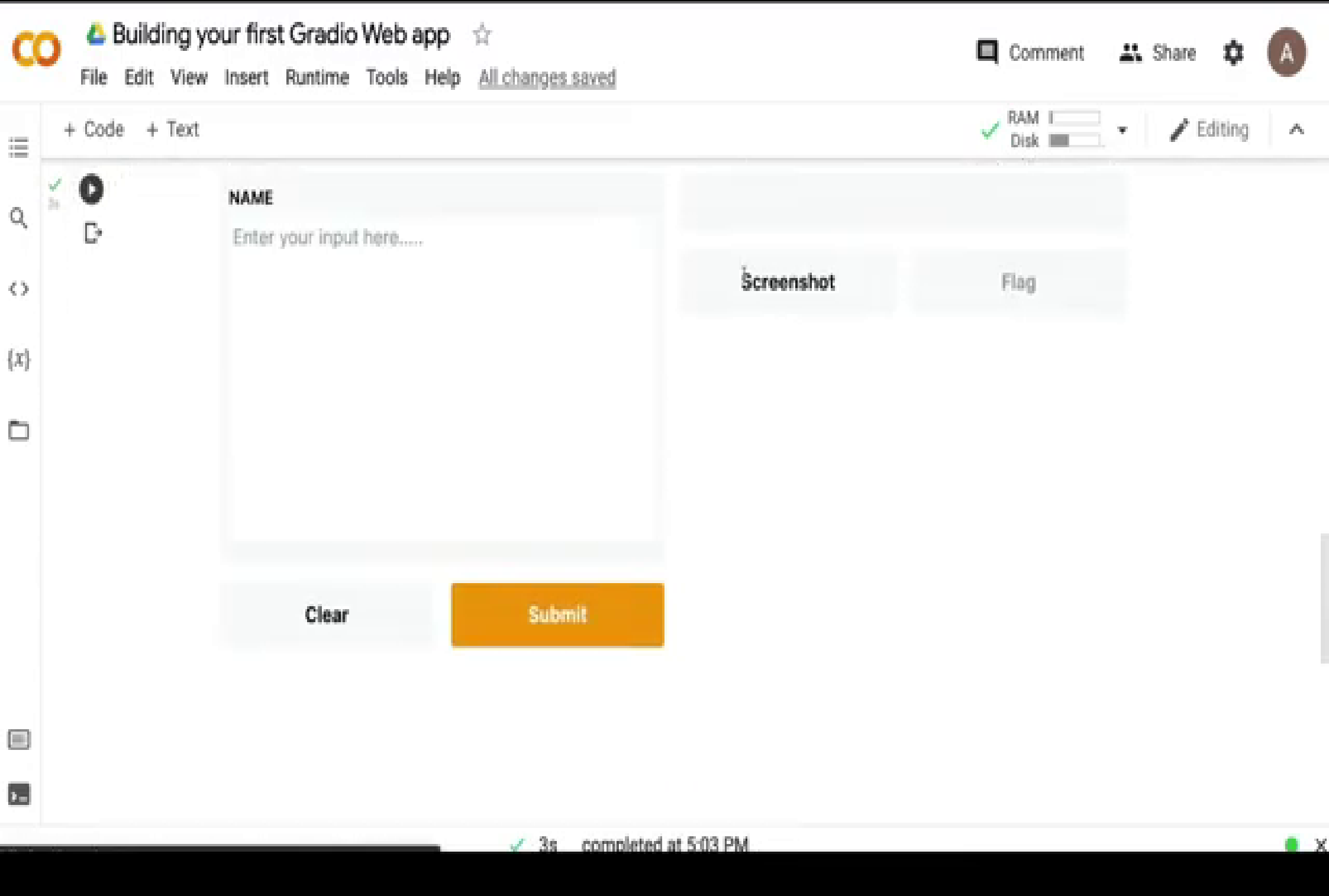
```
interface = gr.Interface(fn = hello_world,
                         inputs = gr.inputs.Textbox(lines = 10, placeholder="Enter your input here....."),
                         outputs = "text")
```

```
!interface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://44140.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

Executing (2s) C... > laun... > url... > hea... > reque... > reque... > sen... > sen... > urlop... > make requ... > getresponse... > beg... > read stat... > readin... > recv in... > rea...

00:03:37


Building your first Gradio Web app
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
A
+ Code + Text
RAM
Disk
Editing
NAME
Enter your input here.....
Screenshot
Flag
Clear
Submit
3s completed at 5:03 PM

00:03:43

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
        'http://127.0.0.1:7860/',
        'https://31668.gradio.app')
```

```
interface = gr.Interface(fn = hello_world,
                         inputs = gr.inputs.Textbox(lines = 10, placeholder="Enter your input here....."),
                         outputs = "text")
```

```
[15] interface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://44140.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

NAME

Enter your input here.....

Screenshot Flag

3s completed at 5:03 PM

00:04:06

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
        'http://127.0.0.1:7860/',
        'https://31668.gradio.app')
```

```
[14] interface = gr.Interface(fn = hello_world,
                              inputs = gr.inputs.Textbox(lines = 10, placeholder="Enter your input here....."),
                              outputs = "text")
```

```
interface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://44140.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

NAME

Enter your input here.....

Screenshot Flag

3s completed at 5:03 PM

00:04:11

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Editing

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://44140.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

NAME

Enter your input here.....

Screenshot

Flag

Clear

Submit

3s completed at 5:03 PM

00:04:33

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
        'http://127.0.0.1:7860/',
        'https://31668.gradio.app')
```

```
[14] interface = gr.Interface(fn = hello_world,
                              inputs = gr.inputs.Textbox(lines = 10, placeholder="Enter your input here....."),
                              outputs = "text")
```

```
[15] interface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://44140.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

NAME

Enter your input here.....

Screenshot Flag

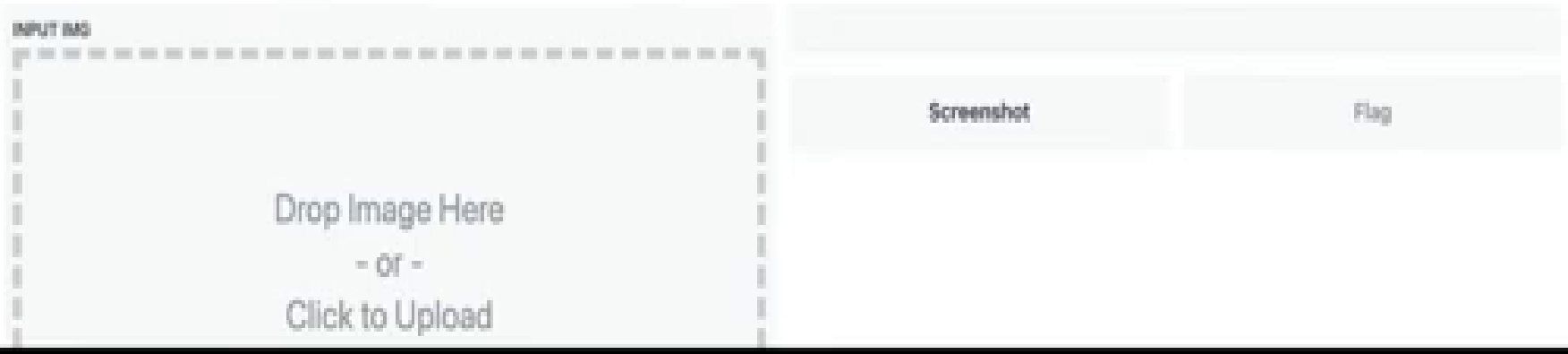
3s completed at 5:03 PM

comparison.

Working with Images

Let's try an image to image function. When using the `Image` component, your function will receive a numpy array of your specified size, with the shape `(width, height, 3)`, where the last dimension represents the RGB values. We'll return an image as well in the form of a numpy array.

