akilus-sayadat-037-iris-dataset-1

August 16, 2023

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[ ]:
```
from sklearn.datasets import load_iris
iris = load_iris()
df = pd.DataFrame(iris.data, columns = iris.feature_names)
column_names = list(df.columns.values)
df.head()
```

[ ]:
```
X = df.iloc[:][:]
y = iris["target"]
dict_bnb = {}
dict_mnb = {}
dict_gnb = {}
dict_dtr = {}
RocAucbnb = {}
RocAucmnb = {}
RocAucgnb = {}
RocAucdtr = {}
print(X, y)
```

[4]:
```
def plot(y_test, y_pred):
  from sklearn.metrics import confusion_matrix
  import seaborn as sns

  print("Confusion Matrix : ")
  cf_matrix = confusion_matrix(y_test, y_pred)
  group_counts = ["{0:0.0f}".format(value) for value in
                  cf_matrix.flatten()]
  group_percentages = ["{0:.2%}".format(value) for value in
                       cf_matrix.flatten()/np.sum(cf_matrix)]
  labels = [f"{v1}\n{v2}" for v1, v2 in
            zip(group_counts,group_percentages)]
  labels = np.asarray(labels).reshape(3,3)
  plt.figure(figsize=(6, 4))
```

```
  sns.heatmap(cf_matrix, annot=labels, fmt='', cmap='Blues', xticklabels = iris.
↪target_names, yticklabels=iris.target_names)
  plt.xlabel('Predicted')
  plt.ylabel('Actual')
  plt.title('Confusion Matrix')
  plt.show()
  print("*********************************************")
```

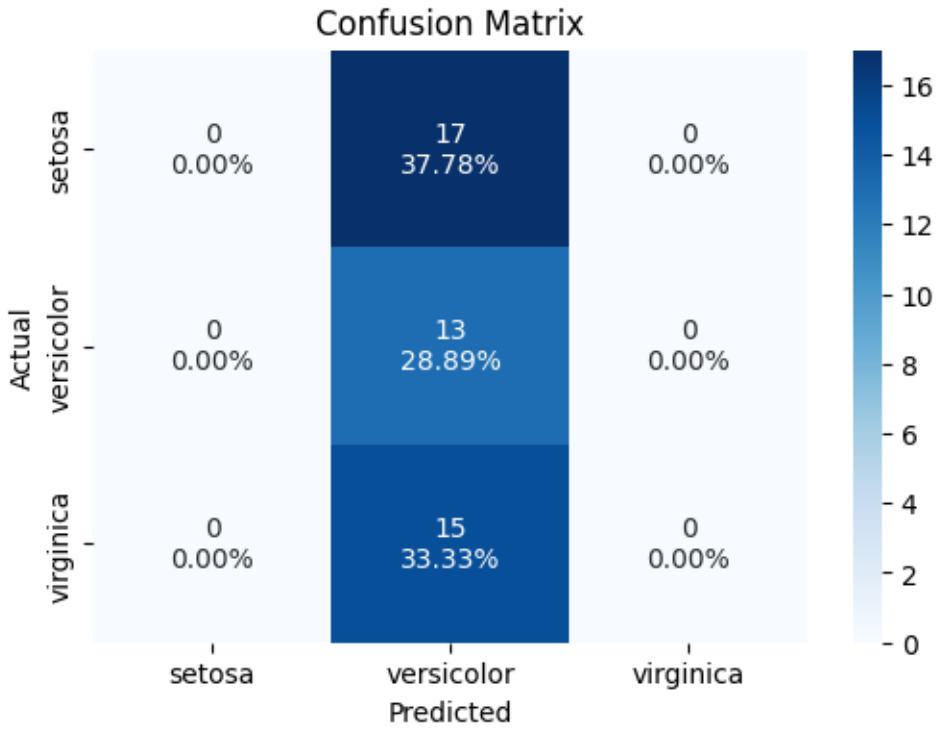
[5]:
```
def reports(y_test, y_pred):
  from sklearn.metrics import classification_report
  plot(y_test, y_pred)
  print("*********************************************")
  print("Classification Evaluation : ")
  print(classification_report(y_test, y_pred, zero_division = 0))
```

0.0.1 Classification using BernoulliNB Naive Bayes

[6]:
```
def FBouBernoulli(split, alpha_value = 1.0, binarize_value = 0.0,␣
↪fit_prior_value = False):
  from sklearn.naive_bayes import BernoulliNB
  from sklearn.metrics import accuracy_score
  from sklearn.model_selection import train_test_split
  from sklearn.preprocessing import StandardScaler
  #scaler = StandardScaler()
  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = split,␣
↪random_state=44)
  #scaler.fit_transform(X_train)
  #scaler.transform(X_test)
  classifier = BernoulliNB(alpha = alpha_value, binarize = binarize_value,␣
↪fit_prior = fit_prior_value)
  classifier.fit(X_train, y_train)
  y_pred = classifier.predict(X_test)
  print("Train-test split: " + str(split))
  print("value: alpha: "+str(alpha_value) + " binarize: " + str(binarize_value)␣
↪+ " fit_prior: " +str(fit_prior_value))
  print("*********************************************")
  accuracy = accuracy_score(y_test, y_pred)
  if str(split) in dict_bnb:
    dict_bnb[str(split)] = max(accuracy, dict_bnb[str(split)])
    if str(split) == '0.3' and accuracy > dict_bnb[str(split)]:
      RocAucbnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
  else:
    dict_bnb[str(split)] = accuracy
    if str(split) == '0.3':
      RocAucbnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
  reports(y_test, y_pred)
```

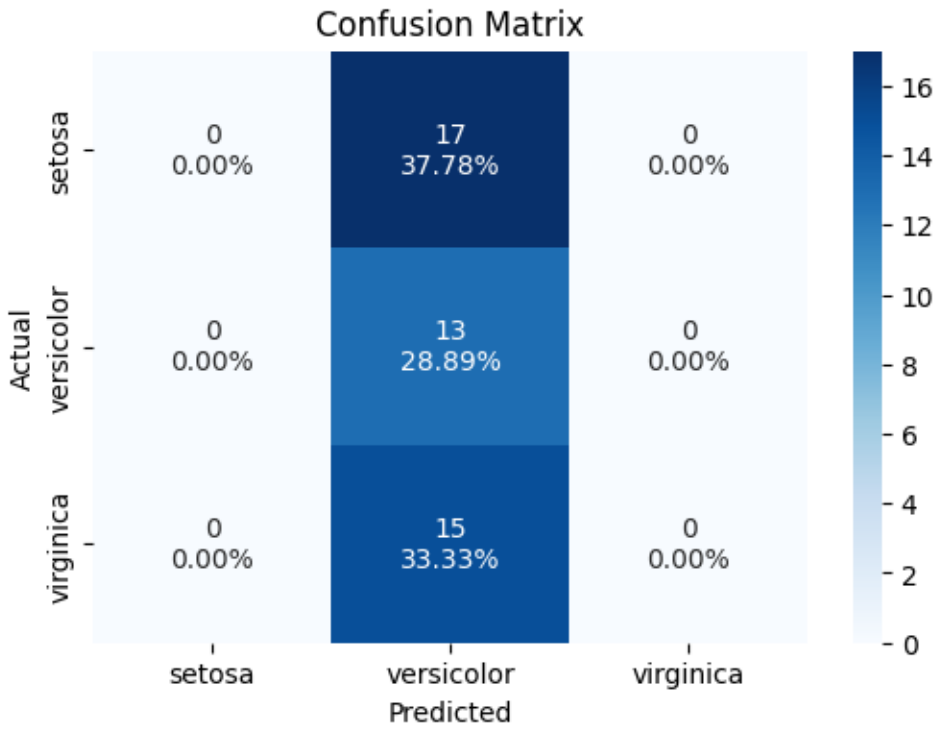
[7]:
```
## Train-Test split 0.3
FBouBernoulli(0.3)
FBouBernoulli(0.3, 1.0)
FBouBernoulli(0.3, 1.0, 1.8)
FBouBernoulli(0.3, 1.0, 1.8, True)
```

```
Train-test split: 0.3
value: alpha: 1.0 binarize: 0.0 fit_prior: False
**********************************************
Confusion Matrix :
```

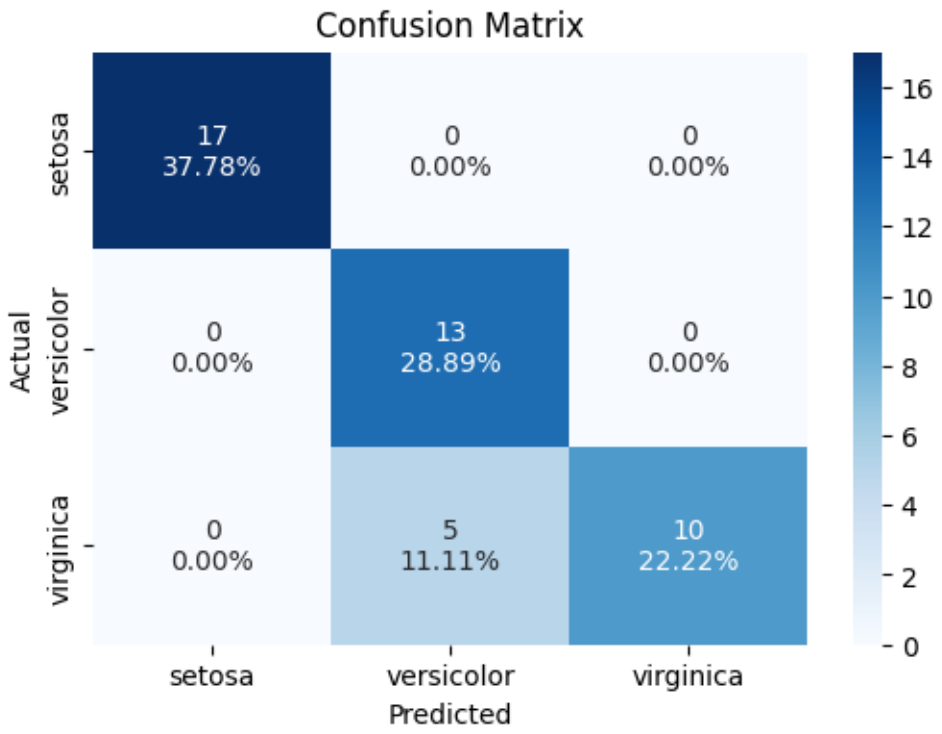



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        17
           1       0.29      1.00      0.45        13
           2       0.00      0.00      0.00        15

