JADAVPUR UNIVERSITY

Faculty of Engineering & Technology

Machine Learning **Engg. Laboratory**

Name Ankush Sil Sarma

Class 4th Year 1st Sem **Roll No.** 002011001042

Date of Experiment 27/07/2023 **Date of Submission** 06/08/2023

Marks Obtained ..... **Signature of Examiner** .....

C O – W O R K E R

NAME	**ROLL**

Experiment No .....

Commence at .....

Name of teacher concerned **Completed at** .....

TITLE

OBJECT

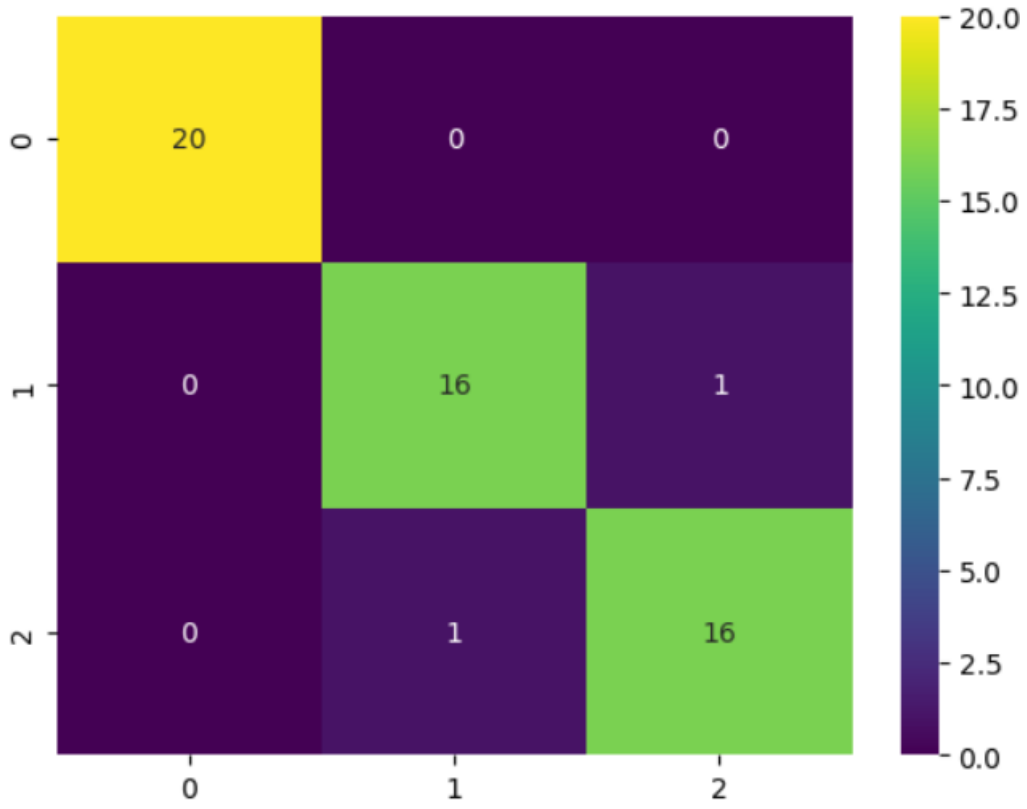
IRIS PLANTS

```
import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

iris = load_iris()
X, y = iris.data, iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.36, random_state=42)

#Gaussian
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)

y_pred_nb = nb_classifier.predict(X_test)

accuracy_nb = accuracy_score(y_test, y_pred_nb)
precision_nb = precision_score(y_test, y_pred_nb, average='weighted')
recall_nb = recall_score(y_test, y_pred_nb, average='weighted')
f1_score_nb = f1_score(y_test, y_pred_nb, average='weighted')
confusion_matrix_nb = confusion_matrix(y_test, y_pred_nb)

#Multinomial
nb1_classifier = MultinomialNB()
nb1_classifier.fit(X_train, y_train)

y_pred_nb1 = nb1_classifier.predict(X_test)

accuracy_nb1 = accuracy_score(y_test, y_pred_nb1)
precision_nb1 = precision_score(y_test, y_pred_nb1, average='weighted')
recall_nb1 = recall_score(y_test, y_pred_nb1, average='weighted')
f1_score_nb1 = f1_score(y_test, y_pred_nb1, average='weighted')
confusion_matrix_nb1 = confusion_matrix(y_test, y_pred_nb1)

#Bernoulli
nb2_classifier = BernoulliNB(alpha = 2.0,binarize = 2.75, fit_prior=
True, class_prior= None)
nb2_classifier.fit(X_train, y_train)
```