```
import gradio as gr
import numpy as np

def sepia(input_img):
  sepia_filter = np.array([[.393, .769, .189],
                           [.349, .686, .168],
                           [.272, .534, .131]])
  sepia_img = input_img.dot(sepia_filter.T)
  sepia_img /= sepia_img.max()
  return sepia_img

iface = gr.Interface(sepia, gr.inputs.Image(shape=(200, 200)), "image")

iface.launch()
```



00:04:42

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

+ Code + Text

RAM Disk Editing

[15]

view the api | built with

```
(<Flask 'gradio.networking'>,
 'http://127.0.0.1:7862/',
 'https://44140.gradio.app')
```

3s completed at 5:03 PM

00:05:02

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[15]

view the api | built with

```
(<Flask 'gradio.networking'>,
 'http://127.0.0.1:7862/',
 'https://44140.gradio.app')
```

```
import gradio as gr
import numpy as np

def sepia(input_img):
  sepia_filter = np.array([[.393, .769, .189],
                           [.349, .686, .168],
                           [.272, .534, .131]])
  sepia_img = input_img.dot(sepia_filter.T)
  sepia_img /= sepia_img.max()
  return sepia_img

iface = gr.Interface(sepia, gr.inputs.Image(shape=(200, 200)), "image")

iface.launch()
```

3s completed at 5:03 PM

00:05:03

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[15]

view the api | built with

```
(<Flask 'gradio.networking'>,
 'http://127.0.0.1:7862/',
 'https://44140.gradio.app')
```

```
import gradio as gr
import numpy as np

def sepia(input_img):
  sepia_filter = np.array([[.393, .769, .189],
                           [.349, .686, .168],
                           [.272, .534, .131]])
  sepia_img = input_img.dot(sepia_filter.T)
  sepia_img /= sepia_img.max()
  return sepia_img

iface = gr.Interface(sepia, gr.inputs.Image(shape=(200, 200)), "image")

iface.launch()
```

3s completed at 5:03 PM

00:05:10

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[10]
```
(<Flask 'gradio.networking'>,
 'http://127.0.0.1:7860/',
 'https://31668.gradio.app')
```

```
interface = gr.Interface(fn = hello_world,
                         inputs = gr.inputs.Textbox(lines = 10, placeholder="Enter your input here....."),
                         outputs = "text")
```

[15]
```
interface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://44140.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

NAME

Enter your input here.....

