```
!pip install gradio
```

```
Collecting gradio
  Downloading gradio-4.26.0-py3-none-any.whl (17.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.1/17.1 MB 43.4 MB/s eta 0:00:00
Collecting aiofiles<24.0,>=22.0 (from gradio)
  Downloading aiofiles-23.2.1-py3-none-any.whl (15 kB)
Requirement already satisfied: altair<6.0,>=4.2.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (4.2.2)
Collecting fastapi (from gradio)
  Downloading fastapi-0.110.1-py3-none-any.whl (91 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 91.9/91.9 kB 9.5 MB/s eta 0:00:00
Collecting ffmpy (from gradio)
  Downloading ffmpy-0.3.2.tar.gz (5.5 kB)
  Preparing metadata (setup.py) ... done
Collecting gradio-client==0.15.1 (from gradio)
  Downloading gradio_client-0.15.1-py3-none-any.whl (313 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 313.6/313.6 kB 22.4 MB/s eta 0:00:00
Collecting httpx>=0.24.1 (from gradio)
  Downloading httpx-0.27.0-py3-none-any.whl (75 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 75.6/75.6 kB 7.4 MB/s eta 0:00:00
Requirement already satisfied: huggingface-hub>=0.19.3 in /usr/local/lib/python3.10/dist-packages (from gradio) (0.20.3)
Requirement already satisfied: importlib-resources<7.0,>=1.3 in /usr/local/lib/python3.10/dist-packages (from gradio) (6.4.0)
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (3.1.3)
Requirement already satisfied: markupsafe~=2.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (2.1.5)
Requirement already satisfied: matplotlib~=3.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (3.7.1)
Requirement already satisfied: numpy~=1.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (1.25.2)
Collecting orjson~=3.0 (from gradio)
  Downloading orjson-3.10.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (144 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 144.8/144.8 kB 14.2 MB/s eta 0:00:00
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from gradio) (24.0)
Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (2.0.3)
Requirement already satisfied: pillow<11.0,>=8.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (9.4.0)
Requirement already satisfied: pydantic>=2.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (2.6.4)
Collecting pydub (from gradio)
  Downloading pydub-0.25.1-py2.py3-none-any.whl (32 kB)
Collecting python-multipart>=0.0.9 (from gradio)
  Downloading python_multipart-0.0.9-py3-none-any.whl (22 kB)
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (6.0.1)
Collecting ruff>=0.2.2 (from gradio)
  Downloading ruff-0.3.7-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (8.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.9/8.9 MB 45.7 MB/s eta 0:00:00
Collecting semantic-version~=2.0 (from gradio)
  Downloading semantic_version-2.10.0-py2.py3-none-any.whl (15 kB)
Collecting tomlkit==0.12.0 (from gradio)
  Downloading tomlkit-0.12.0-py3-none-any.whl (37 kB)
Requirement already satisfied: typer[all]<1.0,>=0.9 in /usr/local/lib/python3.10/dist-packages (from gradio) (0.9.4)
Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/python3.10/dist-packages (from gradio) (4.11.0)
Collecting uvicorn>=0.14.0 (from gradio)
  Downloading uvicorn-0.29.0-py3-none-any.whl (60 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.8/60.8 kB 6.3 MB/s eta 0:00:00
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from gradio-client==0.15.1->gradio) (2023.6.0)
Collecting websockets<12.0,>=10.0 (from gradio-client==0.15.1->gradio)
  Downloading websockets-11.0.3-cp310-cp310-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (129 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 129.9/129.9 kB 13.7 MB/s eta 0:00:00
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) (4.19.2)
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) (2024.2.2)
```

```
import gradio as gr#importing the installed library
```

```
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report,accuracy_score
```

```
df=pd.read_csv("/content/Iris.csv")
```

```
df.head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Next steps: Generate code with df View recommended plots

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   SepalLengthCm  150 non-null    float64
 1   SepalWidthCm   150 non-null    float64
 2   PetalLengthCm  150 non-null    float64
 3   PetalWidthCm   150 non-null    float64
 4   Species        150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

