How to Build a GUI Using Gradio for Machine Learning Models

Gradio is a GUI library that allows you to create customizable GUI components for your Machine Learning model.

```
!pip install gradio
```

```
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3->gradio) (3.15.4
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3->gradio) (2.31.0
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3->gradio) (4.
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (1.2.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (4.53.
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (1.4.5
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (2.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas<3.0,>=1.0->gradio) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas<3.0,>=1.0->gradio) (2024.1)
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic>=2.0->gradio) (0.
Requirement already satisfied: pydantic-core==2.20.0 in /usr/local/lib/python3.10/dist-packages (from pydantic>=2.0->gradio) (2.2
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0,>=0.12->gradio) (8.1.7)
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0,>=0.12->gradio) (1.5
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.10/dist-packages (from typer<1.0,>=0.12->gradio) (13.7.1)
Collecting starlette<0.38.0,>=0.37.2 (from fastapi->gradio)
  Downloading starlette-0.37.2-py3-none-any.whl (71 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 71.9/71.9 kB 8.6 MB/s eta 0:00:00
Collecting fastapi-cli>=0.0.2 (from fastapi->gradio)
  Downloading fastapi_cli-0.0.4-py3-none-any.whl (9.5 kB)
Collecting ujson!=4.0.2,!=4.1.0,!=4.2.0,!=4.3.0,!=5.0.0,!=5.1.0,>=4.0.1 (from fastapi->gradio)
  Downloading ujson-5.10.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (53 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 53.6/53.6 kB 6.3 MB/s eta 0:00:00
Collecting email_validator>=2.0.0 (from fastapi->gradio)
  Downloading email_validator-2.2.0-py3-none-any.whl (33 kB)
Collecting dnspython>=2.0.0 (from email_validator>=2.0.0->fastapi->gradio)
  Downloading dnspython-2.6.1-py3-none-any.whl (307 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 307.7/307.7 kB 28.6 MB/s eta 0:00:00
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6.0,>=4.2.0
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6.0,>
Requirement already satisfied: rpds-py>=0.7.1 in /usr/local/lib/python3.10/dist-packages (from jsonschema>=3.0->altair<6.0,>=4.2.
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib~=3.0->g
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.11.0->typer<1.0,>=
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.11.0->typer<1.0,
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio->httpx>=0.24.1->gradio) (1.2
Collecting httptools>=0.5.0 (from uvicorn>=0.14.0->gradio)
  Downloading httptools-0.6.1-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 341.4/341.4 kB 37.0 MB/s eta 0:00:00
Collecting python-dotenv>=0.13 (from uvicorn>=0.14.0->gradio)
  Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Collecting uvloop!=0.15.0,!=0.15.1,>=0.14.0 (from uvicorn>=0.14.0->gradio)
  Downloading uvloop-0.19.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.4/3.4 MB 33.1 MB/s eta 0:00:00
Collecting watchfiles>=0.13 (from uvicorn>=0.14.0->gradio)
  Downloading watchfiles-0.22.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 41.2 MB/s eta 0:00:00
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hu
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->
Building wheels for collected packages: ffmpy
  Building wheel for ffmpy (setup.py) ... done
  Created wheel for ffmpy: filename=ffmpy-0.3.2-py3-none-any.whl size=5584 sha256=078c1c622585537dc218c6760322b861f1e06e36c0bba09
  Stored in directory: /root/.cache/pip/wheels/bd/65/9a/671fc6dcde07d4418df0c592f8df512b26d7a0029c2a23dd81
Successfully built ffmpy
Installing collected packages: pydub, ffmpy, websockets, uvloop, ujson, tomlkit, semantic-version, ruff, python-multipart, python
Successfully installed aiofiles-23.2.1 dnspython-2.6.1 email_validator-2.2.0 fastapi-0.111.0 fastapi-cli-0.0.4 ffmpy-0.3.2 gradio
```

Install the required packages

```
import numpy as np

import pandas as pd

import gradio as gr

import warnings

warnings.filterwarnings('ignore')
```