Screenshot Flag

3s completed at 5:03 PM

00:05:19

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

'https://44140.gradio.app')

```
import gradio as gr
import numpy as np

def sepia(input_img):
  sepia_filter = np.array([[.393, .769, .189],
                           [.349, .686, .168],
                           [.272, .534, .131]])
  sepia_img = input_img.dot(sepia_filter.T)
  sepia_img /= sepia_img.max()
  return sepia_img

iface = gr.Interface(sepia, gr.inputs.Image(shape=(200, 200)), "image")

iface.launch()
```

3s completed at 5:03 PM

00:05:40

Building your first Gradio Web app

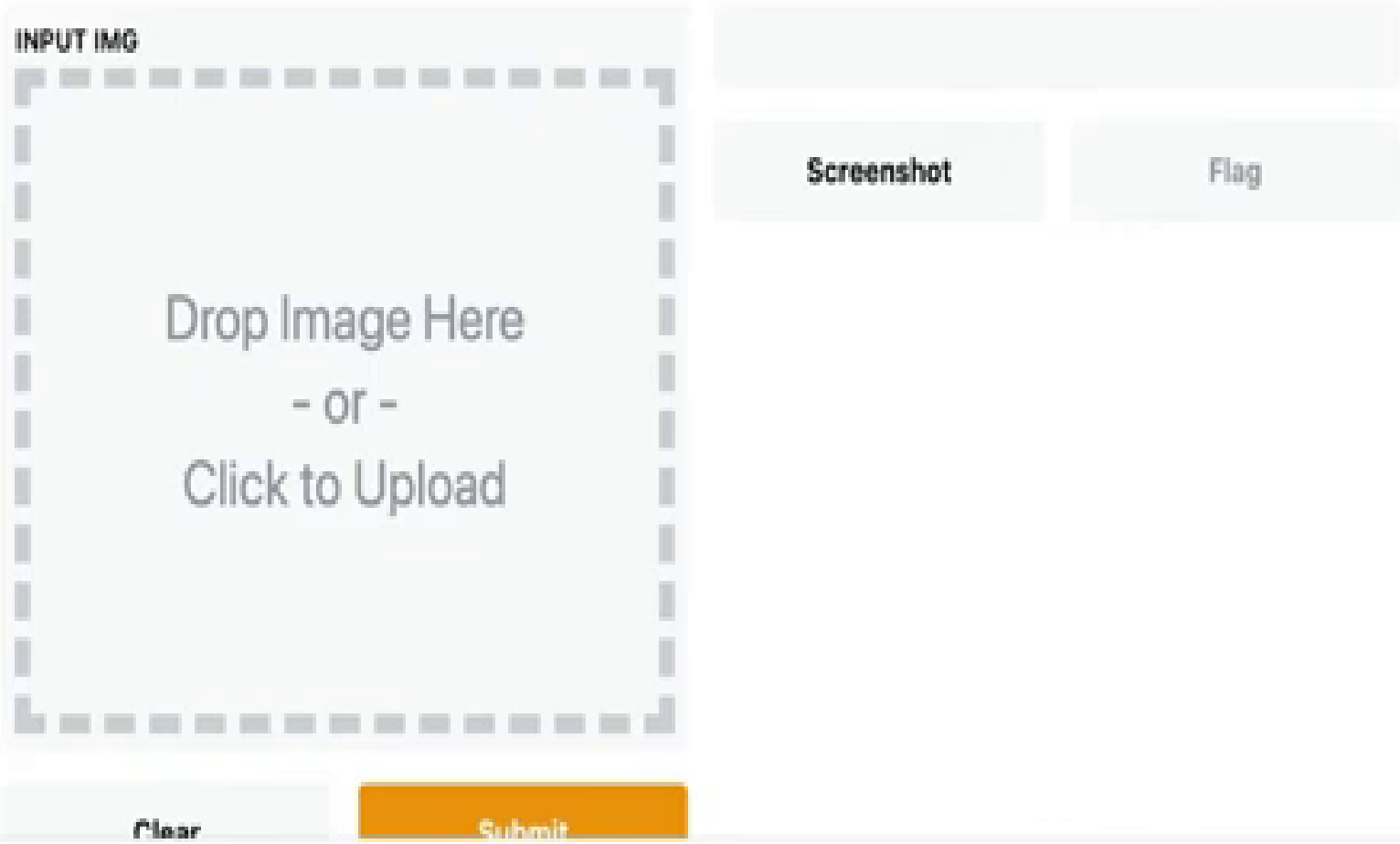