    accuracy                           0.29        45
   macro avg       0.10      0.33      0.15        45
weighted avg       0.08      0.29      0.13        45
```

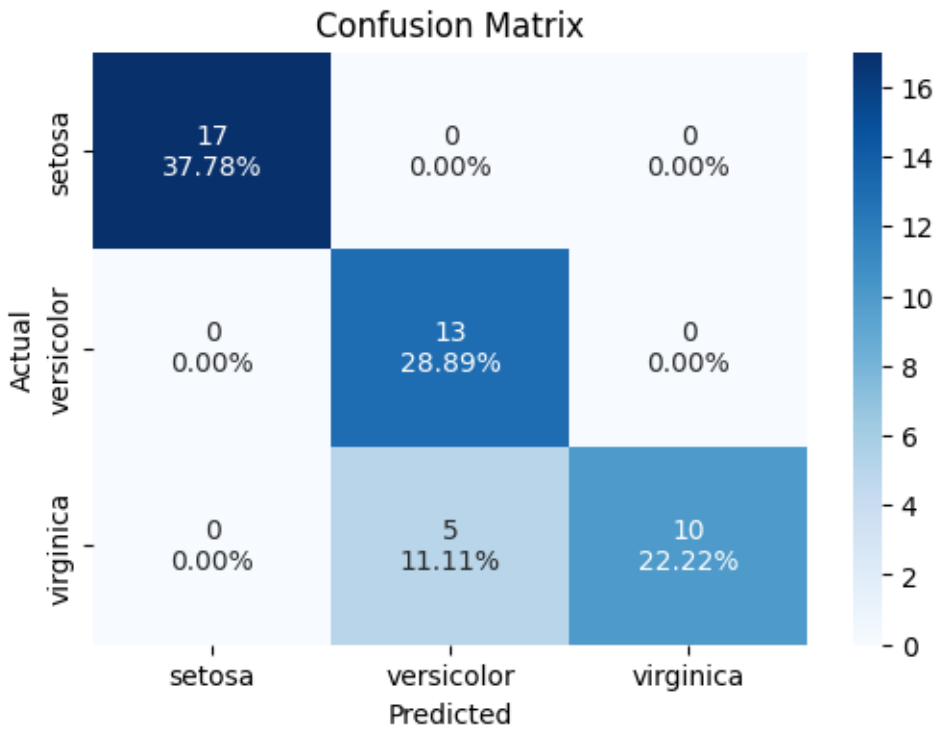
```
Train-test split: 0.3
value: alpha: 1.0 binarize: 0.0 fit_prior: False
*********************************************
Confusion Matrix :
```




```
*********************************************
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        17
           1       0.29      1.00      0.45        13
           2       0.00      0.00      0.00        15

    accuracy                           0.29        45
   macro avg       0.10      0.33      0.15        45
weighted avg       0.08      0.29      0.13        45

Train-test split: 0.3
value: alpha: 1.0 binarize: 1.8 fit_prior: False
*********************************************
Confusion Matrix :
```



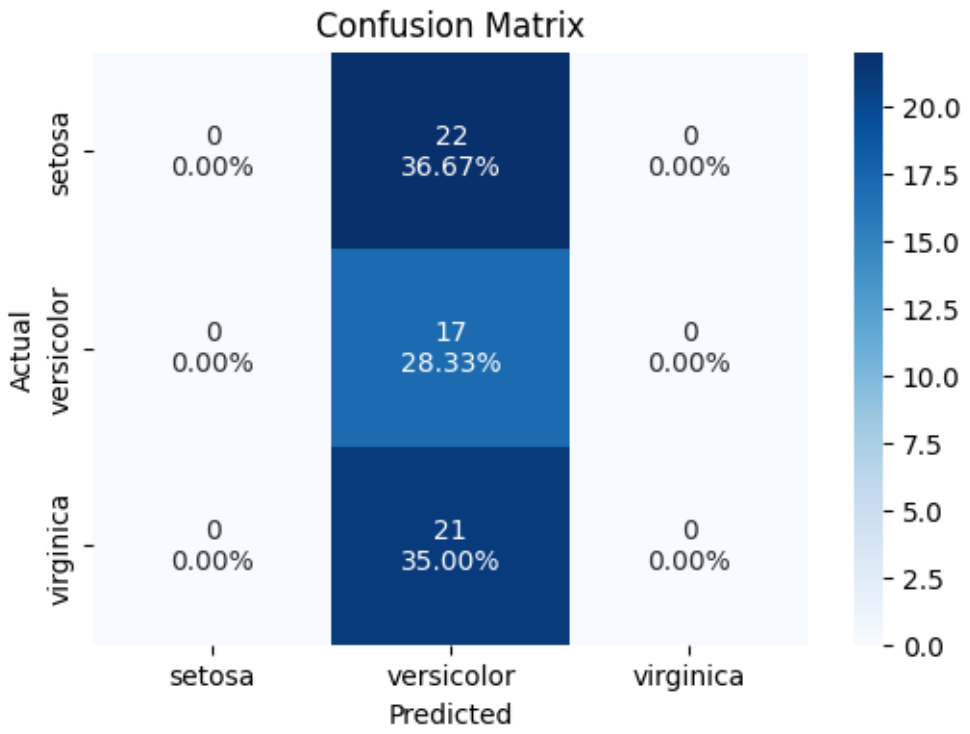

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        17
           1       0.72      1.00      0.84        13
           2       1.00      0.67      0.80        15

    accuracy                           0.89        45
   macro avg       0.91      0.89      0.88        45
weighted avg       0.92      0.89      0.89        45

Train-test split: 0.3
value: alpha: 1.0 binarize: 1.8 fit_prior: True
**********************************************
Confusion Matrix :
```



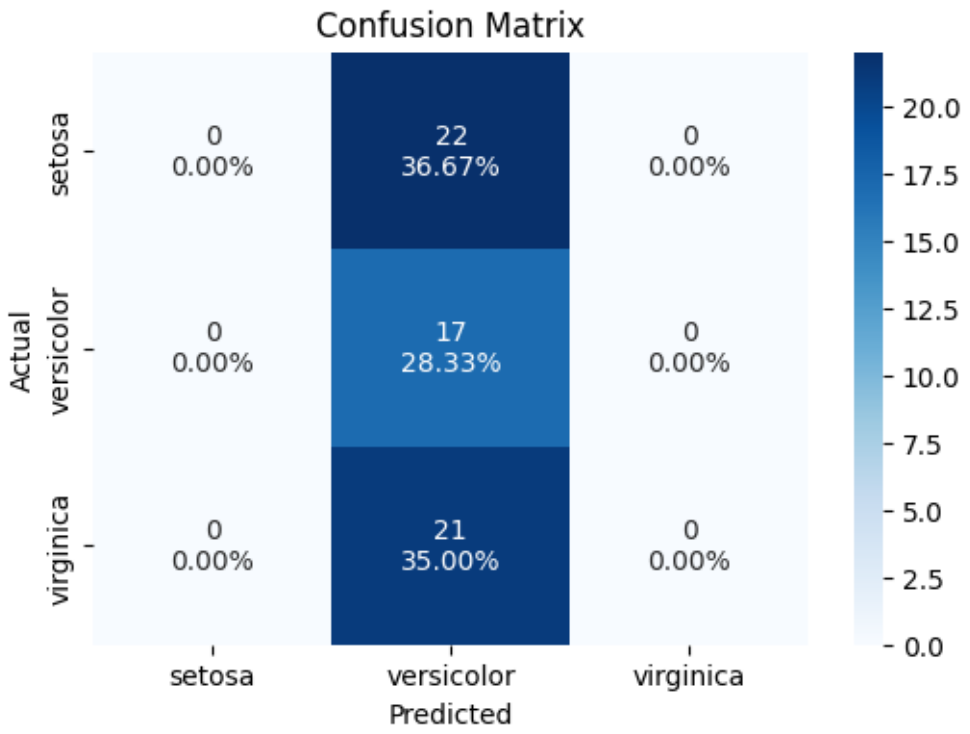

```
*********************************************
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        17
           1       0.72      1.00      0.84        13
           2       1.00      0.67      0.80        15

    accuracy                           0.89        45
   macro avg       0.91      0.89      0.88        45
weighted avg       0.92      0.89      0.89        45
```

[8]:
```
## Train-Test split 0.4
FBouBernoulli(0.4)
FBouBernoulli(0.4, 1.0)
FBouBernoulli(0.4, 1.0, 1.7)
FBouBernoulli(0.4, 1.0, 1.7, True)
```