Import the data

```
data = pd.read_csv('/content/diabetes.csv')
```

```
data.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigree
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

Next steps: Generate code with data View recommended plots

```
print (data.columns)
```

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

we get our X and Y variables

```
x = data.drop(['Outcome'], axis=1)

y = data['Outcome']
```

Split the data

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x,y)
```

Scale our data

```
from sklearn.preprocessing import StandardScaler

#instantiate StandardScaler object
scaler = StandardScaler()

#scale data
x_train_scaled = scaler.fit_transform(x_train)

x_test_scaled = scaler.fit_transform(x_test)
```

train the model

```
#import model object
from sklearn.neural_network import MLPClassifier
model =  MLPClassifier(max_iter=1000,  alpha=1)

#train model on training data
model.fit(x_train_scaled, y_train)

#getting model performance on test data
print("accuracy:", model.score(x_test_scaled, y_test))
```

```
accuracy: 0.75
```

Create the function for Gradio

Here we are going to create a function that will take in the features of the data set which our model was trained on and pass it as an array to our model to predict. Then we are going to build our Gradio web app based on that function.

Gradio builds GUI components for our Machine Learning model based on the function. The function provides a way for Gradio to get input from users and pass it on to the ML model, which will then process it and then pass it back to Gradio which then passes the result out.

First, we will get the feature columns which we will then pass onto our function.

```
#geting our columns

print(data.columns)
```

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

```
def diabetes(Pregnancies, Glucose, Blood_Pressure, SkinThickness, Insulin, BMI, Diabetes_Pedigree, Age):
  x = np.array([Pregnancies,Glucose,Blood_Pressure,SkinThickness,Insulin,BMI,Diabetes_Pedigree,Age])
  prediction = model.predict(x.reshape(1, -1))

  return prediction
```

In the above code, we passed the feature columns from our data as arguments into a function which we named diabetes. Then we turned the arguments into a NumPy array which we then passed onto our model for prediction. Finally we returned the predicted result of our model.

Create our Gradio Interface

```
!pip install gradio
import gradio as gr

# Use gr.Textbox() directly instead of gr.outputs.Textbox()
outputs = gr.Textbox()

app = gr.Interface(fn=diabetes, inputs=['number','number','number','number','number','number','number','number'],
                   outputs=outputs,description="This is a diabetes model")
```

```
Requirement already satisfied: pydantic>=2.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (2.8.0)
Requirement already satisfied: pydub in /usr/local/lib/python3.10/dist-packages (from gradio) (0.25.1)
Requirement already satisfied: python-multipart>=0.0.9 in /usr/local/lib/python3.10/dist-packages (from gradio) (0.0.9)
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (6.0.1)
Requirement already satisfied: ruff>=0.2.2 in /usr/local/lib/python3.10/dist-packages (from gradio) (0.5.1)
Requirement already satisfied: semantic-version~=2.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (2.10.0)
Requirement already satisfied: tomlkit==0.12.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (0.12.0)
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.10/dist-packages (from gradio) (0.12.3)
Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (4.12.2)
Requirement already satisfied: urllib3~=2.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (2.0.7)
Requirement already satisfied: uvicorn>=0.14.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (0.30.1)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from gradio-client==1.0.2->gradio) (2023.6.0)
Requirement already satisfied: websockets<12.0,>=10.0 in /usr/local/lib/python3.10/dist-packages (from gradio-client==1.0.2->grad
Requirement already satisfied: entrypoints in /usr/local/lib/python3.10/dist-packages (from altair<6.0,>=4.2.0->gradio) (0.4)
Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.10/dist-packages (from altair<6.0,>=4.2.0->gradio) (4.19
Requirement already satisfied: toolz in /usr/local/lib/python3.10/dist-packages (from altair<6.0,>=4.2.0->gradio) (0.12.1)
Requirement already satisfied: anyio in /usr/local/lib/python3.10/dist-packages (from httpx>=0.24.1->gradio) (3.7.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from httpx>=0.24.1->gradio) (2024.6.2)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.10/dist-packages (from httpx>=0.24.1->gradio) (1.0.5)
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages (from httpx>=0.24.1->gradio) (3.7)
Requirement already satisfied: sniffio in /usr/local/lib/python3.10/dist-packages (from httpx>=0.24.1->gradio) (1.3.1)
Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.10/dist-packages (from httpcore==1.*->httpx>=0.24.1->gra
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3->gradio) (3.15.4
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3->gradio) (2.31.0
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.19.3->gradio) (4.
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (1.2.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (4.53.
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib~=3.0->gradio) (1.4.5
```

```
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib~=3.0->g
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.11.0->typer<1.0,>=
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.11.0->typer<1.0,
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio->httpx>=0.24.1->gradio) (1.2
Requirement already satisfied: httptools>=0.5.0 in /usr/local/lib/python3.10/dist-packages (from uvicorn>=0.14.0->gradio) (0.6.1)
Requirement already satisfied: python-dotenv>=0.13 in /usr/local/lib/python3.10/dist-packages (from uvicorn>=0.14.0->gradio) (1.0
Requirement already satisfied: uvloop!=0.15.0,!=0.15.1,>=0.14.0 in /usr/local/lib/python3.10/dist-packages (from uvicorn>=0.14.0-
Requirement already satisfied: watchfiles>=0.13 in /usr/local/lib/python3.10/dist-packages (from uvicorn>=0.14.0->gradio) (0.22.0
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hu
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->
```