File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[16]
```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://15052.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```



3s completed at 5:05 PM

00:05:49

Building your first Gradio Web app

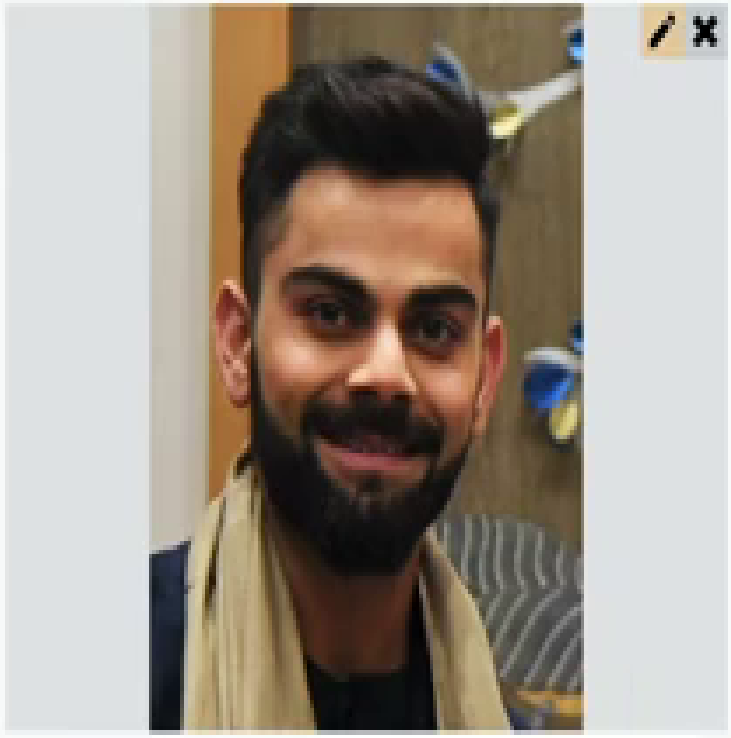
File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

+ Code + Text

RAM Disk Editing

INPUT IMG



2.0s

Screenshot Flag

Clear Submit

3s completed at 5:05 PM

00:05:52

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

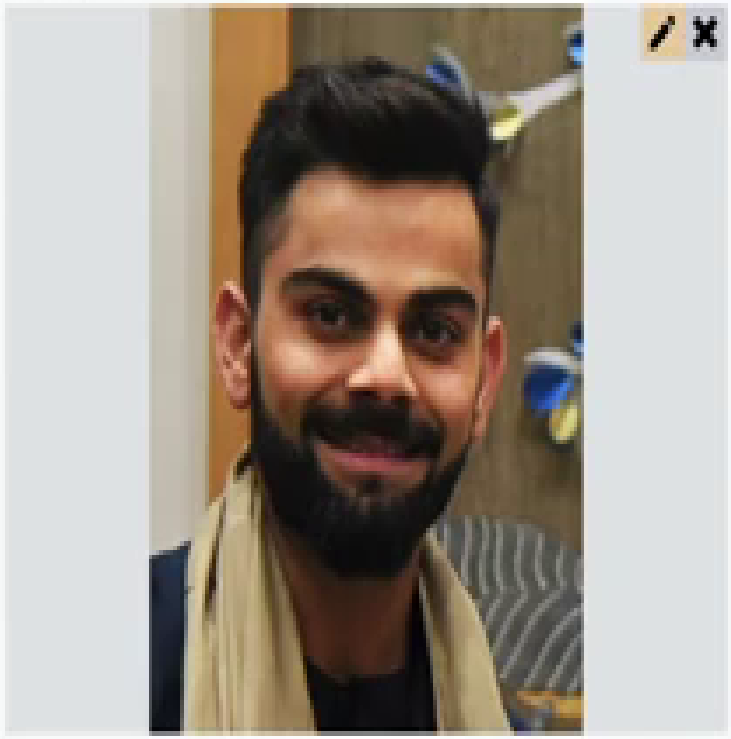
+ Code + Text

RAM Disk

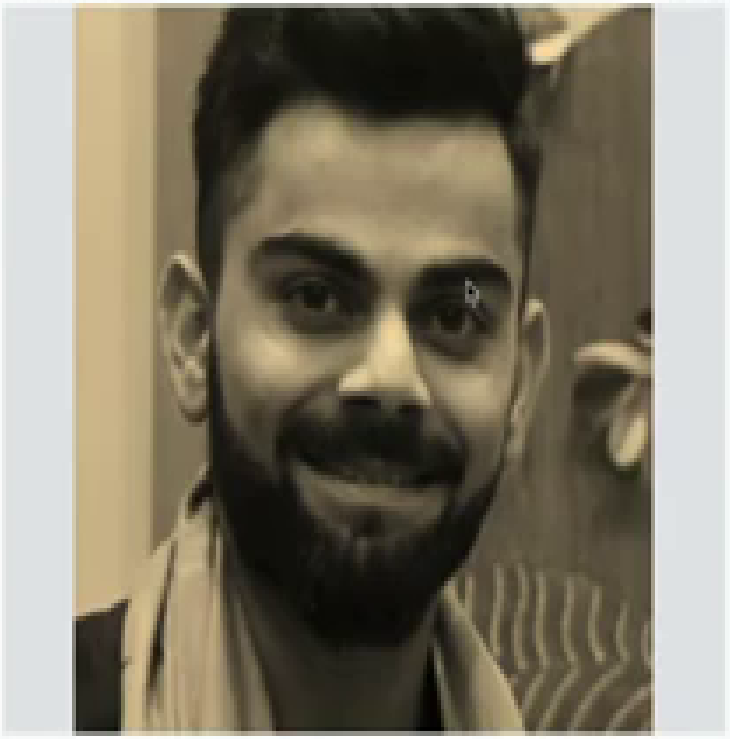
Editing