```
df.isnull().sum()
```

```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

```
df.describe()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

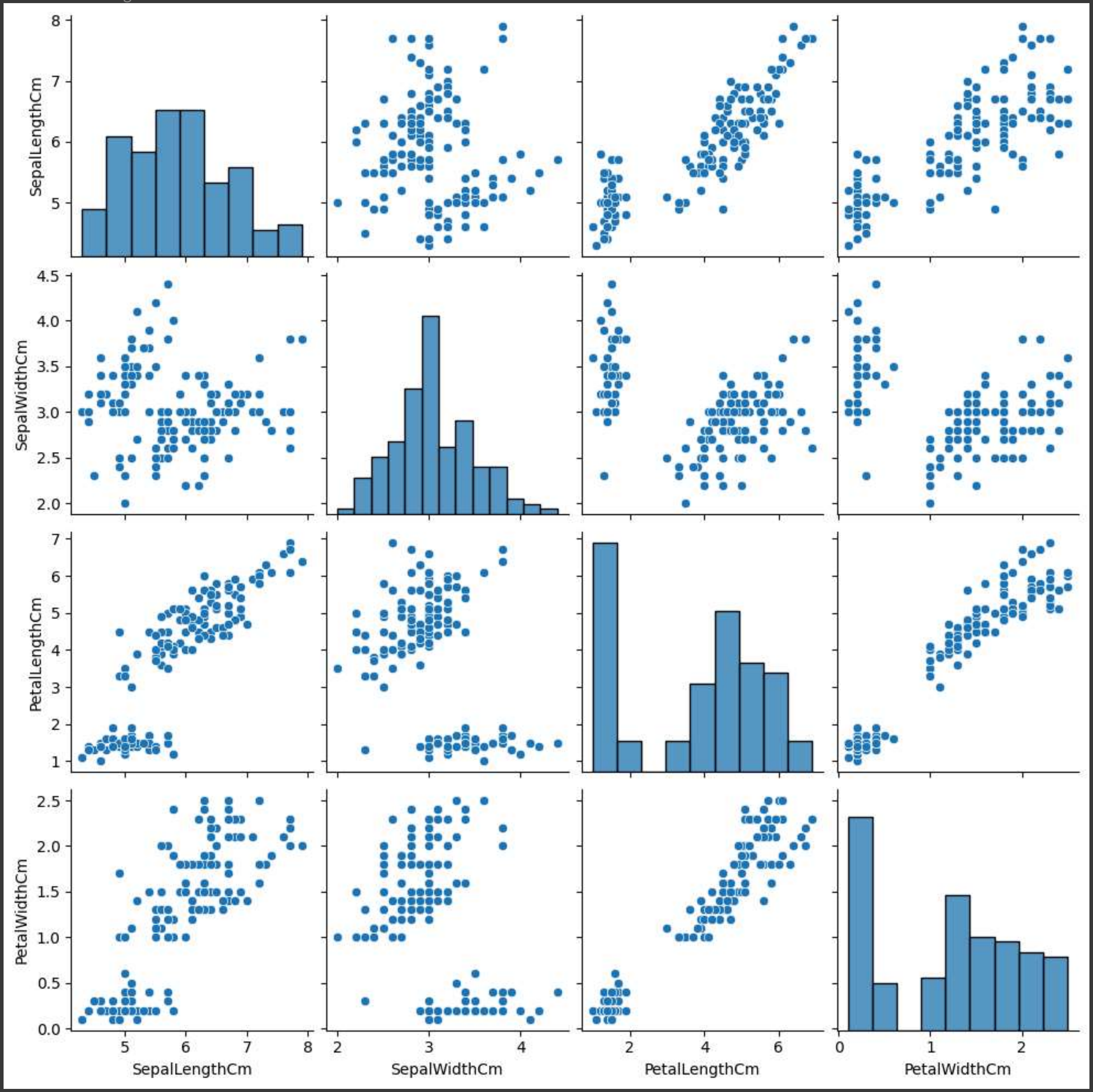
```
df['Species'].unique()#checking the class labels
```

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

Visualization

```
sns.pairplot(df)
```

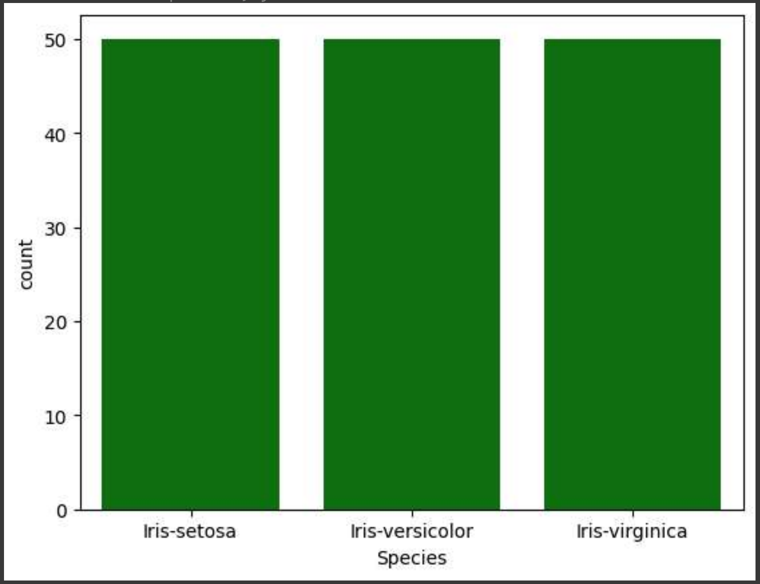
```
<seaborn.axisgrid.PairGrid at 0x7a2affc76da0>
```



Count Plot To check Class Im-Balence Problem