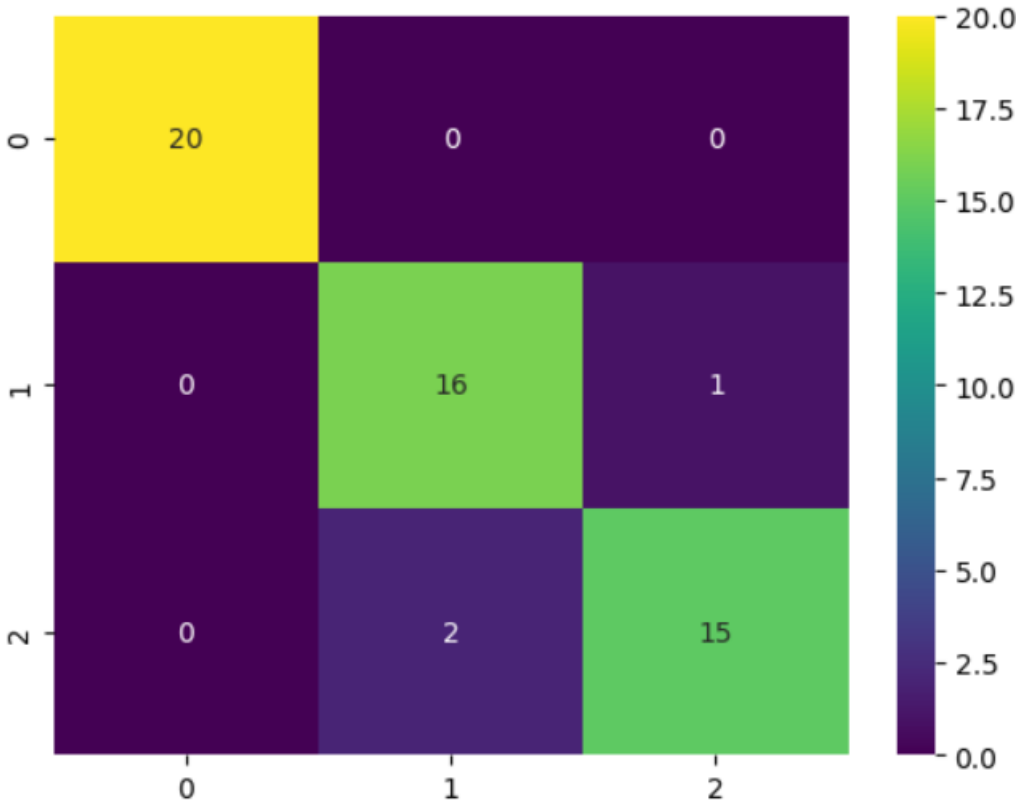
```
y_pred_nb2 = nb2_classifier.predict(X_test)

accuracy_nb2 = accuracy_score(y_test, y_pred_nb2)
precision_nb2 = precision_score(y_test, y_pred_nb2, average='weighted')
recall_nb2 = recall_score(y_test, y_pred_nb2, average='weighted')
f1_score_nb2 = f1_score(y_test, y_pred_nb2, average='weighted')
confusion_matrix_nb2 = confusion_matrix(y_test, y_pred_nb2)

#DecisionTree

dt_classifier = DecisionTreeClassifier(criterion = 'gini',
max_depth=10, random_state=42)
dt_classifier.fit(X_train, y_train)

y_pred_dt = dt_classifier.predict(X_test)


accuracy_dt = accuracy_score(y_test, y_pred_dt)
precision_dt = precision_score(y_test, y_pred_dt, average='weighted')
recall_dt = recall_score(y_test, y_pred_dt, average='weighted')
f1_score_dt = f1_score(y_test, y_pred_dt, average='weighted')
confusion_matrix_dt = confusion_matrix(y_test, y_pred_dt)

print("Gaussian:")
print(f"Accuracy: {accuracy_nb}")
print(f"Precision: {precision_nb}")
print(f"Recall: {recall_nb}")
print(f"F1 Score: {f1_score_nb}")
print("Confusion Matrix:")
print(confusion_matrix_nb)

cm = confusion_matrix(y_test,y_pred_nb)
sns.heatmap(cm,annot=True,fmt="d",cmap='viridis')
plt.show()


print("MulnomialNB:")
print(f"Accuracy: {accuracy_nb1}")
print(f"Precision: {precision_nb1}")
print(f"Recall: {recall_nb1}")
print(f"F1 Score: {f1_score_nb1}")
print("Confusion Matrix:")
print(confusion_matrix_nb1)

cm1 = confusion_matrix(y_test,y_pred_nb1)
sns.heatmap(cm1,annot=True,fmt="d",cmap='viridis')
plt.show()
```

```
print("Bernoulli:")
print(f"Accuracy: {accuracy_nb2}")
print(f"Precision: {precision_nb2}")
print(f"Recall: {recall_nb2}")
print(f"F1 Score: {f1_score_nb2}")
print("Confusion Matrix:")
print(confusion_matrix_nb2)

cm2 = confusion_matrix(y_test,y_pred_nb2)
sns.heatmap(cm2,annot=True,fmt="d",cmap='viridis')
plt.show()


print("\nDecision Tree Classifier:")
print(f"Accuracy: {accuracy_dt}")
print(f"Precision: {precision_dt}")
print(f"Recall: {recall_dt}")
print(f"F1 Score: {f1_score_dt}")
print("Confusion Matrix:")
print(confusion_matrix_dt)

cm3 = confusion_matrix(y_test,y_pred_dt)
sns.heatmap(cm3,annot=True,fmt="d",cmap='viridis')
plt.show()


plt.figure(figsize=(15, 7))
tree.plot_tree(dt_classifier, filled=True,
feature_names=iris.feature_names, class_names=iris.target_names)
plt.show()
```

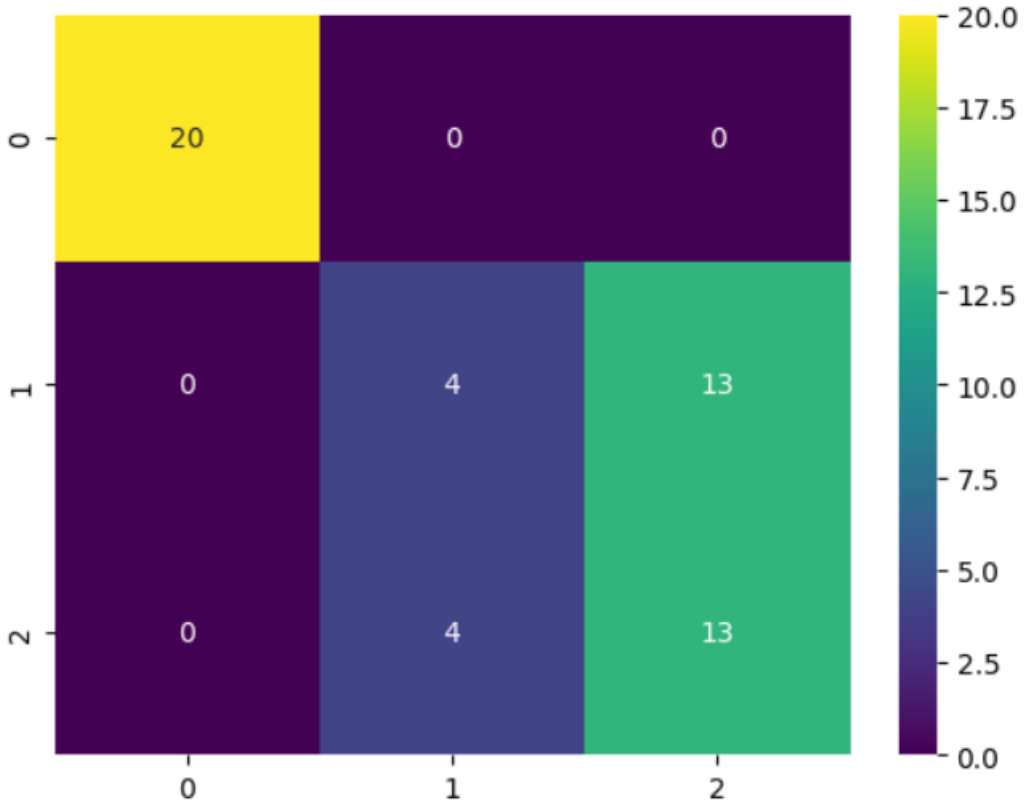
```
Gaussian:
Accuracy: 0.9629629629629629
Precision: 0.9629629629629629
Recall: 0.9629629629629629
F1 Score: 0.9629629629629629
Confusion Matrix:
[[20  0  0]
 [ 0 16  1]
 [ 0  1 16]]
```