```
Running on public URL: https://15052.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

INPUT IMG



OUTPUT 0.01s



Clear Submit Screenshot Flag

3s completed at 5:05 PM

00:06:07

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

```
import gradio as gr
import numpy as np

def sepia(input_img):
  sepia_filter = np.array([[.393, .769, .189],
                           [.349, .686, .168],
                           [.272, .534, .131]])
  sepia_img = input_img.dot(sepia_filter.T)
  sepia_img /= sepia_img.max()
  return sepia_img

iface = gr.Interface(sepia, gr.inputs.Image(shape=(200, 200)), "image")

iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://15052.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

3s completed at 5:05 PM

00:06:26

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

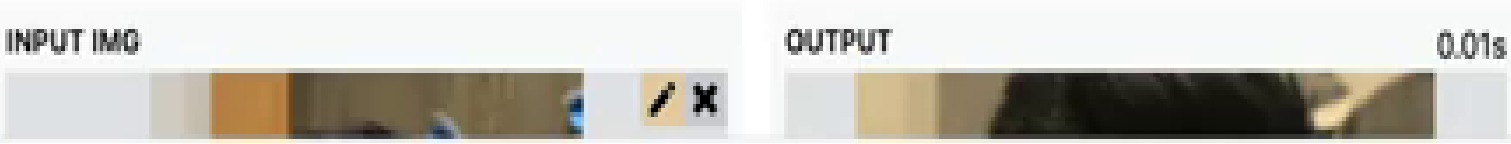
Comment Share

+ Code + Text

RAM Disk Editing

```
import gradio as gr
import numpy as np

def sepia(input_img):
  sepia_filter = np.array([[.393, .769, .189],
                           [.349, .686, .168],
                           [.272, .534, .131]])
  sepia_img = input_img.dot(sepia_filter.T)
  sepia_img /= sepia_img.max()
  return sepia_img

iface = gr.Interface(sepia, gr.inputs.Image(shape=(200, 200)), "image")

iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://15052.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```



3s completed at 5:05 PM

00:06:30

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

[15]

view the api | built with

```
(<Flask 'gradio.networking'>,
 'http://127.0.0.1:7862/',
 'https://44140.gradio.app')
```

```
import gradio as gr
import numpy as np

def sepia(input_img):
  sepia_filter = np.array([[.393, .769, .189],
                           [.349, .686, .168],
                           [.272, .534, .131]])
  sepia_img = input_img.dot(sepia_filter.T)
  sepia_img /= sepia_img.max()
  return sepia_img

iface = gr.Interface(sepia, gr.inputs.Image(shape=(200, 200)), "image")

iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
```

3s completed at 5:05 PM

00:06:39

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[15] (<Flask 'gradio.networking'>,
      'http://127.0.0.1:7862/',
      'https://44140.gradio.app')
```

```
import gradio as gr
import numpy as np

def sepia(input_img):
  sepia_filter = np.array([[.393, .769, .189],
                           [.349, .686, .168],
                           [.272, .534, .131]])
  sepia_img = input_img.dot(sepia_filter.T)
  sepia_img /= sepia_img.max()
  return sepia_img

iface = gr.Interface(sepia, gr.inputs.Image(shape=(200, 200)), "image")

iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://15052.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

3s completed at 5:05 PM

00:07:01

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

+ Code + Text

RAM Disk Editing

view the api | built with

```
(<Flask 'gradio.networking'>,
 'http://127.0.0.1:7863/',
 'https://15052.gradio.app')
```

3s completed at 5:05 PM

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

+ Code + Text

RAM Disk Editing

```
(<Flask 'gradio.networking'>,
 'http://127.0.0.1:7863/',
 'https://15052.gradio.app')
```

[ ]

3s completed at 5:05 PM

Working with Data

You can use Gradio to support inputs and outputs from your typical data libraries, such as numpy arrays, pandas dataframes, and plotly graphs. Take a look at the demo below (ignore the complicated data manipulation in the function!)

```
import gradio as gr
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def sales_projections(employee_data):
    sales_data = employee_data.iloc[:, 1:4].astype("int").to_numpy()
    regression_values = np.apply_along_axis(lambda row:
        np.array(np.polyld(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 14, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
    ],
```

00:07:09

```
    regression_values = np.apply_along_axis(lambda row:
        np.array(np.poly1d(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 14, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
)
iface.launch()
```

Enter sales figures for employees to predict sales trajectory over year.

EMPLOYEE DATA

	Name	Jan Sales	Feb Sales	Mar Sales
1	Jon	12	14	18

Screenshot

Flag

00:07:10

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Editing

```
[16] ('http://127.0.0.1:7863/',
      'https://15052.gradio.app')
```

3s completed at 5:05 PM

00:07:13

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

[16]

```
(<Flask 'gradio.networking'>,
 'http://127.0.0.1:7863/',
 'https://15052.gradio.app')
```

```
import gradio as gr
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def sales_projections(employee_data):
    sales_data = employee_data.iloc[:, 1:4].astype("int").to_numpy()
    regression_values = np.apply_along_axis(lambda row:
        np.array(np.poly1d(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values
```

3s completed at 5:05 PM

00:07:21

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share A

+ Code + Text

RAM Disk Editing

```
import gradio as gr
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

def sales_projections(employee_data):
    sales_data = employee_data.iloc[:, 1:4].astype("int").to_numpy()
    regression_values = np.apply_along_axis(lambda row:
        np.array(np.polyld(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 14, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
```

3s completed at 5:05 PM

00:07:30

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk

Editing

```
        np.array(np.polyld(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 14, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
)
iface.launch()
```

3s completed at 5:05 PM

00:07:32

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help Saving...