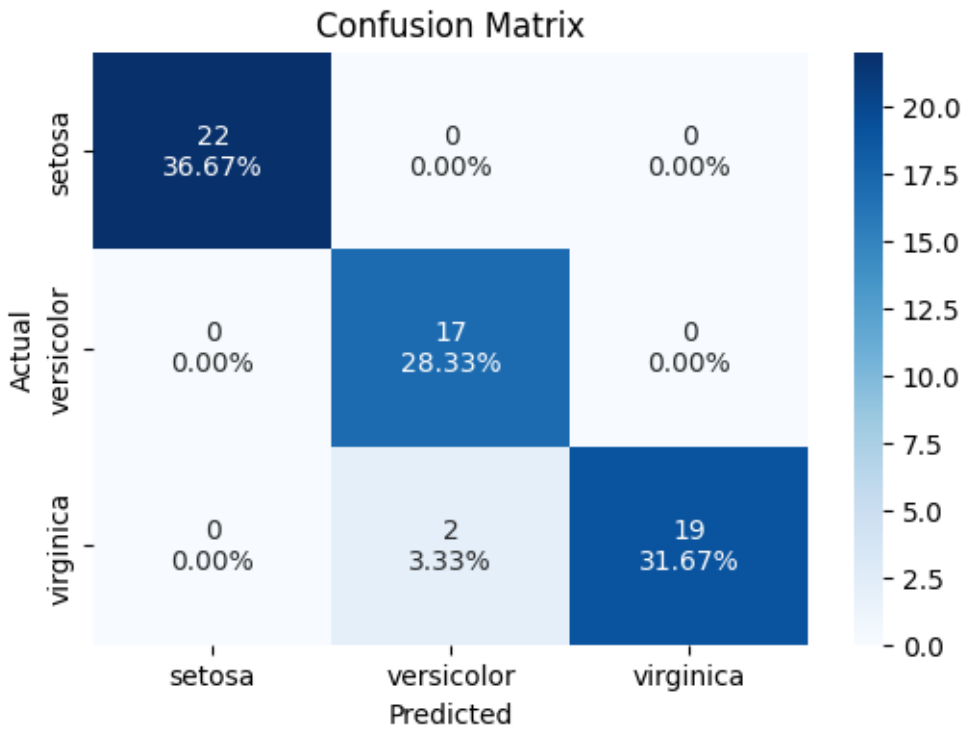
```
Train-test split: 0.4
value: alpha: 1.0 binarize: 0.0 fit_prior: False
*********************************************
Confusion Matrix :
```



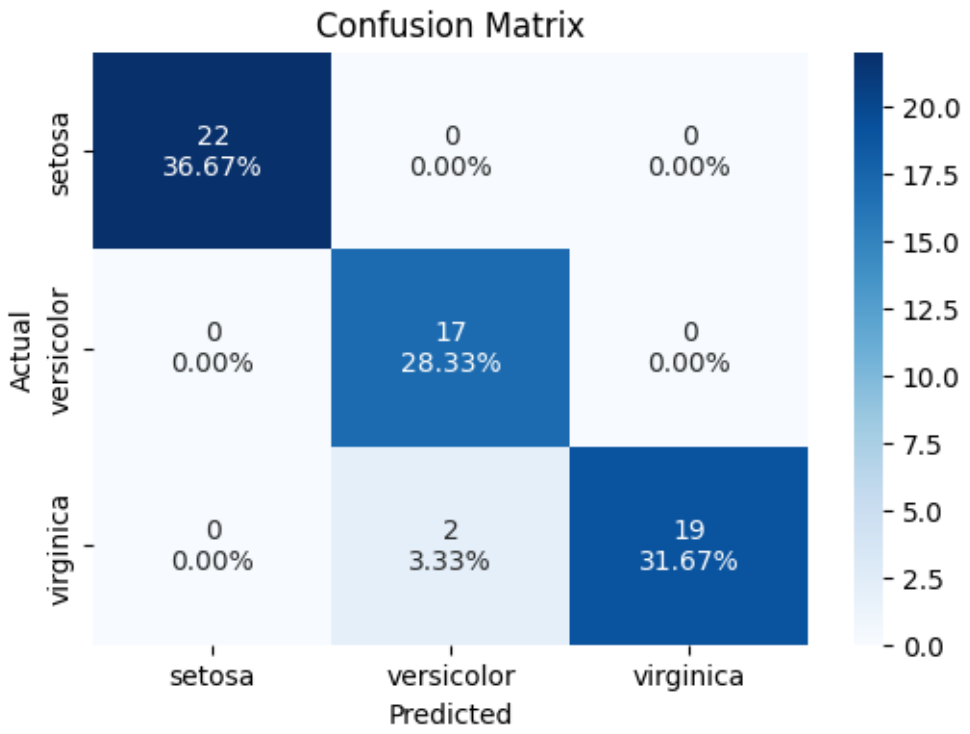

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        22
           1       0.28      1.00      0.44        17
           2       0.00      0.00      0.00        21

    accuracy                           0.28        60
   macro avg       0.09      0.33      0.15        60
weighted avg       0.08      0.28      0.13        60

Train-test split: 0.4
value: alpha: 1.0 binarize: 0.0 fit_prior: False
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        22
           1       0.28      1.00      0.44        17
           2       0.00      0.00      0.00        21

    accuracy                           0.28        60
   macro avg       0.09      0.33      0.15        60
weighted avg       0.08      0.28      0.13        60

Train-test split: 0.4
value: alpha: 1.0 binarize: 1.7 fit_prior: False
**********************************************
Confusion Matrix :
```



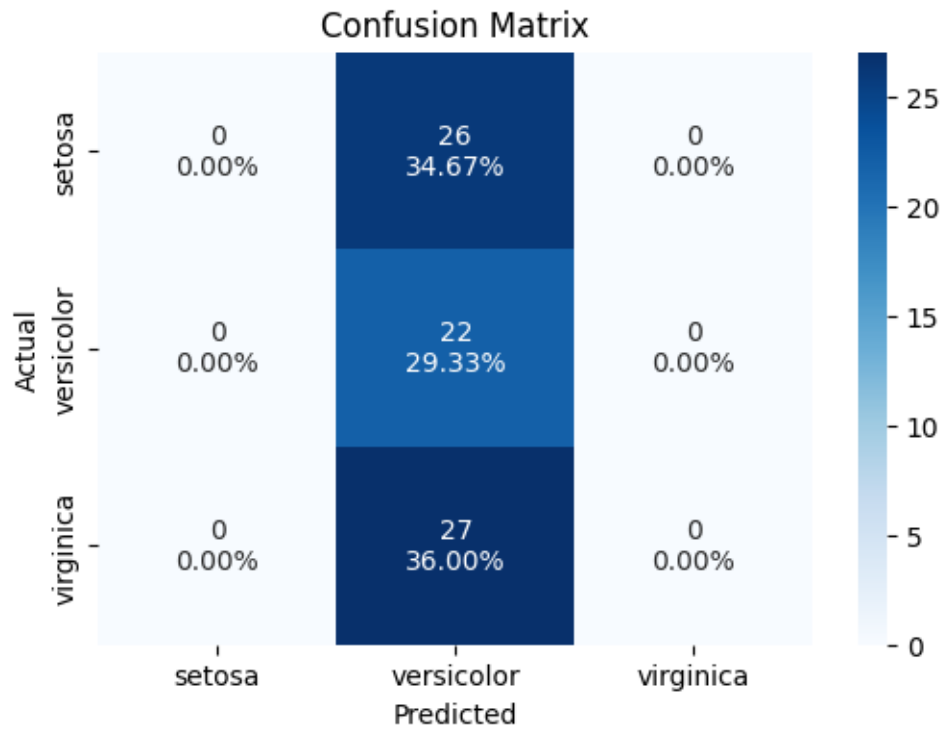

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        22
           1       0.89      1.00      0.94        17
           2       1.00      0.90      0.95        21

    accuracy                           0.97        60
   macro avg       0.96      0.97      0.96        60
weighted avg       0.97      0.97      0.97        60

Train-test split: 0.4
value: alpha: 1.0 binarize: 1.7 fit_prior: True
**********************************************
Confusion Matrix :
```



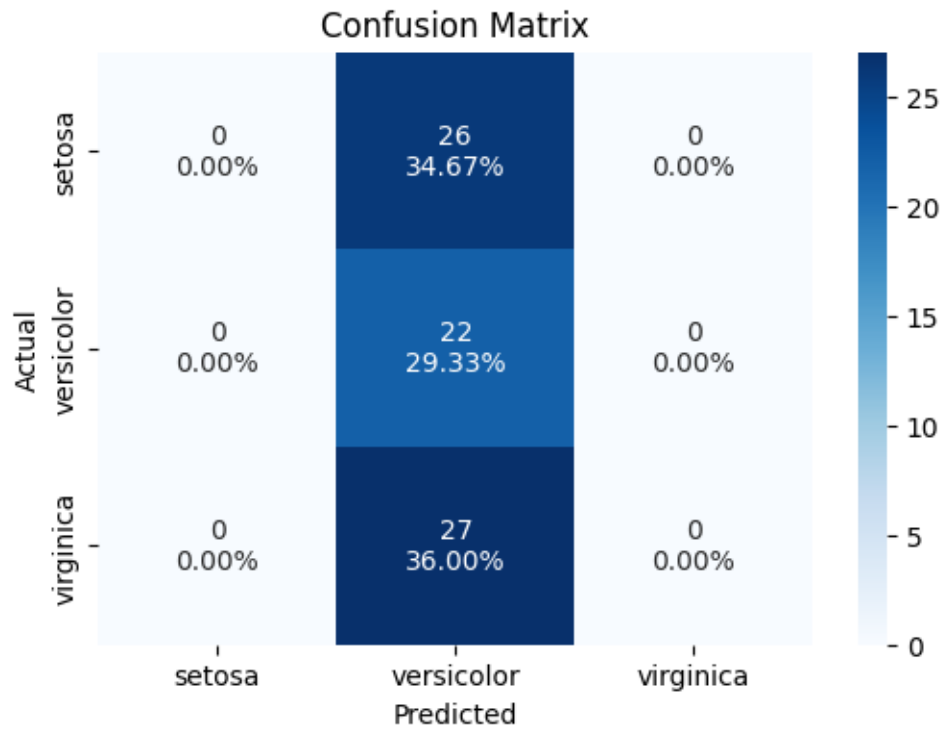

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        22
           1       0.89      1.00      0.94        17
           2       1.00      0.90      0.95        21