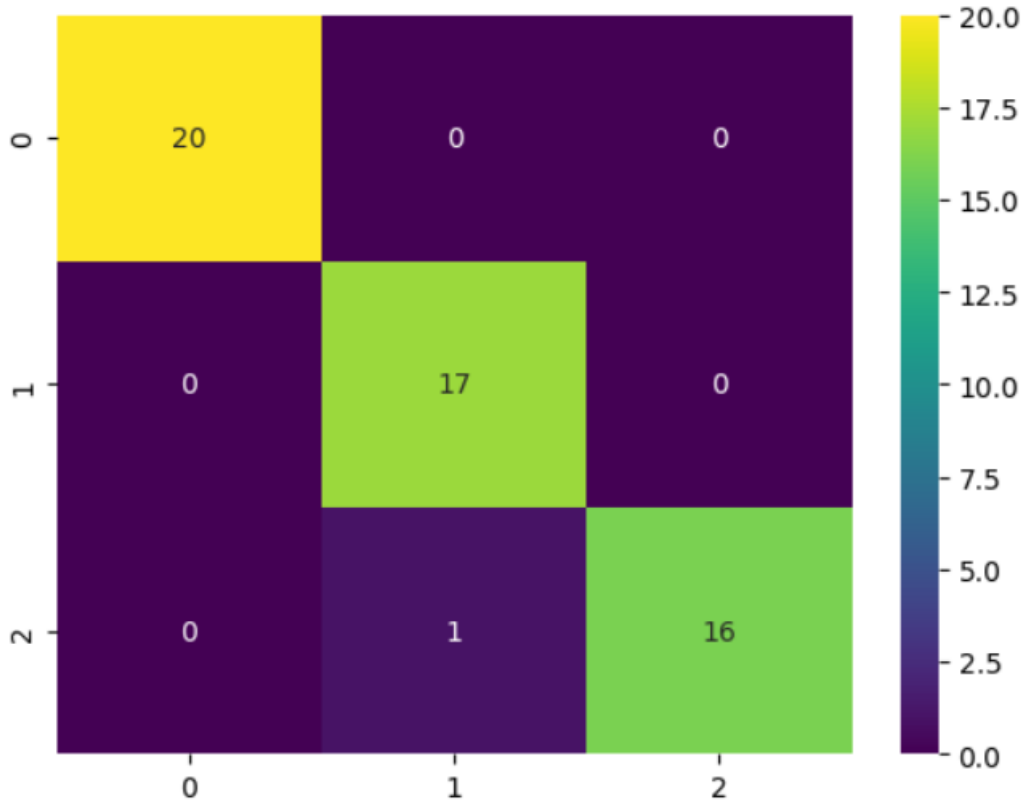
```
MulnomialNB:
Accuracy: 0.9444444444444444
Precision: 0.9453446502057614
Recall: 0.9444444444444444
F1 Score: 0.9443963443963445
Confusion Matrix:
[[20  0  0]
 [ 0 16  1]
 [ 0  2 15]]
```

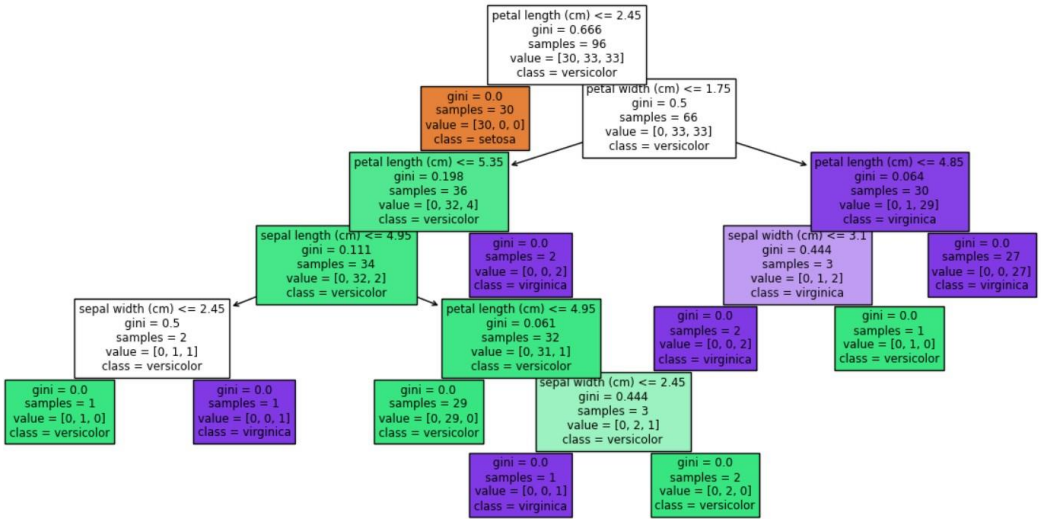


```
Bernoulli:
Accuracy: 0.6851851851851852
Precision: 0.6851851851851852
Recall: 0.6851851851851852
F1 Score: 0.6614642549526271
Confusion Matrix:
[[20  0  0]
 [ 0  4 13]
 [ 0  4 13]]
```



```
Decision Tree Classifier:
Accuracy: 0.9814814814814815
Precision: 0.9825102880658436
Recall: 0.9814814814814815
F1 Score: 0.9814654481321148
Confusion Matrix:
[[20  0  0]
 [ 0 17  0]
 [ 0  1 16]]
```