Comment Share

+ Code + Text

RAM Disk Editing

```
    regression_values = np.apply_along_axis(lambda row:
        np.array(np.polyld(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 14, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
)
```

3s completed at 5:05 PM

00:07:35

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
def sales_projections(employee_data):
    sales_data = employee_data.iloc[:, 1:4].astype("int").to_numpy()
    regression_values = np.apply_along_axis(lambda row:
        np.array(np.polyld(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 14, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
    ],
```

3s completed at 5:05 PM

00:07:43

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
def sales_projections(employee_data):
    sales_data = employee_data.iloc[:, 1:4].astype("int").to_numpy()
    regression_values = np.apply_along_axis(lambda row:
        np.array(np.poly1d(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 14, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
```

3s completed at 5:05 PM

00:07:45

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
        np.array(np.polyld(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 14, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
)
iface.launch()
```

3s completed at 5:05 PM

00:07:59

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 14, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
)
iface.launch()
```

3s completed at 5:05 PM

00:08:03

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 14, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
)
iface.launch()
```

3s completed at 5:05 PM

00:08:05

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk

Editing

```
        [
            "dataframe",
            "plot",
            "numpy"
        ],
        description="Enter sales figures for employees to predict sales trajectory over year."
    )
iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
```

Executing (2s) Cell > launch() > setup_tunnel() > create_tunnel() > connect() > start_client() > wait() > wait()

00:08:08

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share A

+ Code + Text

RAM Disk Editing

```
Running on public URL: https://31333.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

3s completed at 5:09 PM

00:08:14

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

```
    description="Enter sales figures for employees to predict sales trajectory over year."
)
iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://31333.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

Enter sales figures for employees to predict sales trajectory over year.

EMPLOYEE DATA

	Name	Jan Sal	Feb Sa	Mar Sa
1	Jon	12	14	18
2	Alice	14	17	2
3	Sana	8	9.5	12

Clear Submit

OUTPUT 1 0.04s

3s completed at 5:09 PM

00:08:17

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

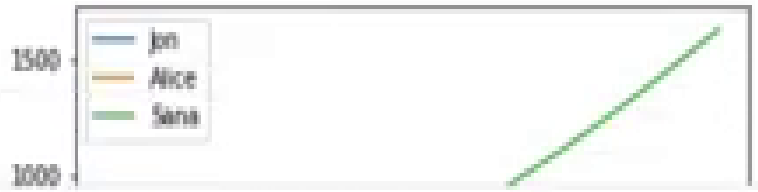
```
    description="Enter sales figures for employees to predict sales trajectory over year."
)
iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://31333.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

OUTPUT 1 0.04s

Name	Jan Sales	Feb Sales	Mar Sales
Jon	12	14	18
Alice	14	17	2
Sana	8	9.5	12

OUTPUT 2



3s completed at 5:09 PM

00:08:26

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help Saving...

Comment Share

+ Code + Text

RAM Disk Editing

```
    sales_data = employee_data.iloc[:, 1:4].astype("int").to_numpy()
    regression_values = np.apply_along_axis(lambda row:
        np.array(np.polyld(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 14, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
```

3s completed at 5:09 PM

00:08:29

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Editing

```
)
iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://31333.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

Clear

Submit

OUTPUT 1

0.04s

Name	Jan Sales	Feb Sales	Mar Sales
Jon	12	14	18
Alice	14	17	2
Sana	8	9.5	12

OUTPUT 2

3s completed at 5:09 PM

00:08:30

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

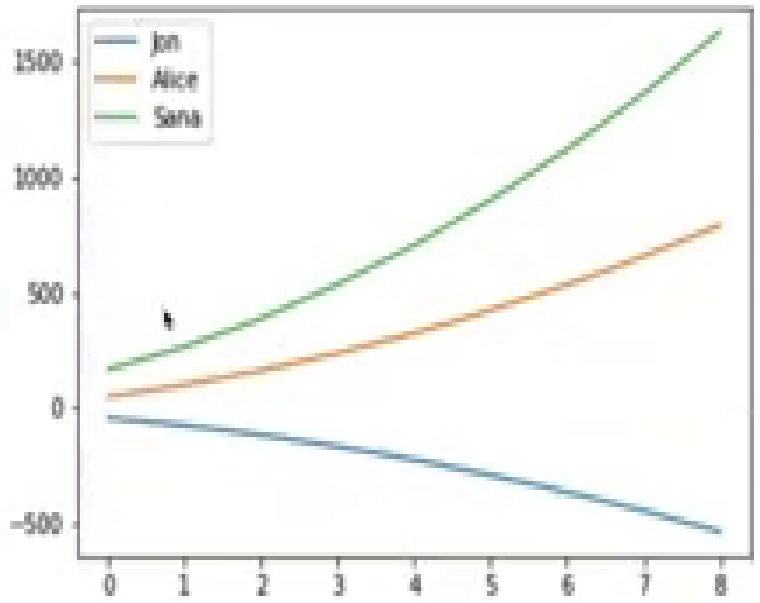
Comment Share

+ Code + Text

RAM Disk Editing

```
)
iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://31333.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```



OUTPUT 3

3s completed at 5:09 PM

00:08:34

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

+ Code + Text

RAM Disk Editing

```
This share link will expire in 72 hours. To get longer links, send an email to: support@g
```

Enter sales figures for employees to predict sales trajectory over year.