    accuracy                           0.97        60
   macro avg       0.96      0.97      0.96        60
weighted avg       0.97      0.97      0.97        60
```

[9]:
```
## Train-Test split 0.5
FBouBernoulli(0.5)
FBouBernoulli(0.5, 1.0)
FBouBernoulli(0.5, 1.0, 1.75)
FBouBernoulli(0.5, 1.0, 1.75, True)
```

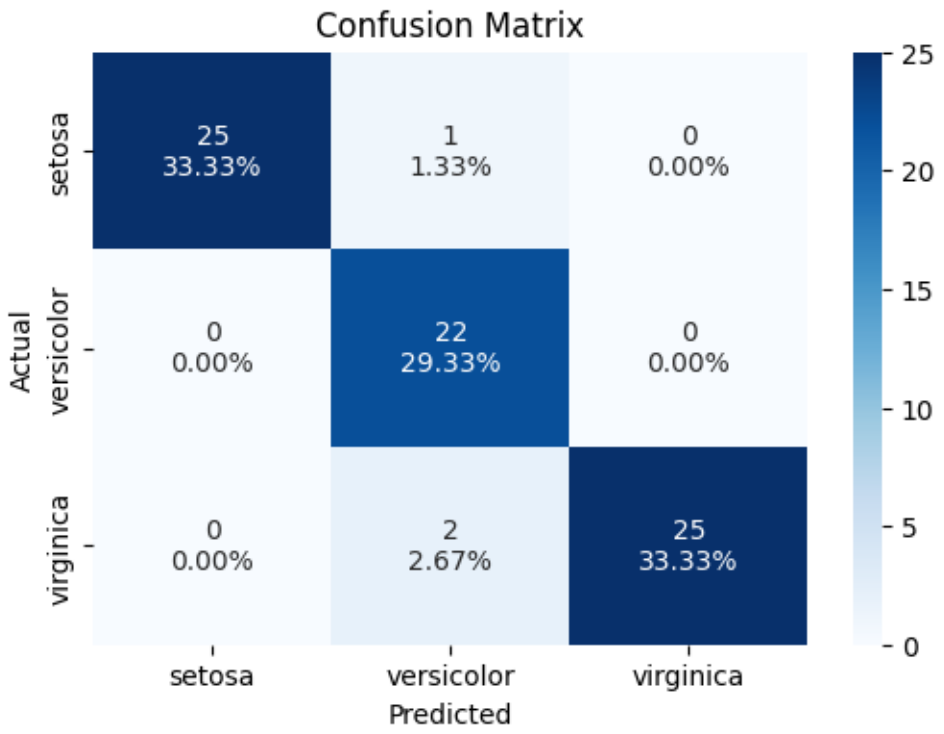
```
Train-test split: 0.5
value: alpha: 1.0 binarize: 0.0 fit_prior: False
**********************************************
Confusion Matrix :
```



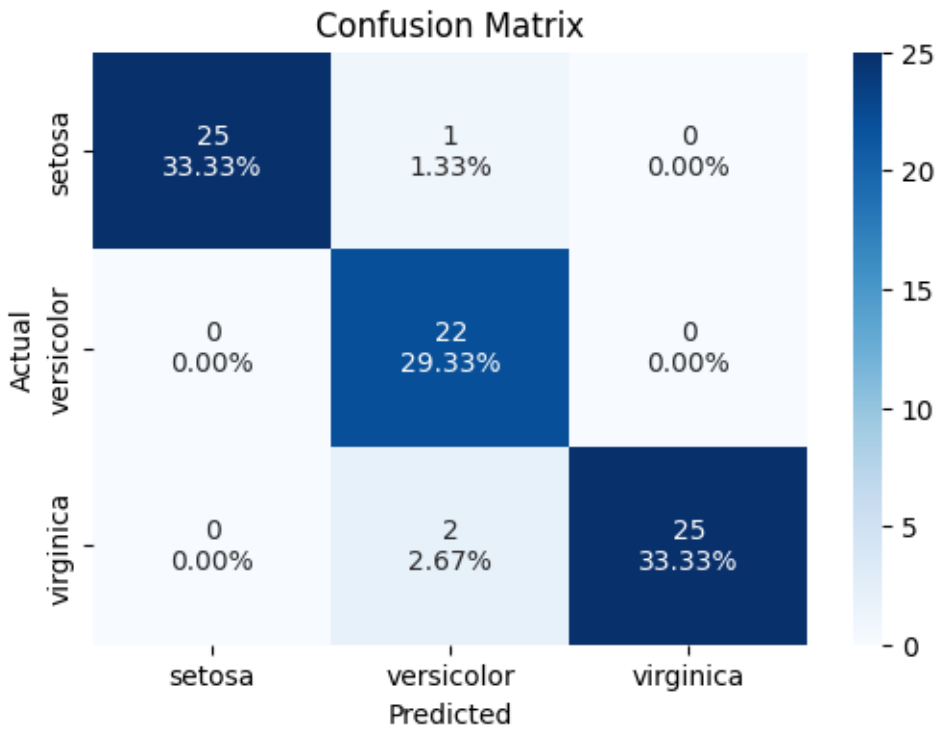

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        26
           1       0.29      1.00      0.45        22
           2       0.00      0.00      0.00        27

    accuracy                           0.29        75
   macro avg       0.10      0.33      0.15        75
weighted avg       0.09      0.29      0.13        75

Train-test split: 0.5
value: alpha: 1.0 binarize: 0.0 fit_prior: False
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        26
           1       0.29      1.00      0.45        22
           2       0.00      0.00      0.00        27

    accuracy                           0.29        75
   macro avg       0.10      0.33      0.15        75
weighted avg       0.09      0.29      0.13        75

Train-test split: 0.5
value: alpha: 1.0 binarize: 1.75 fit_prior: False
**********************************************
Confusion Matrix :
```

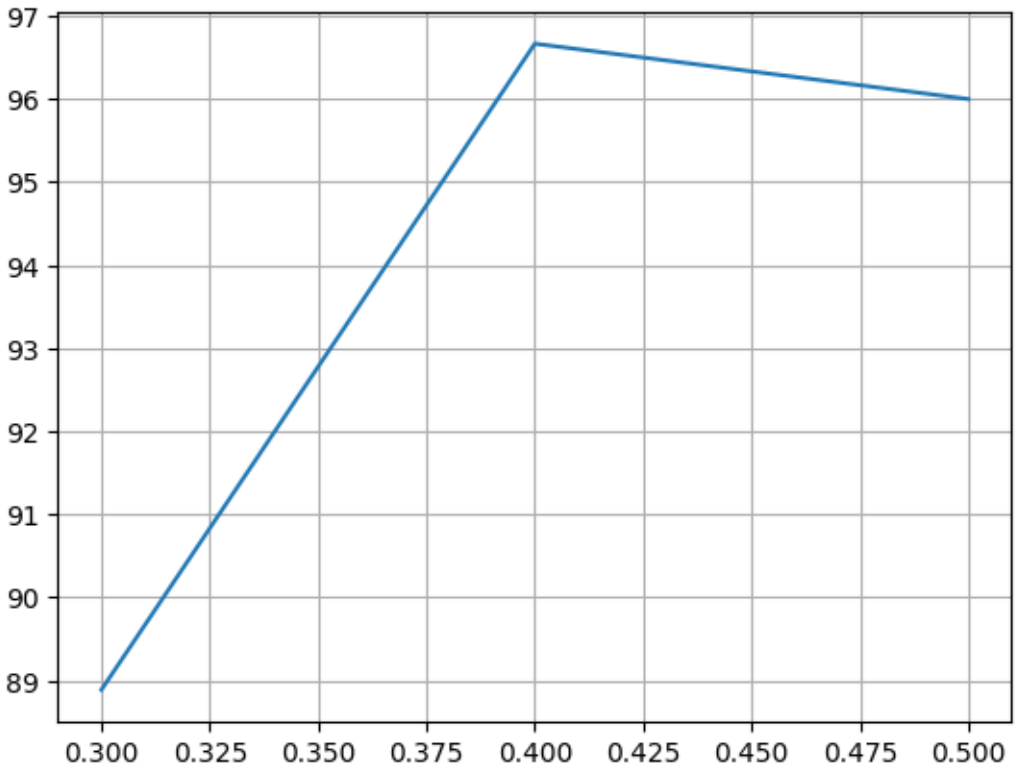



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      0.96      0.98        26
           1       0.88      1.00      0.94        22
           2       1.00      0.93      0.96        27

    accuracy                           0.96        75
   macro avg       0.96      0.96      0.96        75
weighted avg       0.96      0.96      0.96        75

Train-test split: 0.5
value: alpha: 1.0 binarize: 1.75 fit_prior: True
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      0.96      0.98        26
           1       0.88      1.00      0.94        22
           2       1.00      0.93      0.96        27