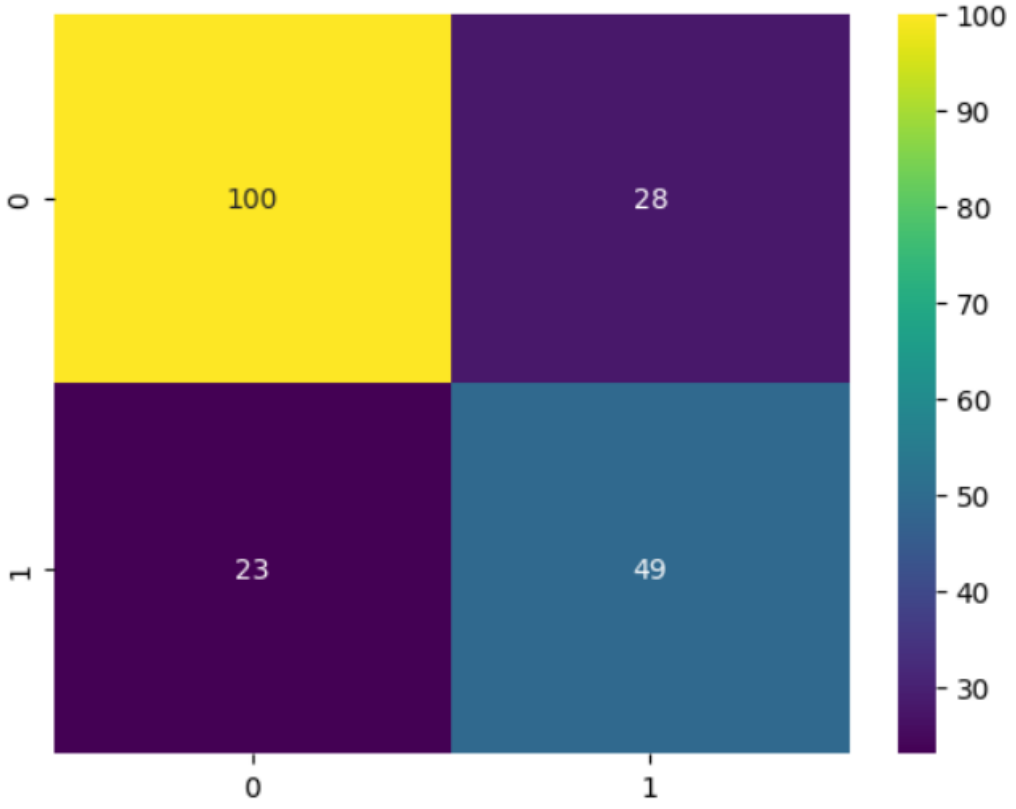
DIABETES

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn import tree
from sklearn.metrics import accuracy_score, confusion_matrix,
precision_score, recall_score, f1_score
import matplotlib.pyplot as plt
import seaborn as sns

!gdown 1eI88UoS-3qf4yunMBls5UqkaRf3U5omQ
diabetes = pd.read_csv('diabetes.csv')
diabetes["Outcome"]=diabetes["Outcome"].apply(str)

X = diabetes.drop("Outcome", axis=1)
y = diabetes["Outcome"]

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.26, random_state=42)

#Gaussian
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)

y_pred_nb = nb_classifier.predict(X_test)

accuracy_nb = accuracy_score(y_test, y_pred_nb)
precision_nb = precision_score(y_test, y_pred_nb, average='weighted')
recall_nb = recall_score(y_test, y_pred_nb, average='weighted')
f1_score_nb = f1_score(y_test, y_pred_nb, average='weighted')
confusion_matrix_nb = confusion_matrix(y_test, y_pred_nb)

#Multinomial
nb1_classifier = MultinomialNB()
nb1_classifier.fit(X_train, y_train)

y_pred_nb1 = nb1_classifier.predict(X_test)

accuracy_nb1 = accuracy_score(y_test, y_pred_nb1)
precision_nb1 = precision_score(y_test, y_pred_nb1, average='weighted')
recall_nb1 = recall_score(y_test, y_pred_nb1, average='weighted')
f1_score_nb1 = f1_score(y_test, y_pred_nb1, average='weighted')
confusion_matrix_nb1 = confusion_matrix(y_test, y_pred_nb1)

#Bernoulli
```

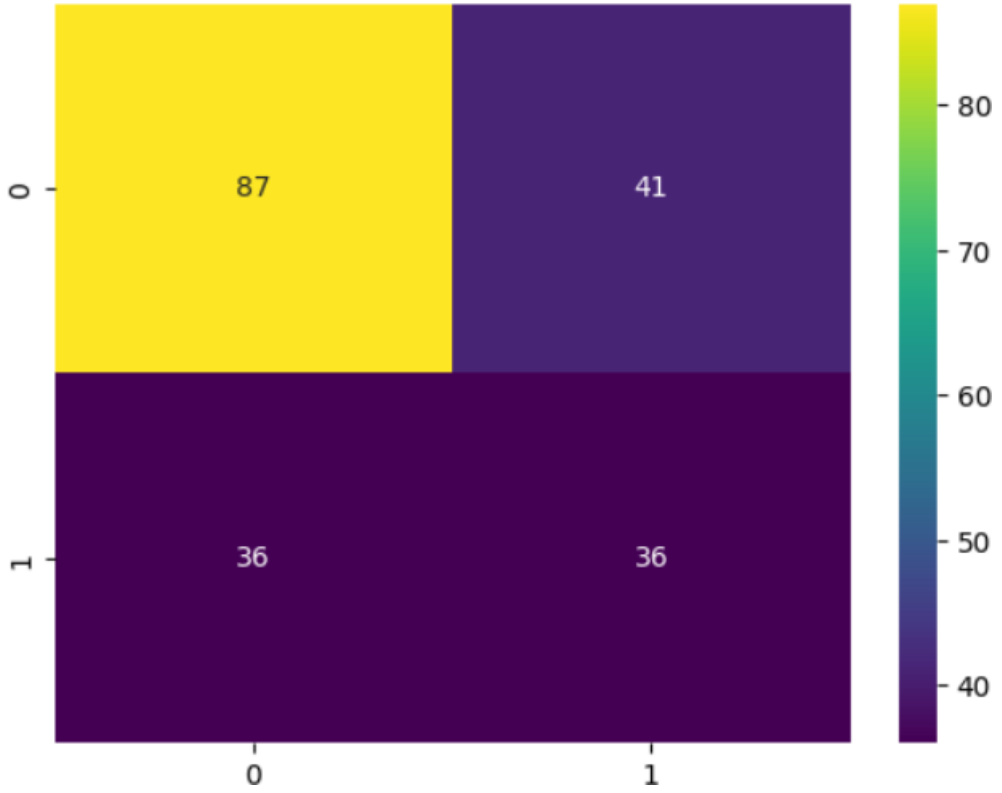
```
nb2_classifier = BernoulliNB(alpha = 2.0,binarize = 2.75, fit_prior=
True, class_prior= None)
nb2_classifier.fit(X_train, y_train)

y_pred_nb2 = nb2_classifier.predict(X_test)

accuracy_nb2 = accuracy_score(y_test, y_pred_nb2)
precision_nb2 = precision_score(y_test, y_pred_nb2, average='weighted')
recall_nb2 = recall_score(y_test, y_pred_nb2, average='weighted')
f1_score_nb2 = f1_score(y_test, y_pred_nb2, average='weighted')
confusion_matrix_nb2 = confusion_matrix(y_test, y_pred_nb2)