EMPLOYEE DATA

	Name	Jan Sal	Feb Sa	Mar Sa
1	Jon	12	14	18
2	Alice	14	17	2
3	Sana	8	9.5	12

Clear Submit

OUTPUT 1 0.04s

Name	Jan Sales	Feb Sales	Mar Sales
Jon	12	14	18
Alice	14	17	2
Sana	8	9.5	12

OUTPUT 2

3s completed at 5:09 PM

00:08:37

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

```
def sales_projections(employee_data):
    sales_data = employee_data.iloc[:, 1:4].astype("int").to_numpy()
    regression_values = np.apply_along_axis(lambda row:
        np.array(np.poly1d(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return employee_data, plt.gcf(), regression_values

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 100, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
    ]
```

3s completed at 5:09 PM

00:08:39

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share A

+ Code + Text

RAM Disk

Editing

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
```

...

Executing (0s) C... › laun... › version.ch... › g... › reque... › reque... › se... › se... › urlop... › make_req... › getrespo... › beg... › read_sta... › readin... › recv_i... › rea...

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share A

+ Code + Text

RAM Disk Editing

```
iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 100, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
)
iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://50350.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

Enter sales figures for employees to predict sales trajectory over year.

EMPLOYEE DATA

	Name	Jan Sal	Feb Sa	Mar Sa

3s completed at 5:09 PM

00:08:47

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

```
        "numpy"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
)
iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://50350.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

Enter sales figures for employees to predict sales trajectory over year.

EMPLOYEE DATA

	Name	Jan Sal	Feb Sa	Mar Sa
1	Jon	12	100	18
2	Alice	14	17	2
3	Sana	8	9.5	12

Screenshot Flag

Clear Submit

3s completed at 5:09 PM

00:08:51

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://50350.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

Enter sales figures for employees to predict sales trajectory over year.

EMPLOYEE DATA

	Name	Jan Sal	Feb Sa	Mar Sa
1	Jon	12	100	18
2	Alice	14	17	2
3	Sana	8	9.5	12

Clear Submit

OUTPUT 1 0.02s

Name	Jan Sales	Feb Sales	Mar Sales
Jon	12	100	18

3s completed at 5:09 PM

00:08:53

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

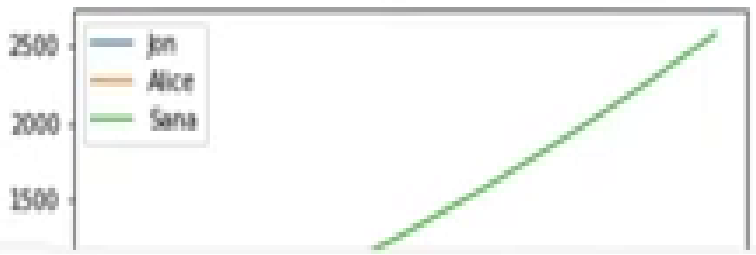
```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://50350.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

Clear Submit

OUTPUT 1 0.02s

Name	Jan Sales	Feb Sales	Mar Sales
Jon	12	100	18
Alice	14	17	2
Sana	8	9.5	12

OUTPUT 2



3s completed at 5:09 PM

00:09:05

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk

Editing

```
    regression_values = np.apply_along_axis(lambda row:
        np.array(np.poly1d(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return plt.gcf()

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 100, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [
        "dataframe",
        "plot",
        "numpy"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
)
```

3s completed at 5:09 PM

00:09:09

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

```
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return plt.gcf()

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 100, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [

        "plot",
        "numpy"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
)
iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://50350.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

3s completed at 5:09 PM

00:09:13

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share A

+ Code + Text

RAM Disk Editing

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
```

...

Executing (1s) C... > laun... > setup tu... > url requ... > urlop... > op... > op... > call ch... > https o... > do_o... > getrespo... > be... > read_st... > readi... > recv_i... > rea...

00:09:16

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share A

+ Code + Text

RAM Disk Editing

Enter sales figures for employees to predict sales trajectory over year.