    accuracy                           0.96        75
   macro avg       0.96      0.96      0.96        75
weighted avg       0.96      0.96      0.96        75
```

[10]:
```
x_points = [float(key) for key in dict_bnb]
y_points = [i*100 for i in dict_bnb.values()]
plt.plot(x_points, y_points)
plt.grid(True)
plt.show()
```

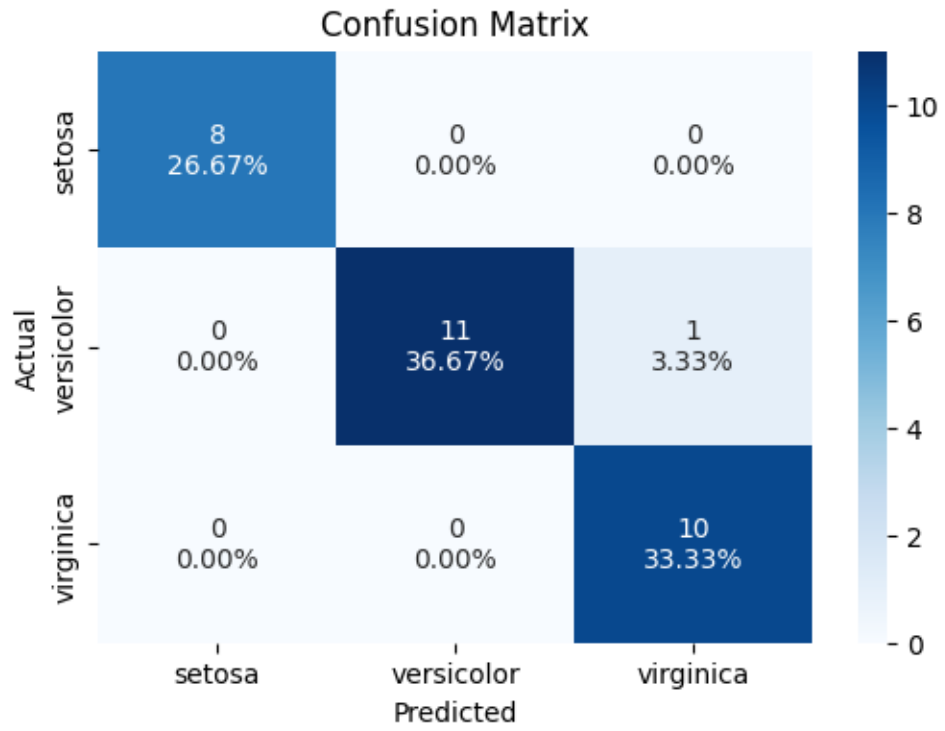


0.1 Classification using Multinomial Naive Bayes

[11]:
```
def FMultinomial(split, alpha_value = 1.0):
  from sklearn.naive_bayes import MultinomialNB
  from sklearn.metrics import accuracy_score
  from sklearn.model_selection import train_test_split
  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = split)
  classifier = MultinomialNB(alpha = alpha_value)
  classifier.fit(X_train, y_train)
  y_pred = classifier.predict(X_test)
  print("Train-test split: " + str(split))
  print("value: alpha: "+str(alpha_value))
  print("*********************************************")
  accuracy = accuracy_score(y_test, y_pred)
  if str(split) in dict_mnb:
    dict_mnb[str(split)] = max(accuracy, dict_mnb[str(split)])
    if str(split) == '0.3' and accuracy > dict_mnb[str(split)]:
      RocAucmnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
  else:
    dict_mnb[str(split)] = accuracy
    if str(split) == '0.3':
```

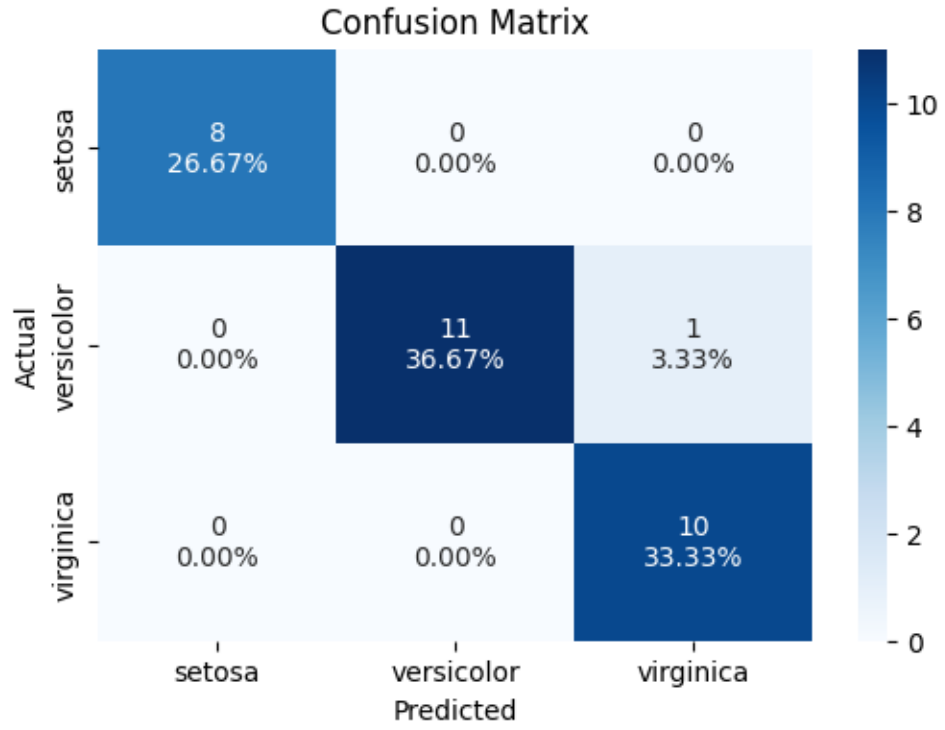
```
      RocAucmnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
  reports(y_test, y_pred)
  reports(y_test, y_pred)

## Train-Test split 0.2
FMultinomial(0.2)
FMultinomial(0.2, 1.8)
```

```
Train-test split: 0.2
value: alpha: 1.0
**********************************************
Confusion Matrix :
```

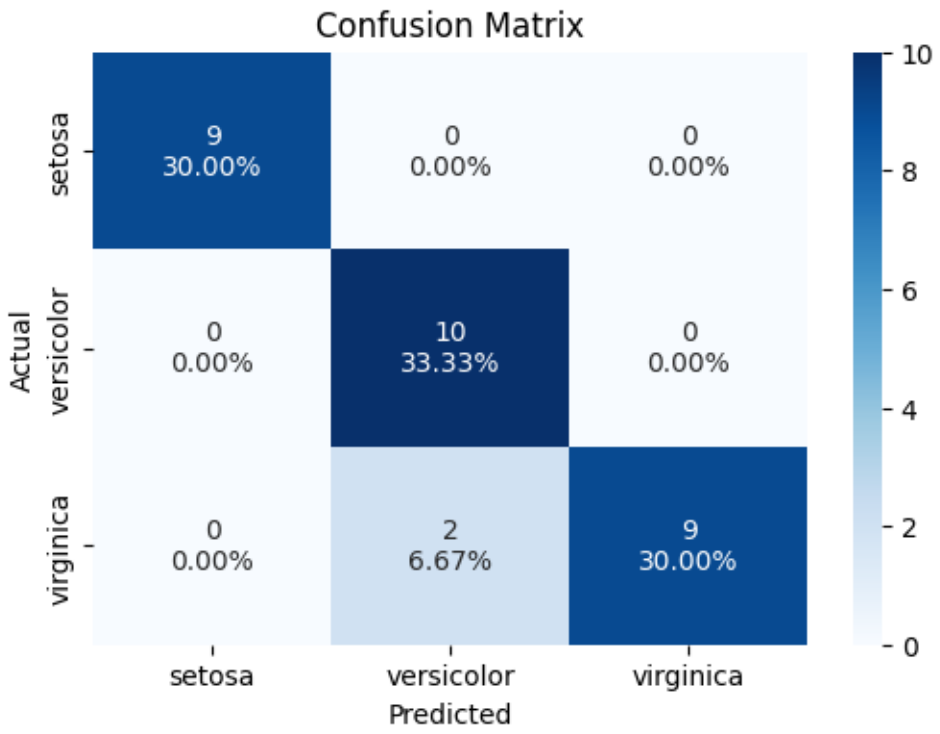



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         8
           1       1.00      0.92      0.96        12
           2       0.91      1.00      0.95        10

    accuracy                           0.97        30
   macro avg       0.97      0.97      0.97        30
```

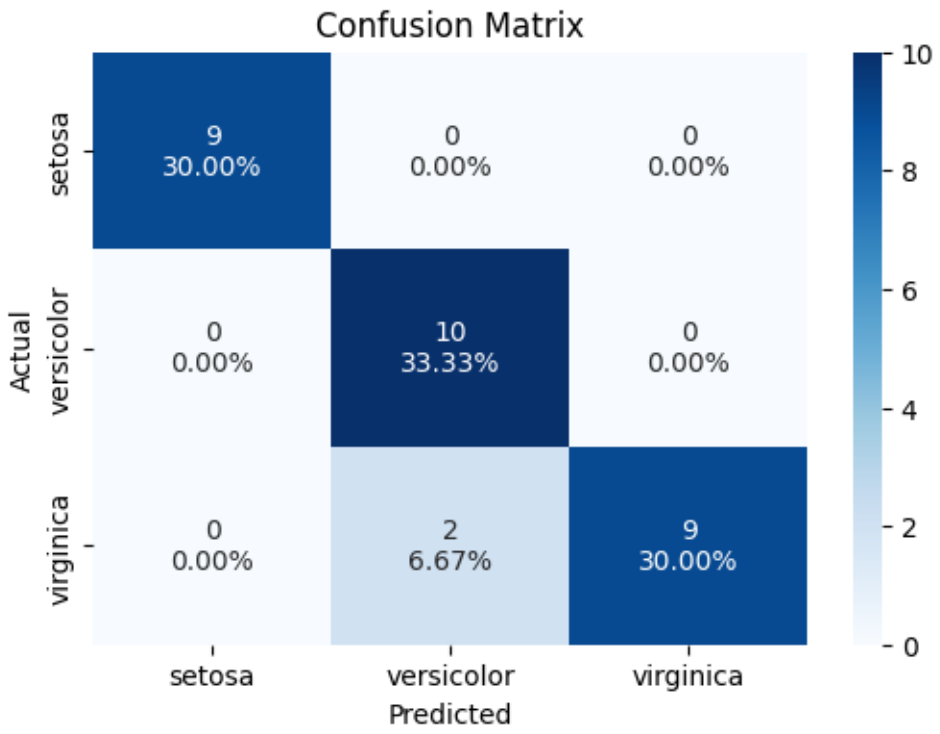
```
weighted avg       0.97      0.97      0.97        30

Confusion Matrix :
```




```
*********************************************
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         8
           1       1.00      0.92      0.96        12
           2       0.91      1.00      0.95        10

    accuracy                           0.97        30
   macro avg       0.97      0.97      0.97        30
weighted avg       0.97      0.97      0.97        30

Train-test split: 0.2
value: alpha: 1.8
*********************************************
Confusion Matrix :
```