#DecisionTree

dt_classifier = DecisionTreeClassifier(criterion = 'gini',
max_depth=10, random_state=42)
dt_classifier.fit(X_train, y_train)

y_pred_dt = dt_classifier.predict(X_test)


accuracy_dt = accuracy_score(y_test, y_pred_dt)
precision_dt = precision_score(y_test, y_pred_dt, average='weighted')
recall_dt = recall_score(y_test, y_pred_dt, average='weighted')
f1_score_dt = f1_score(y_test, y_pred_dt, average='weighted')
confusion_matrix_dt = confusion_matrix(y_test, y_pred_dt)

print("Gaussian:")
print(f"Accuracy: {accuracy_nb}")
print(f"Precision: {precision_nb}")
print(f"Recall: {recall_nb}")
print(f"F1 Score: {f1_score_nb}")
print("Confusion Matrix:")
print(confusion_matrix_nb)

cm = confusion_matrix(y_test,y_pred_nb)
sns.heatmap(cm,annot=True,fmt="d",cmap='viridis')
plt.show()

print("MulnomialNB:")
print(f"Accuracy: {accuracy_nb1}")
print(f"Precision: {precision_nb1}")
print(f"Recall: {recall_nb1}")
print(f"F1 Score: {f1_score_nb1}")
print("Confusion Matrix:")
print(confusion_matrix_nb1)

cm1 = confusion_matrix(y_test,y_pred_nb1)
sns.heatmap(cm1,annot=True,fmt="d",cmap='viridis')
```

```
plt.show()

print("Bernoulli:")
print(f"Accuracy: {accuracy_nb2}")
print(f"Precision: {precision_nb2}")
print(f"Recall: {recall_nb2}")
print(f"F1 Score: {f1_score_nb2}")
print("Confusion Matrix:")
print(confusion_matrix_nb2)

cm2 = confusion_matrix(y_test,y_pred_nb2)
sns.heatmap(cm2,annot=True,fmt="d",cmap='viridis')
plt.show()

print("\nDecision Tree Classifier:")
print(f"Accuracy: {accuracy_dt}")
print(f"Precision: {precision_dt}")
print(f"Recall: {recall_dt}")
print(f"F1 Score: {f1_score_dt}")
print("Confusion Matrix:")
print(confusion_matrix_dt)

cm3 = confusion_matrix(y_test,y_pred_dt)
sns.heatmap(cm3,annot=True,fmt="d",cmap='viridis')
plt.show()


