```
import io
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, f1_score, classification_report, confusion_matrix
from google.colab import files
```

```
print("📁 Upload your diabetes CSV file")
uploaded = files.upload()

# Auto-detect the uploaded filename
filename = list(uploaded.keys())[0]
df = pd.read_csv(io.BytesIO(uploaded[filename]), header=None)
```

```
📁 Upload your diabetes CSV file
Choose Files pima-indian...tes.data.csv
pima-indians-diabetes.data.csv(text/csv) - 23279 bytes, last modified: 11/8/2025 - 100% done
Saving pima-indians-diabetes.data.csv to pima-indians-diabetes.data (2).csv
```

```
df.columns = ['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin',
              'BMI','DiabetesPedigreeFunction','Age','Outcome']
```

```
X = df.drop('Outcome', axis=1)
y = df['Outcome']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
model = GaussianNB().fit(X_train, y_train)
y_pred = model.predict(X_test)
```

```
acc = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
print(f"\n✅ Accuracy: {acc:.4f}")
print(f"✅ F1-Score: {f1:.4f}\n")
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
✅ Accuracy: 0.7344
✅ F1-Score: 0.6483

Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.76      0.79       123
           1       0.62      0.68      0.65        69

    accuracy                           0.73       192
   macro avg       0.71      0.72      0.72       192
weighted avg       0.74      0.73      0.74       192
```

```
print(f"\nAccuracy: {accuracy_score(y_test, y_pred):.4f}")
print(f"F1-Score: {f1_score(y_test, y_pred):.4f}")
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

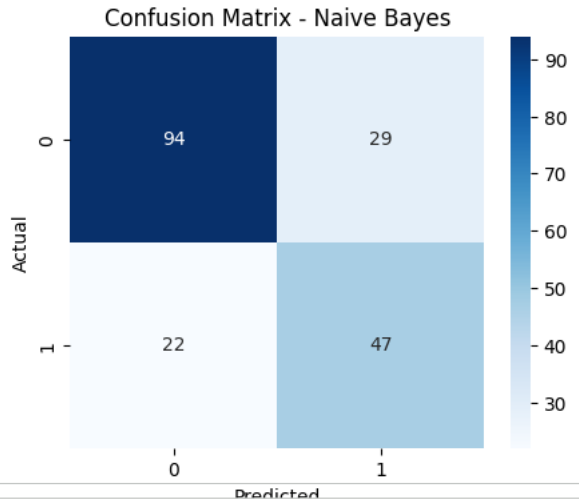
```
Accuracy: 0.7344
F1-Score: 0.6483

Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.76      0.79       123
           1       0.62      0.68      0.65        69

    accuracy                           0.73       192
   macro avg       0.71      0.72      0.72       192
weighted avg       0.74      0.73      0.74       192
```

```
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(5,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix - Naive Bayes")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```




```
f1_scores = [f1_score(y_test, y_pred, pos_label=0), f1_score(y_test, y_pred, pos_label=1)]
plt.bar(['Non-Diabetic','Diabetic'], f1_scores, color=['green','red'])
plt.title("F1-Scores for Each Class")
plt.ylabel("F1-Score")
plt.show()
```