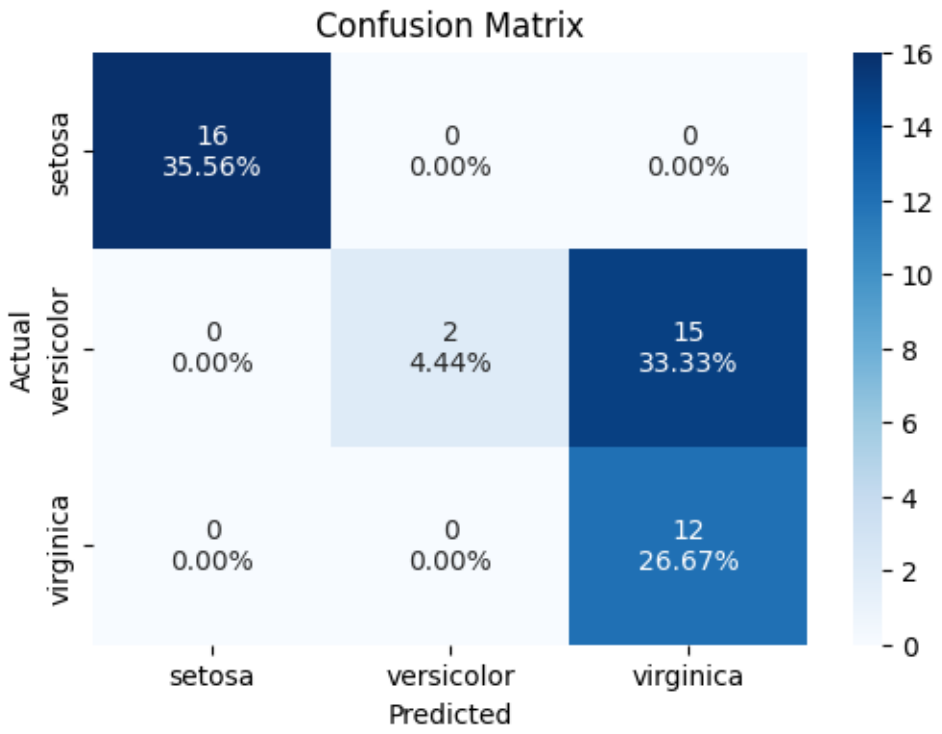

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         9
           1       0.83      1.00      0.91        10
           2       1.00      0.82      0.90        11

    accuracy                           0.93        30
   macro avg       0.94      0.94      0.94        30
weighted avg       0.94      0.93      0.93        30

Confusion Matrix :
```



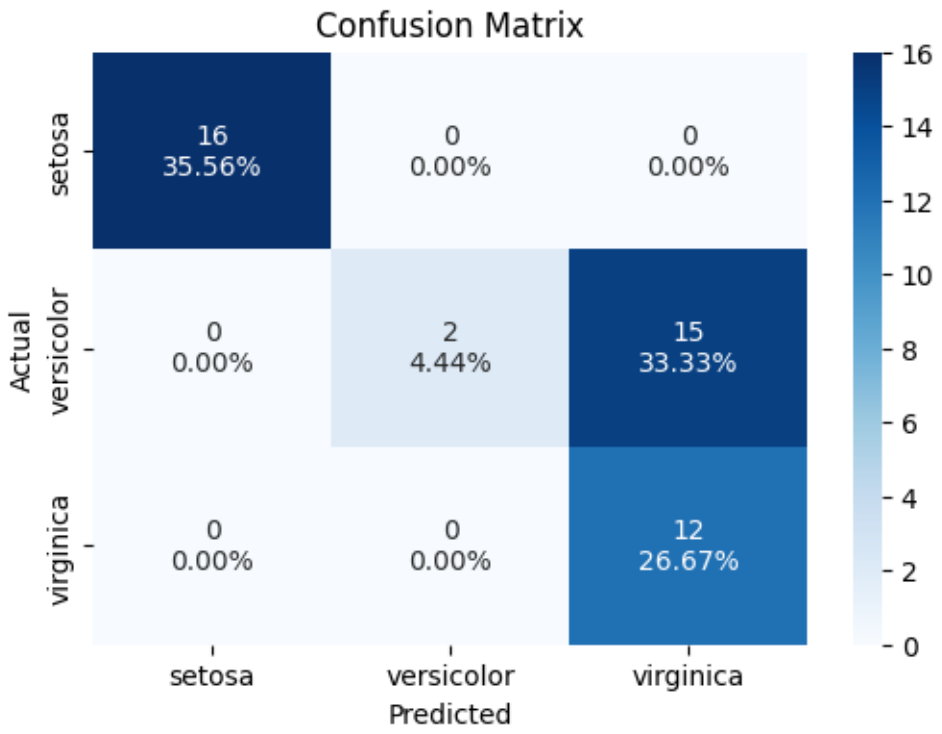

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         9
           1       0.83      1.00      0.91        10
           2       1.00      0.82      0.90        11

    accuracy                           0.93        30
   macro avg       0.94      0.94      0.94        30
weighted avg       0.94      0.93      0.93        30
```

[12]:
```
## Train-Test split 0.3
FMultinomial(0.3)
FMultinomial(0.3, 1.6)
```

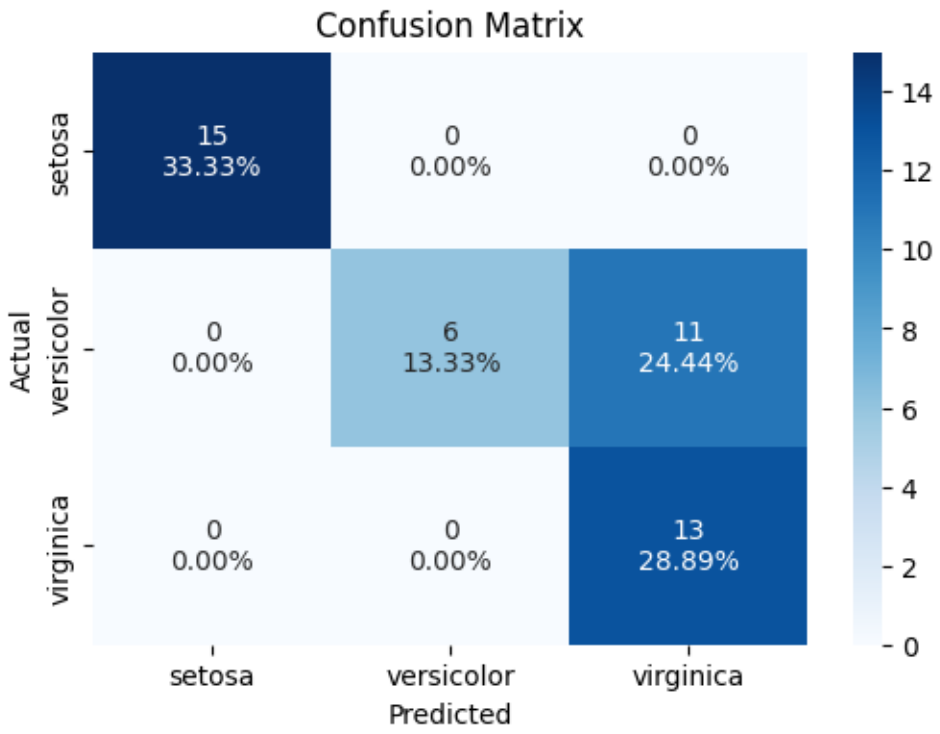
```
Train-test split: 0.3
value: alpha: 1.0
**********************************************
Confusion Matrix :
```



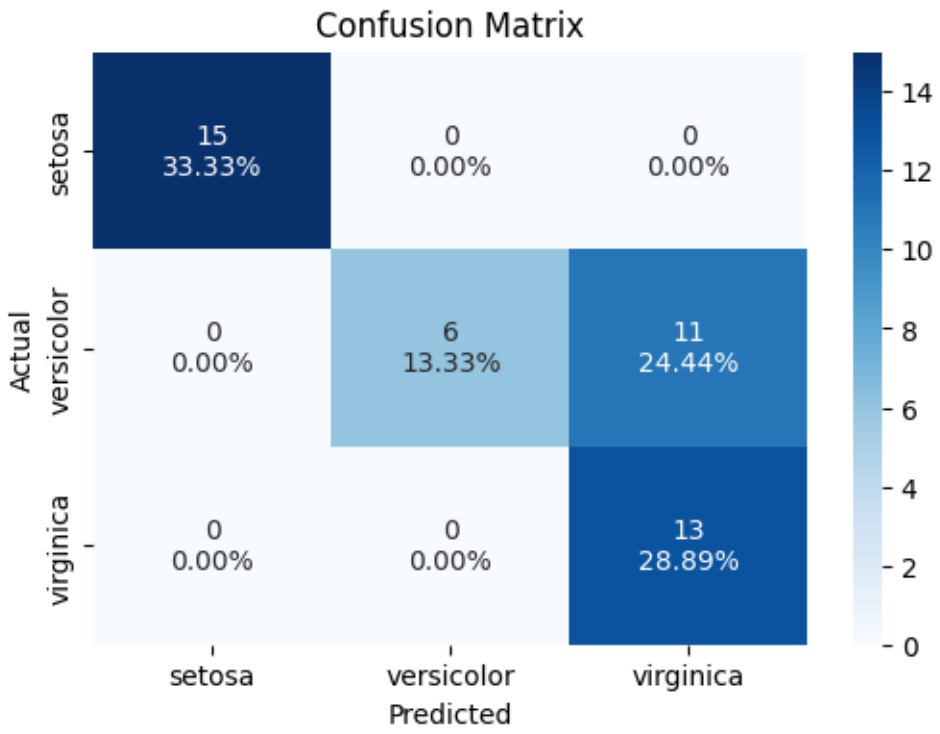

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        16
           1       1.00      0.12      0.21        17
           2       0.44      1.00      0.62        12

    accuracy                           0.67        45
   macro avg       0.81      0.71      0.61        45
weighted avg       0.85      0.67      0.60        45

Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        16
           1       1.00      0.12      0.21        17
           2       0.44      1.00      0.62        12

    accuracy                           0.67        45
   macro avg       0.81      0.71      0.61        45
weighted avg       0.85      0.67      0.60        45