plt.figure(figsize=(15, 10))
tree.plot_tree(dt_classifier, filled=True, feature_names=X.columns,
class_names=y.unique())
plt.show()
```

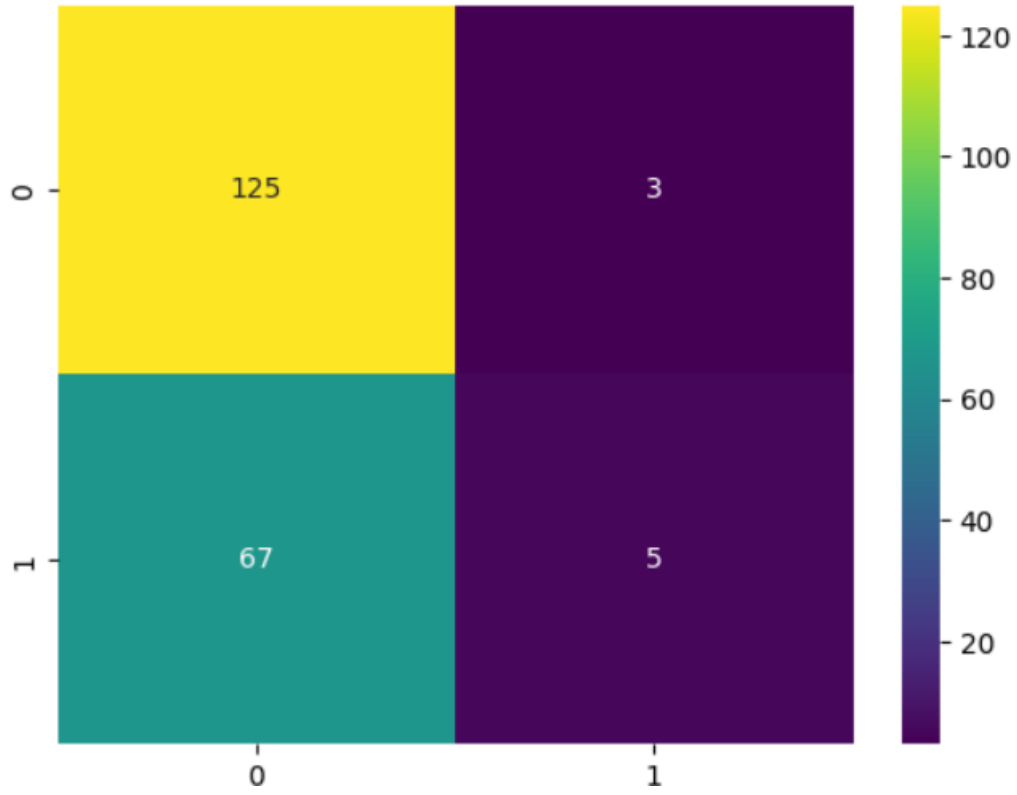
```
Gaussian:
Accuracy: 0.745
Precision: 0.7494161123429416
Recall: 0.745
F1 Score: 0.7467386828524826
Confusion Matrix:
[[100  28]
 [ 23  49]]
```



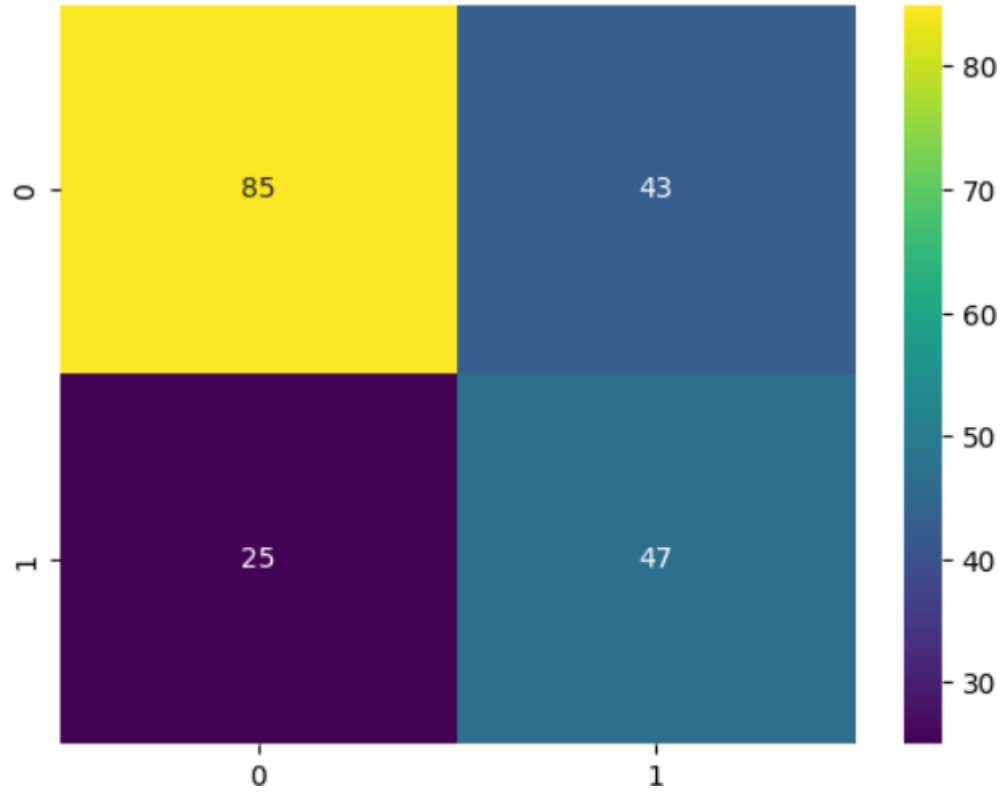
```
MulnomialNB:
Accuracy: 0.615
Precision: 0.6209946151409566
Recall: 0.615
F1 Score: 0.6176250701890424
Confusion Matrix:
[[87 41]
 [36 36]]
```

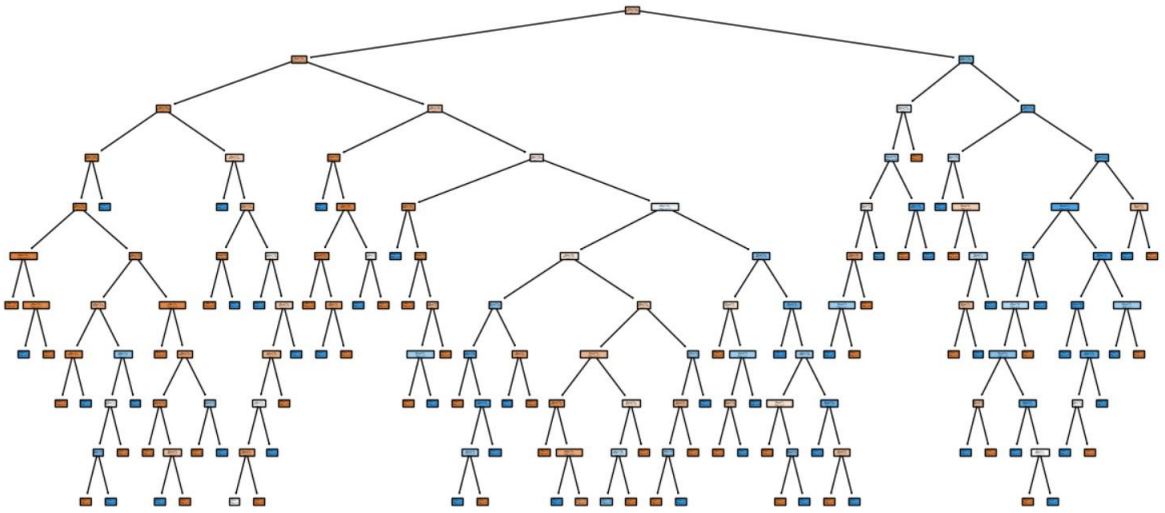


```
Bernoulli:
Accuracy: 0.65
Precision: 0.6416666666666666
Recall: 0.65
F1 Score: 0.545
Confusion Matrix:
[[125   3]
 [ 67   5]]
```