The first thing we did above was to create a variable named outputs which holds the GUI component for our model result. The result of our model's prediction will be outputted in a text box.

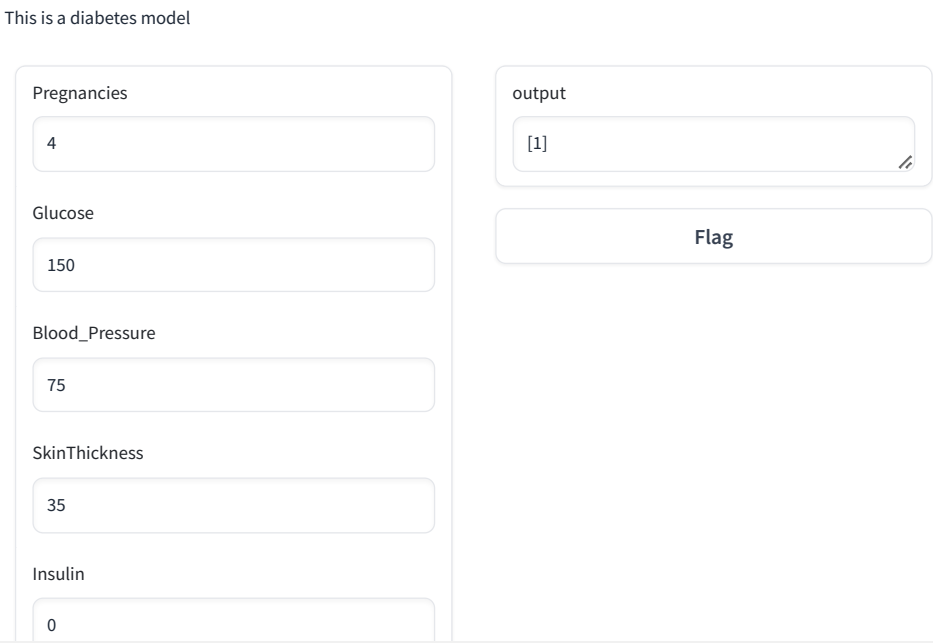
The inputs represent the feature columns that are present in our dataset – the same 8 feature column names we passed into our diabetes function.

```
app.launch()
```

```
Setting queue=True in a Colab notebook requires sharing enabled. Setting `share=True` (you can turn this off by setting `share=False

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
Running on public URL: https://d57d11b20d59277f39.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from Terminal to deploy to Spa
```




```
#To provide a shareable link
app.launch(share=True)
```

```
Rerunning server... use `close()` to stop if you need to change `launch()` parameters
----
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
Running on public URL: https://d57d11b20d59277f39.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run
```

This is a diabetes model

Pregnancies

0

Glucose

0

Blood_Pressure

0

SkinThickness

0

Insulin