Train-test split: 0.3
value: alpha: 1.6
**********************************************
Confusion Matrix :
```



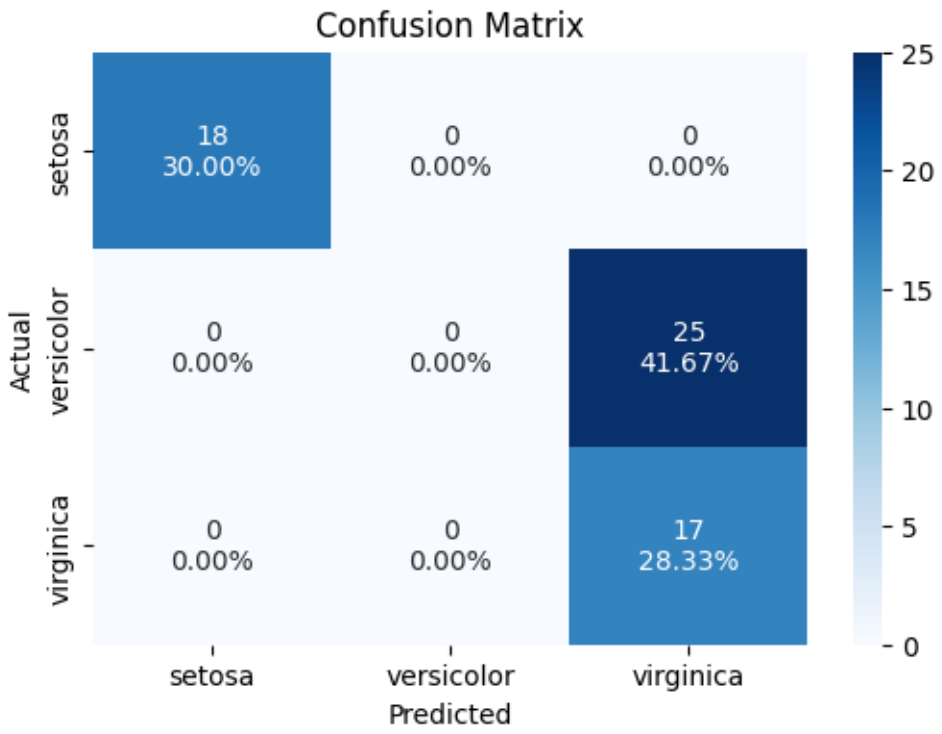

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        15
           1       1.00      0.35      0.52        17
           2       0.54      1.00      0.70        13

    accuracy                           0.76        45
   macro avg       0.85      0.78      0.74        45
weighted avg       0.87      0.76      0.73        45

Confusion Matrix :
```



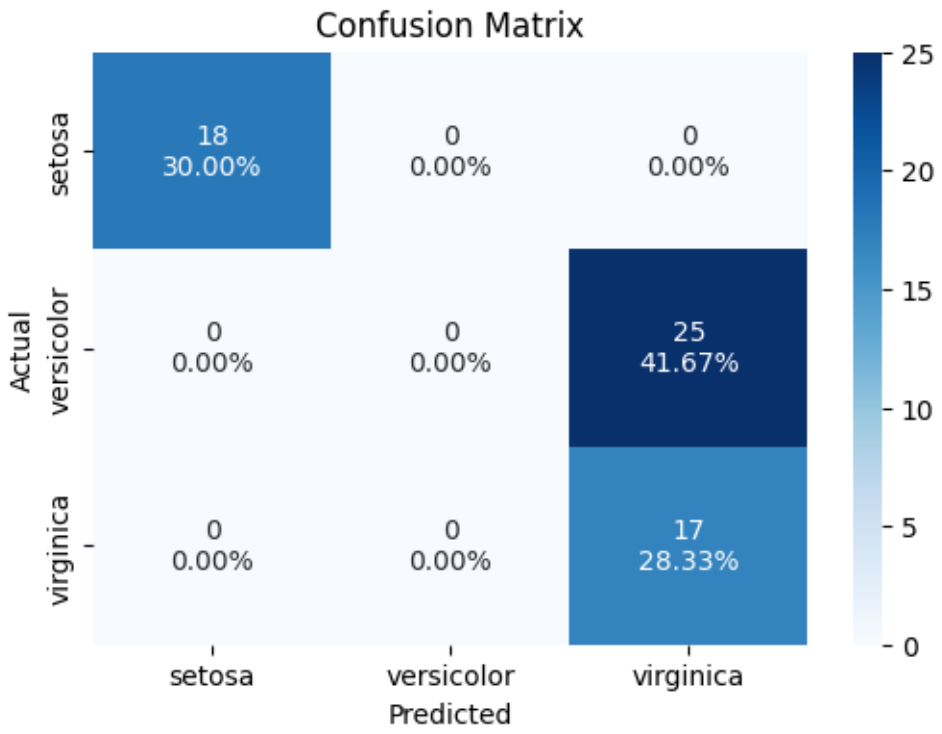

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        15
           1       1.00      0.35      0.52        17
           2       0.54      1.00      0.70        13

    accuracy                           0.76        45
   macro avg       0.85      0.78      0.74        45
weighted avg       0.87      0.76      0.73        45
```

[13]:
```
## Train-Test split 0.4
FMultinomial(0.4)
FMultinomial(0.4, 1.4)
```

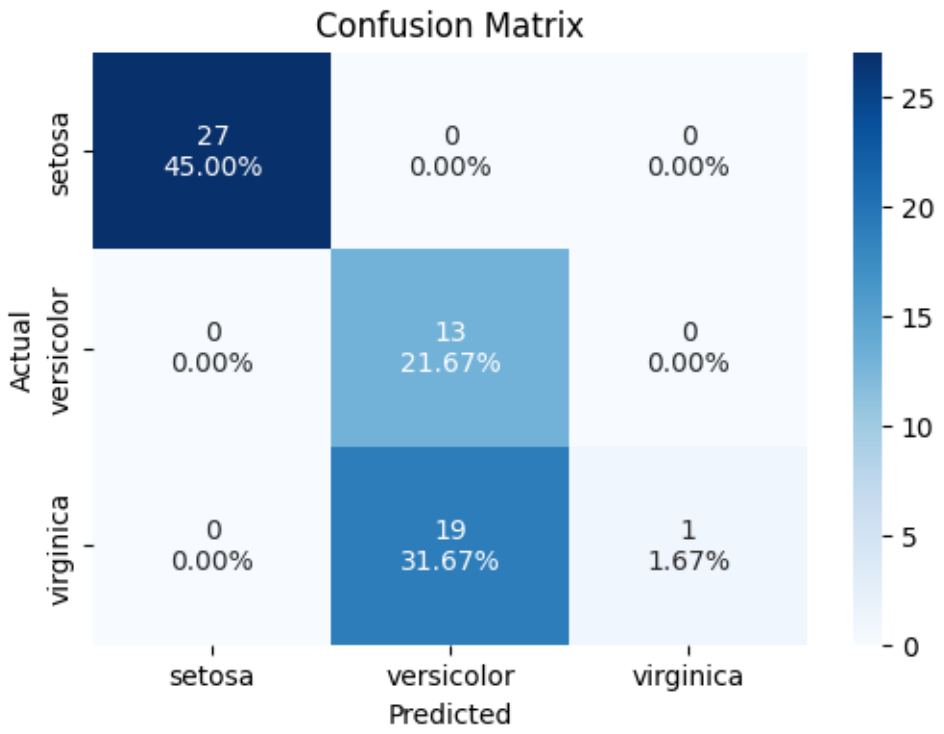
```
Train-test split: 0.4
value: alpha: 1.0
**********************************************
Confusion Matrix :
```



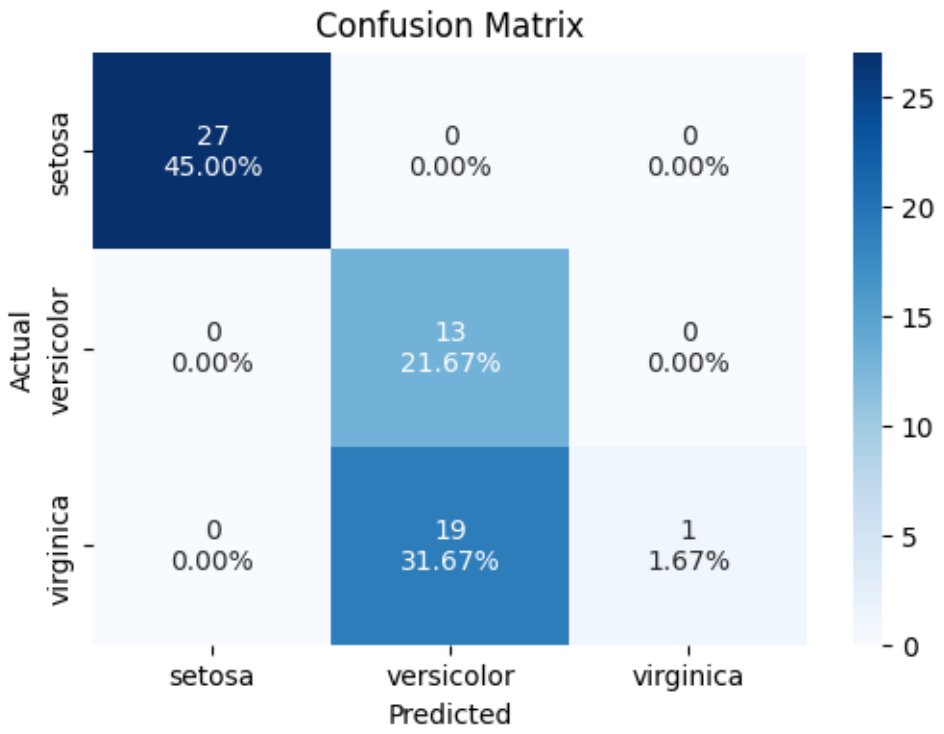

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        18
           1       0.00      0.00      0.00        25
           2       0.40      1.00      0.58        17

    accuracy                           0.58        60
   macro avg       0.47      0.67      0.53        60
weighted avg       0.41      0.58      0.46        60

Confusion Matrix :
```




```
*********************************************
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        18
           1       0.00      0.00      0.00        25
           2       0.40      1.00      0.58        17

    accuracy                           0.58        60
   macro avg       0.47      0.67      0.53        60
weighted avg       0.41      0.58      0.46        60

Train-test split: 0.4
value: alpha: 1.4
*********************************************
Confusion Matrix :
```