```
Decision Tree Classifier:
Accuracy: 0.66
Precision: 0.6825454545454545
Recall: 0.66
F1 Score: 0.6660317460317461
Confusion Matrix:
[[85 43]
 [25 47]]
```





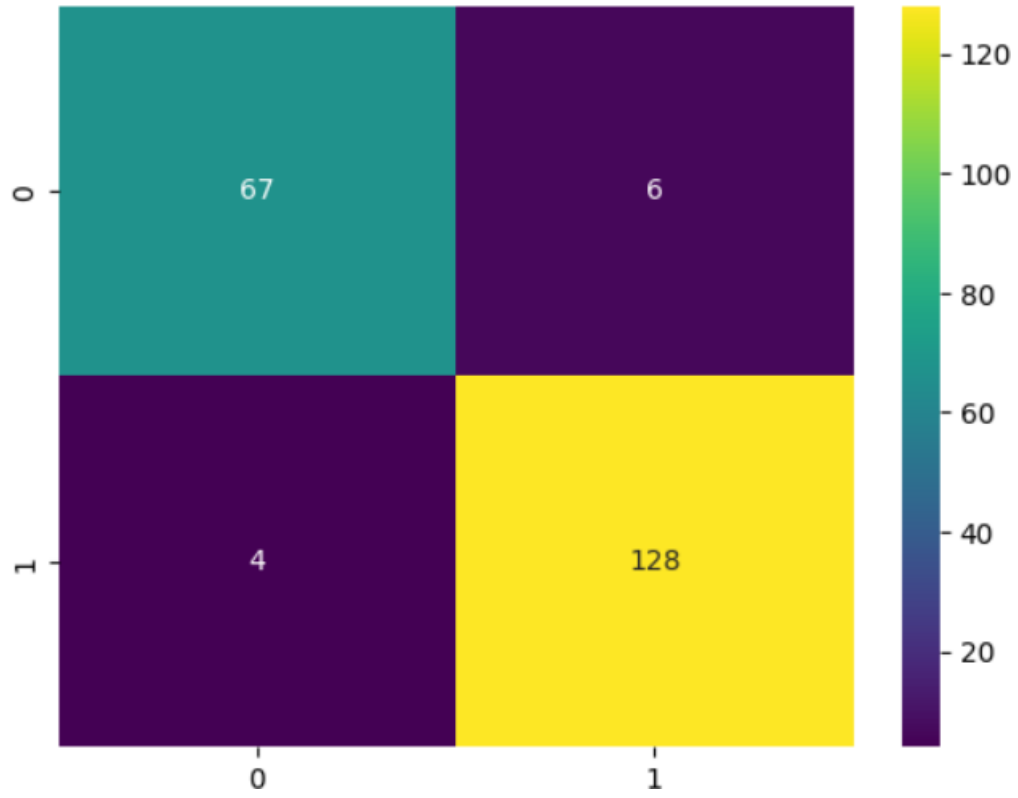
BREAST CANCER

```
import numpy as np
from sklearn.datasets  import  load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB, MultinomialNB, BernoulliNB
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

cancer = load_breast_cancer()
X, y = cancer.data, cancer.target

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.36, random_state=42)

#Gaussian
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)

y_pred_nb = nb_classifier.predict(X_test)

accuracy_nb = accuracy_score(y_test, y_pred_nb)
precision_nb = precision_score(y_test, y_pred_nb, average='weighted')
recall_nb = recall_score(y_test, y_pred_nb, average='weighted')
f1_score_nb = f1_score(y_test, y_pred_nb, average='weighted')
confusion_matrix_nb = confusion_matrix(y_test, y_pred_nb)

#Multinomial
nb1_classifier = MultinomialNB()
nb1_classifier.fit(X_train, y_train)

y_pred_nb1 = nb1_classifier.predict(X_test)

accuracy_nb1 = accuracy_score(y_test, y_pred_nb1)
precision_nb1 = precision_score(y_test, y_pred_nb1, average='weighted')
recall_nb1 = recall_score(y_test, y_pred_nb1, average='weighted')
f1_score_nb1 = f1_score(y_test, y_pred_nb1, average='weighted')
confusion_matrix_nb1 = confusion_matrix(y_test, y_pred_nb1)

#Bernoulli
nb2_classifier = BernoulliNB(alpha = 2.0,binarize = 2.75, fit_prior=
True, class_prior= None)
nb2_classifier.fit(X_train, y_train)
```

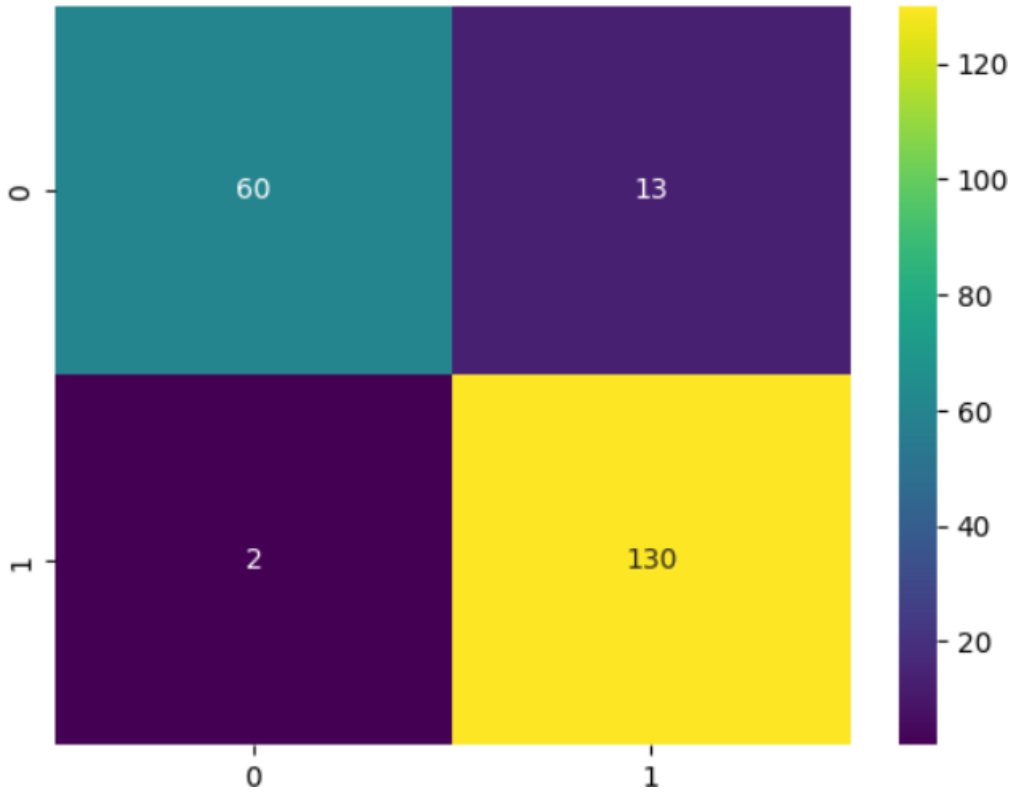
```
y_pred_nb2 = nb2_classifier.predict(X_test)

accuracy_nb2 = accuracy_score(y_test, y_pred_nb2)
precision_nb2 = precision_score(y_test, y_pred_nb2, average='weighted')
recall_nb2 = recall_score(y_test, y_pred_nb2, average='weighted')
f1_score_nb2 = f1_score(y_test, y_pred_nb2, average='weighted')
confusion_matrix_nb2 = confusion_matrix(y_test, y_pred_nb2)