EMPLOYEE DATA

	Name	Jan Sal	Feb Sa	Mar Sa
1	Jon	12	100	18
2	Alice	14	17	2
3	Sana	8	9.5	12

Screenshot

Flag

Clear

Submit

3s completed at 5:10 PM

00:09:18

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

```
            "plot"
        ],
        description="Enter sales figures for employees to predict sales trajectory over year."
    )
    iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://22580.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

Enter sales figures for employees to predict sales trajectory over year.

EMPLOYEE DATA

	Name	Jan Sal	Feb Sa	Mar Sa
1	Jon	12	100	18
2	Alice	14	17	2
3	Sana	8	9.5	12

0.0s

Screenshot Flag

Clear Submit

3s completed at 5:10 PM

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

+ Code + Text

RAM Disk Editing

```
This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

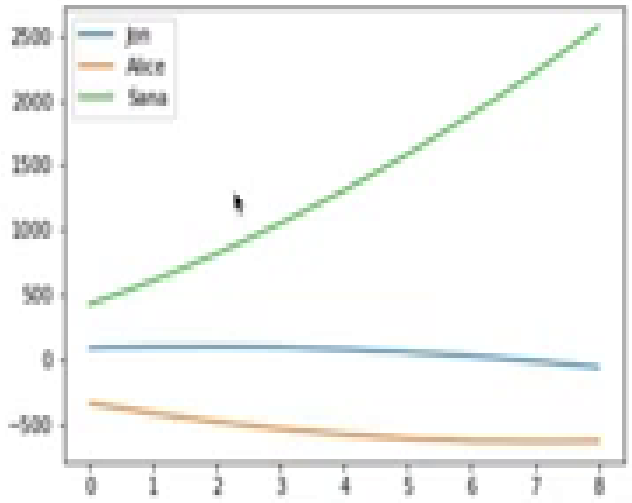
Enter sales figures for employees to predict sales trajectory over year.

EMPLOYEE DATA

	Name	Jan Sal	Feb Sa	Mar Sa
1	Jon	12	100	18
2	Alice	14	17	2
3	Sana	8	9.5	12

Clear Submit

OUTPUT 0.02s



3s completed at 5:10 PM

00:09:52

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Editing

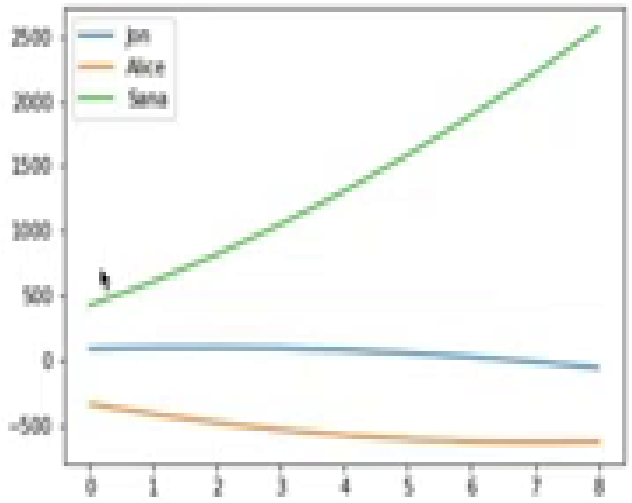
Enter sales figures for employees to predict sales trajectory over year.

EMPLOYEE DATA

	Name	Jan Sal	Feb Sa	Mar Sa
1	Jon	12	100	18
2	Alice	14	17	2
3	Sana	8	9.5	12

Clear Submit

OUTPUT 0.02s



Screenshot Flag

3s completed at 5:10 PM

00:09:58

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
            np.arange(3,12), 0), len(sales_data), axis=0)
        projected_values = np.array([
            month * month * regression[0] + month * regression[1] + regression[2]
            for month, regression in zip(projected_months, regression_values)])
        plt.plot(projected_values.T)
        plt.legend(employee_data["Name"])
        return plt.gcf()

    iface = gr.Interface(sales_projections,
        gr.inputs.Dataframe(
            headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
            default=[["Jon", 12, 100, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
        ),
        [

            "plot"
        ],
        description="Enter sales figures for employees to predict sales trajectory over year."
    )
    iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://22580.gradio.app
```

3s completed at 5:10 PM

00:10:27

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return plt.gcf()

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 100, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [

        "plot"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
)
iface.launch()
```

```
Colab notebook detected. To show errors in colab notebook, set `debug=True` in `launch()`
Running on public URL: https://22580.gradio.app

This share link will expire in 72 hours. To get longer links, send an email to: support@gradio.app
```

3s completed at 5:10 PM

00:10:55

Building your first Gradio Web app

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
    regression_values = np.apply_along_axis(lambda row:
        np.array(np.polyld(np.polyfit([0,1,2], row, 2))), 0, sales_data)
    projected_months = np.repeat(np.expand_dims(
        np.arange(3,12), 0), len(sales_data), axis=0)
    projected_values = np.array([
        month * month * regression[0] + month * regression[1] + regression[2]
        for month, regression in zip(projected_months, regression_values)])
    plt.plot(projected_values.T)
    plt.legend(employee_data["Name"])
    return plt.gcf()

iface = gr.Interface(sales_projections,
    gr.inputs.Dataframe(
        headers=["Name", "Jan Sales", "Feb Sales", "Mar Sales"],
        default=[["Jon", 12, 100, 18], ["Alice", 14, 17, 2], ["Sana", 8, 9.5, 12]]
    ),
    [

        "plot"
    ],
    description="Enter sales figures for employees to predict sales trajectory over year."
)
iface.launch()
```

3s completed at 5:10 PM