```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix, precision_score, recall_score, f1_score
import matplotlib.pyplot as plt
import seaborn as sns

path="/content/drive/MyDrive/Dataset/Admission_prediction.csv"
df=pd.read_csv(path)

# Display first few rows of the dataset
print("Dataset Preview:")
print(df.head())
```

```
Dataset Preview:
   Serial No.  GRE_Score  TOEFL_Score  University Rating  SOP  LOR
CGPA  \
0           1      337.0        118.0                4.0  4.5  4.5
9.65
1           2      324.0        107.0                4.0  4.0  4.5
8.87
2           3        NaN        104.0                3.0  3.0  3.5
8.00
3           4      322.0        110.0                3.0  3.5  2.5
8.67
4           5      314.0        103.0                2.0  2.0  3.0
8.21

   Research  Chance_of_Admit
0         1             0.92
1         1             0.76
2         1             0.72
3         1             0.80
4         0             0.65
```

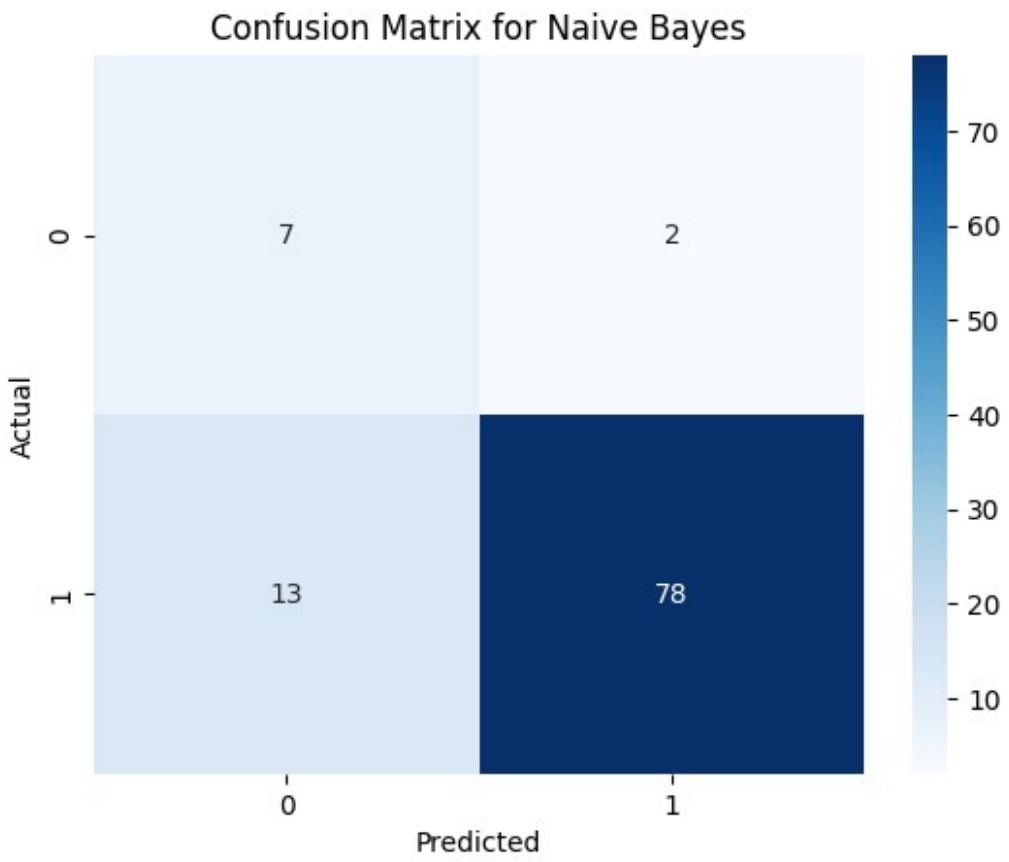
```
# Handle missing values by filling with the median
df.fillna(df.median(), inplace=True)

# Naive Bayes Classification
nb_df = df.copy()
X_nb = nb_df.drop('Chance_of_Admit', axis=1)
y_nb = (nb_df['Chance_of_Admit'] >= 0.5).astype(int)
X_train_nb, X_test_nb, y_train_nb, y_test_nb = train_test_split(X_nb,
y_nb, test_size=0.2, random_state=42)
X_train_nb.shape, X_test_nb.shape, y_train_nb.shape, y_test_nb.shape
```

```
((400, 8), (100, 8), (400,), (100,))
```

```python
nb = GaussianNB()
nb.fit(X_train_nb, y_train_nb)
y_pred_nb = nb.predict(X_test_nb)

# Evaluate Naive Bayes
print("Naive Bayes Classification:")
print(f"Accuracy: {accuracy_score(y_test_nb, y_pred_nb)}")
print(f"Precision: {precision_score(y_test_nb, y_pred_nb)}")
print(f"Recall: {recall_score(y_test_nb, y_pred_nb)}")
print(f"F1 Score: {f1_score(y_test_nb, y_pred_nb)}")
print(classification_report(y_test_nb, y_pred_nb))
```

```
Naive Bayes Classification:
Accuracy: 0.85
Precision: 0.975
Recall: 0.8571428571428571
F1 Score: 0.9122807017543859
              precision    recall  f1-score   support

           0       0.35      0.78      0.48         9
           1       0.97      0.86      0.91        91

    accuracy                           0.85       100
   macro avg       0.66      0.82      0.70       100
weighted avg       0.92      0.85      0.87       100
```

```python
cm_nb = confusion_matrix(y_test_nb, y_pred_nb)
sns.heatmap(cm_nb, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix for Naive Bayes')
plt.show()
```


Confusion Matrix for Naive Bayes
Actual
Predicted
0
1
7
2
13
78
70
60
50
40
30
20
10