#DecisionTree

dt_classifier = DecisionTreeClassifier(criterion = 'gini',
max_depth=10, random_state=42)
dt_classifier.fit(X_train, y_train)

y_pred_dt = dt_classifier.predict(X_test)


accuracy_dt = accuracy_score(y_test, y_pred_dt)
precision_dt = precision_score(y_test, y_pred_dt, average='weighted')
recall_dt = recall_score(y_test, y_pred_dt, average='weighted')
f1_score_dt = f1_score(y_test, y_pred_dt, average='weighted')
confusion_matrix_dt = confusion_matrix(y_test, y_pred_dt)

print("Gaussian:")
print(f"Accuracy: {accuracy_nb}")
print(f"Precision: {precision_nb}")
print(f"Recall: {recall_nb}")
print(f"F1 Score: {f1_score_nb}")
print("Confusion Matrix:")
print(confusion_matrix_nb)

cm = confusion_matrix(y_test,y_pred_nb)
sns.heatmap(cm,annot=True,fmt="d",cmap='viridis')
plt.show()


print("MulnomialNB:")
print(f"Accuracy: {accuracy_nb1}")
print(f"Precision: {precision_nb1}")
print(f"Recall: {recall_nb1}")
print(f"F1 Score: {f1_score_nb1}")
print("Confusion Matrix:")
print(confusion_matrix_nb1)

cm1 = confusion_matrix(y_test,y_pred_nb1)
sns.heatmap(cm1,annot=True,fmt="d",cmap='viridis')
plt.show()
```

```
print("Bernoulli:")
print(f"Accuracy: {accuracy_nb2}")
print(f"Precision: {precision_nb2}")
print(f"Recall: {recall_nb2}")
print(f"F1 Score: {f1_score_nb2}")
print("Confusion Matrix:")
print(confusion_matrix_nb2)

cm2 = confusion_matrix(y_test,y_pred_nb2)
sns.heatmap(cm2,annot=True,fmt="d",cmap='viridis')
plt.show()


print("\nDecision Tree Classifier:")
print(f"Accuracy: {accuracy_dt}")
print(f"Precision: {precision_dt}")
print(f"Recall: {recall_dt}")
print(f"F1 Score: {f1_score_dt}")
print("Confusion Matrix:")
print(confusion_matrix_dt)

cm3 = confusion_matrix(y_test,y_pred_dt)
sns.heatmap(cm3,annot=True,fmt="d",cmap='viridis')
plt.show()


plt.figure(figsize=(15, 10))
tree.plot_tree(dt_classifier, filled=True,
feature_names=cancer.feature_names, class_names=cancer.target_names)
plt.show()
```

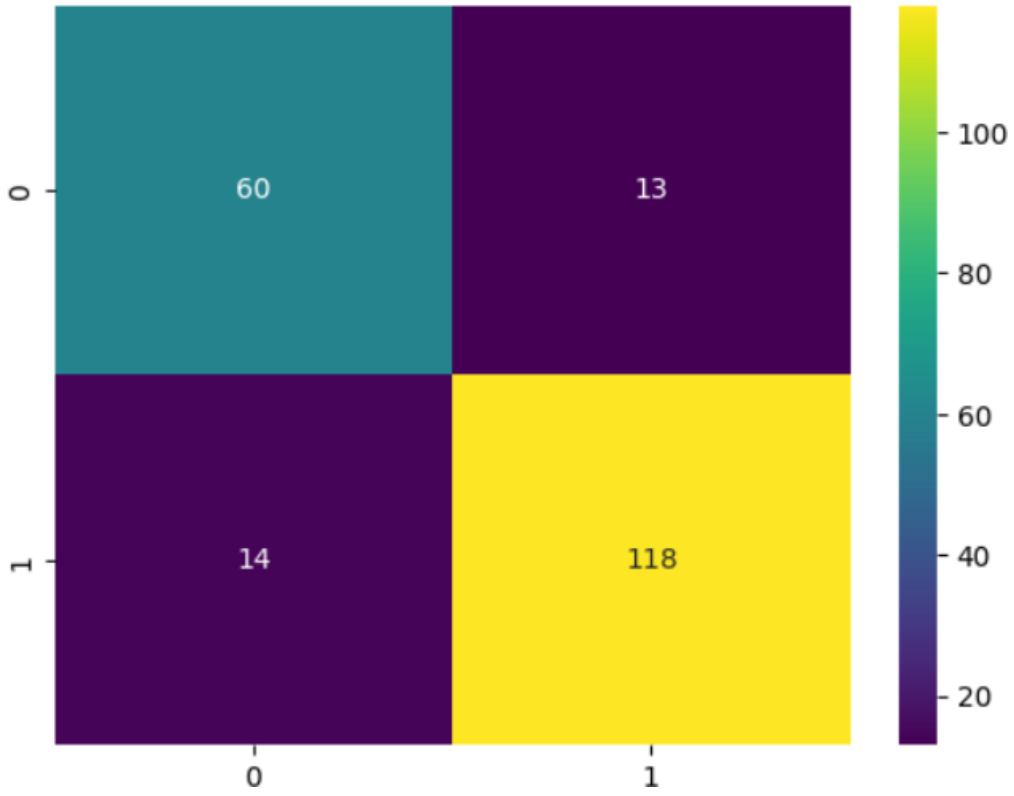
```
Gaussian:
Accuracy: 0.9512195121951219
Precision: 0.9511067130852096
Recall: 0.9512195121951219
F1 Score: 0.9510641441000877
Confusion Matrix:
[[ 67   6]
 [  4 128]]
```



```
MulnomialNB:
Accuracy: 0.926829268292683
Precision: 0.9299763965381589
Recall: 0.926829268292683
F1 Score: 0.9253116531165311
Confusion Matrix:
[[ 60  13]
 [  2 130]]
```



```
Bernoulli:
Accuracy: 0.8682926829268293
Precision: 0.8687314758437454
Recall: 0.8682926829268293
F1 Score: 0.8684902728595842
Confusion Matrix:
[[ 60  13]
 [ 14 118]]
```



Decision Tree Classifier:
Accuracy: 0.9219512195121952
Precision: 0.9225966228893058
Recall: 0.9219512195121952
F1 Score: 0.9221806800283806
Confusion Matrix:
[[66 7]
 [9 123]]