BAYESIAN

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.naive_bayes import GaussianNB
```

In [ ]:
```
df=pd.read_csv('Naive-Bayes-Classification-Data.csv',sep=',')
```

In [ ]:
```
df
```

Out[ ]:

	glucose	bloodpressure	diabetes
0	40	85	0
1	40	92	0
2	45	63	1
3	45	80	0
4	40	73	1
...	...	...	...
990	45	87	0
991	40	83	0
992	40	83	0
993	40	60	1
994	45	82	0

995 rows × 3 columns

In [ ]:
```
X=df.drop(["diabetes"],axis=1)
y=df["diabetes"]
```

In [ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
```

In [ ]:
```
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [ ]:
```
nvclassifier = GaussianNB()
nvclassifier.fit(X_train, y_train)
```

Out[ ]:
```
GaussianNB
GaussianNB()
```

In [ ]:
```
y_pred = nvclassifier.predict(X_test)
```

In [ ]:
```
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

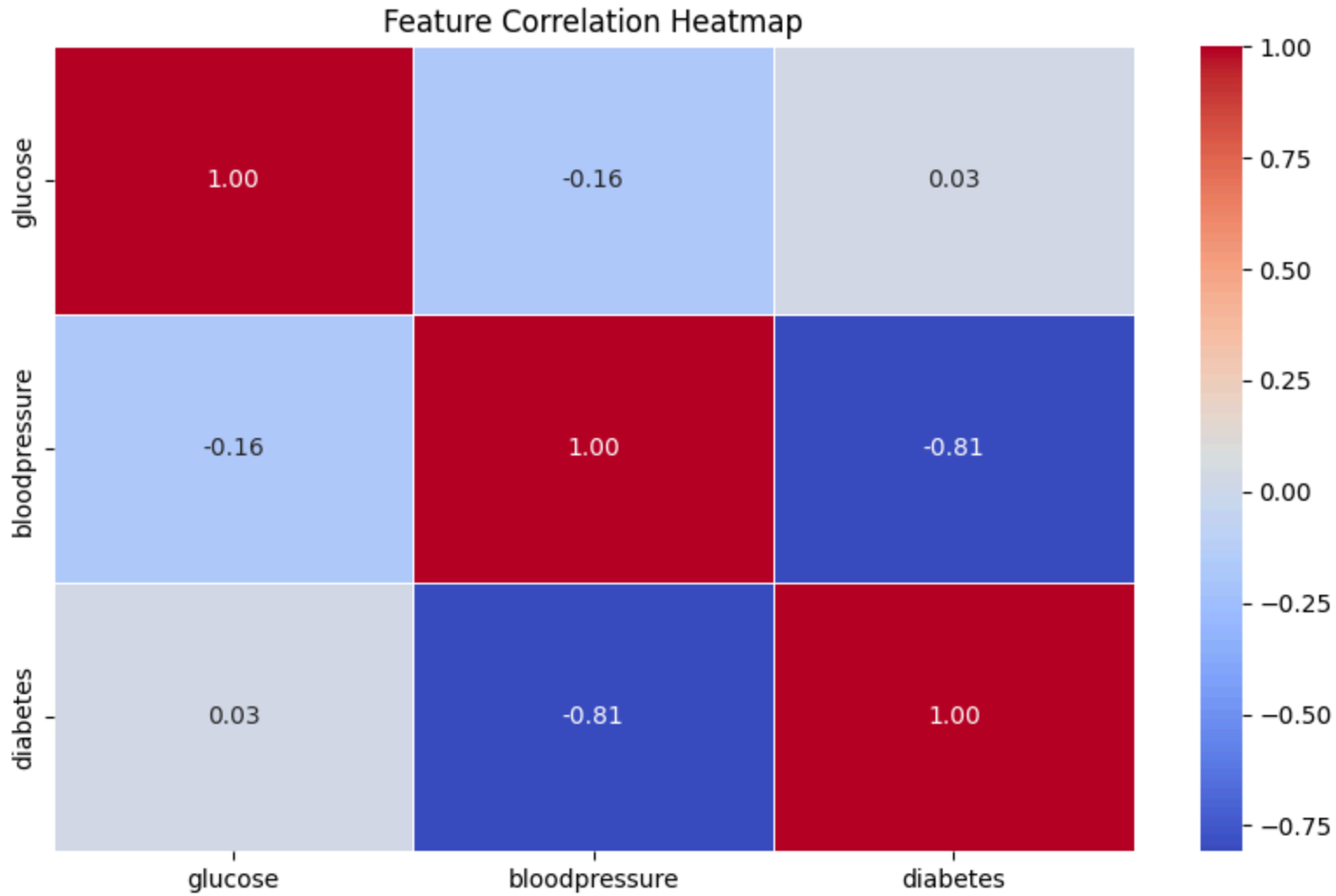
```
Accuracy: 0.95
```

In [ ]:
```
print("Actual values:", y_test[:10].values)
print("Predicted values:", y_pred[:10])
```

```
Actual values: [1 0 1 1 0 1 1 1 0 1]
Predicted values: [1 0 1 1 0 1 1 1 0 1]
```

In [ ]:
```
corr_matrix = df.corr()

plt.figure(figsize=(10, 6))
sns.heatmap(corr_matrix, annot=True, fmt=".2f", cmap="coolwarm", linewidths=0.5)
plt.title("Feature Correlation Heatmap")
plt.show()
```


Feature Correlation Heatmap
glucose
bloodpressure
diabetes
1.00
-0.16
0.03
-0.16
1.00
-0.81
0.03
-0.81
1.00
glucose
bloodpressure
diabetes
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75