```
sns.countplot(x=df['Species'],color="green")
```

```
<Axes: xlabel='Species', ylabel='count'>
```



Feature Engineering

```
X=df.drop(['Species'],axis=1)
Y=df['Species']
```

Train_Test Splitting of data

```
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.3,random_state=20)
```

Model Object Creation

```
model=GaussianNB()
```

Model Training

```
model.fit(x_train,y_train)
```

```
▾ GaussianNB
GaussianNB()
```

Model Predictions

```
y_pred=model.predict(x_test)
```

Model Evaluation

```
print("Classification Report : \n",classification_report(y_test,y_pred))
print("Accuracy Score Report : \n",accuracy_score(y_pred,y_test))
```

```
Classification Report :
                  precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        13
Iris-versicolor       0.82      1.00      0.90        18
 Iris-virginica       1.00      0.71      0.83        14

       accuracy                           0.91        45
      macro avg       0.94      0.90      0.91        45
   weighted avg       0.93      0.91      0.91        45

Accuracy Score Report :
 0.9111111111111111
```

Function creation for gradio user interface

```
def fun1(s_l,s_w,p_l,p_w):

  data = {
      "SepalLengthCm": [s_l],
      "SepalWidthCm": [s_w],
      "PetalLengthCm":[p_l],
      "PetalWidthCm":[p_w]}

  df_sample=pd.DataFrame(data)
  res= model.predict(df_sample)

  return res
```

Sample Checking

```
fun1(5.1,3.5,1.4,0.2)
```

```
array(['Iris-setosa'], dtype='<U15')
```

Interface creation by using Gradio library

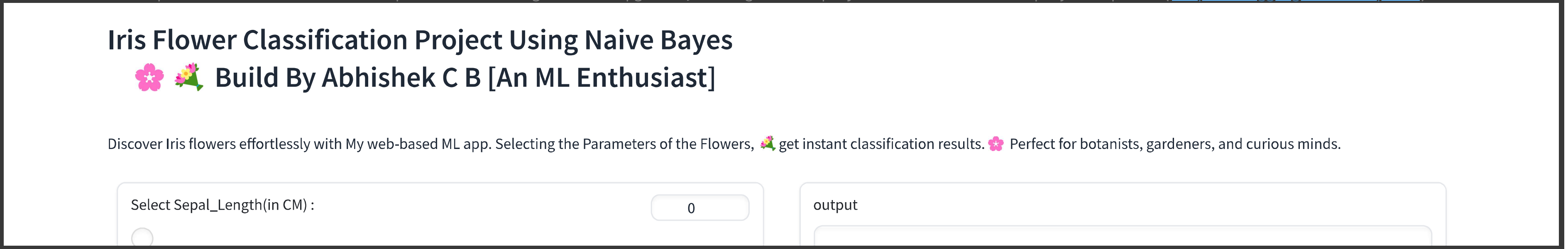
```
app=gr.Interface(fn=fun1,
                 inputs=[gr.Slider(0,10,label="Select Sepal_Length(in CM) :")
                 ,gr.Slider(0,10,label="Select Sepal_Width (in CM) :"),
                 gr.Slider(0,10,label="Select Petal_Length (in CM) :"),
```

Launching the interface

```
app.launch(share=True)
```

```
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
Running on public URL: https://6d803386506e32a9e0.gradio.live

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from Terminal to deploy to Spaces (https://huggingface.co/spaces)
```



Iris Flower Classification Project Using Naive Bayes
🌸 💐 Build By Abhishek C B [An ML Enthusiast]

Discover Iris flowers effortlessly with My web-based ML app. Selecting the Parameters of the Flowers, 💐 get instant classification results. 🌸 Perfect for botanists, gardeners, and curious minds.