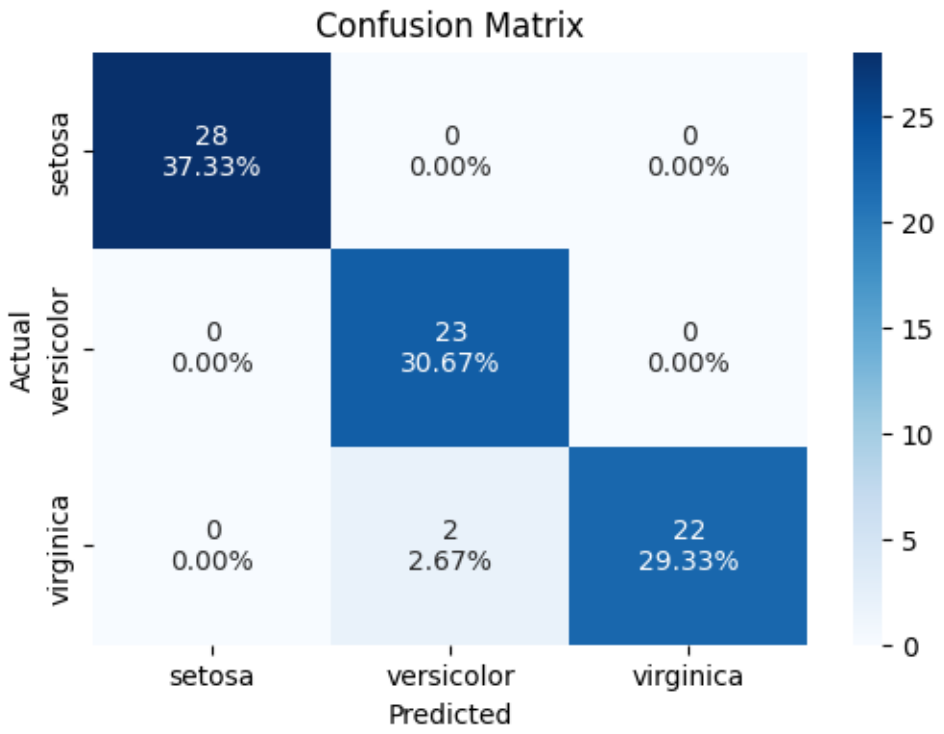

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        27
           1       0.41      1.00      0.58        13
           2       1.00      0.05      0.10        20

    accuracy                           0.68        60
   macro avg       0.80      0.68      0.56        60
weighted avg       0.87      0.68      0.61        60

Confusion Matrix :
```



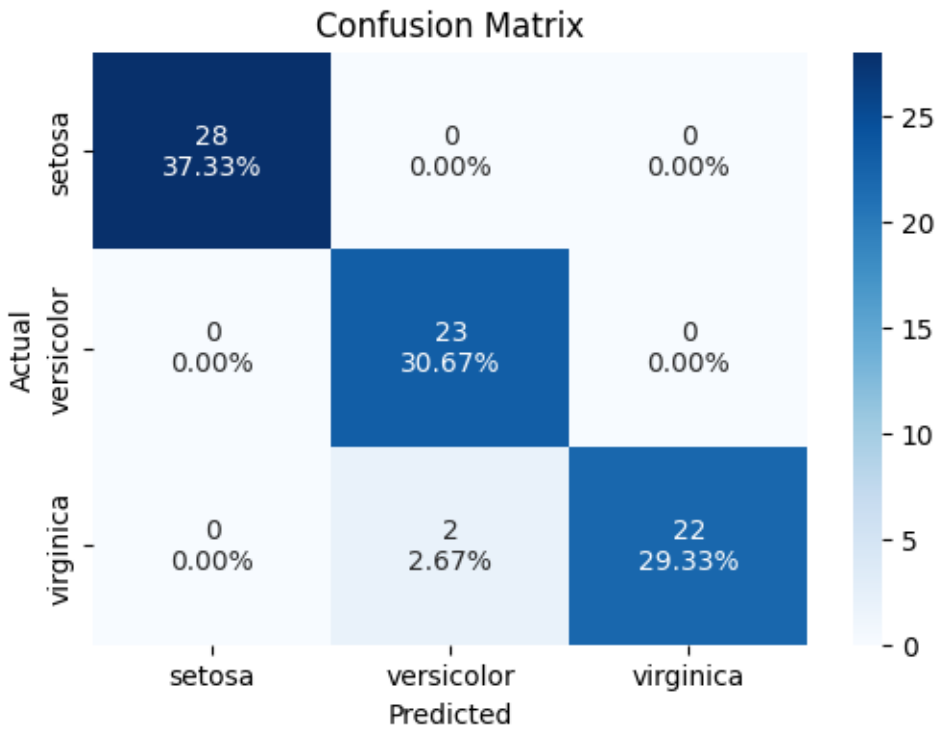

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        27
           1       0.41      1.00      0.58        13
           2       1.00      0.05      0.10        20

    accuracy                           0.68        60
   macro avg       0.80      0.68      0.56        60
weighted avg       0.87      0.68      0.61        60
```

[14]:
```
## Train-Test split 0.5
FMultinomial(0.5)
FMultinomial(0.5, 1.5)
```

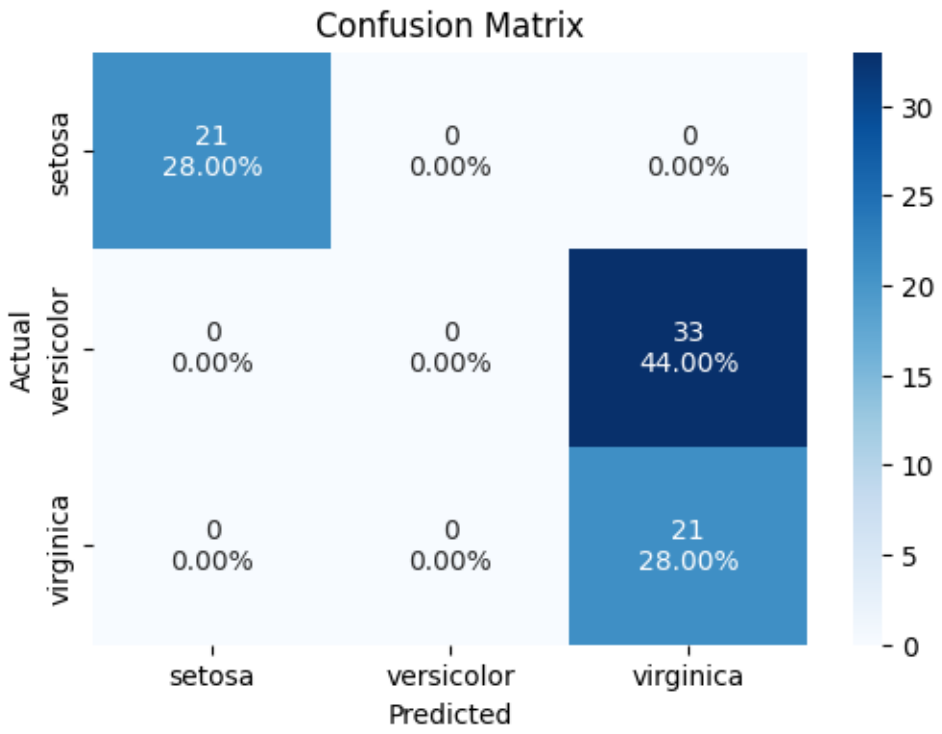
```
Train-test split: 0.5
value: alpha: 1.0
**********************************************
Confusion Matrix :
```



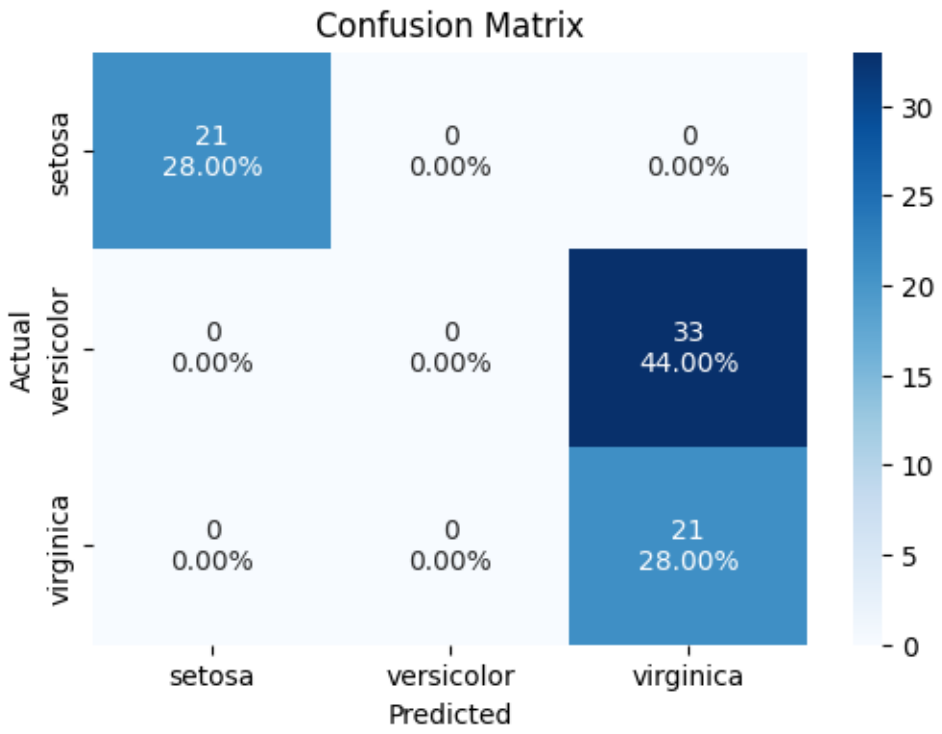

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        28
           1       0.92      1.00      0.96        23
           2       1.00      0.92      0.96        24

    accuracy                           0.97        75
   macro avg       0.97      0.97      0.97        75
weighted avg       0.98      0.97      0.97        75

Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        28
           1       0.92      1.00      0.96        23
           2       1.00      0.92      0.96        24

    accuracy                           0.97        75
   macro avg       0.97      0.97      0.97        75
weighted avg       0.98      0.97      0.97        75

Train-test split: 0.5
value: alpha: 1.5
**********************************************
Confusion Matrix :
```

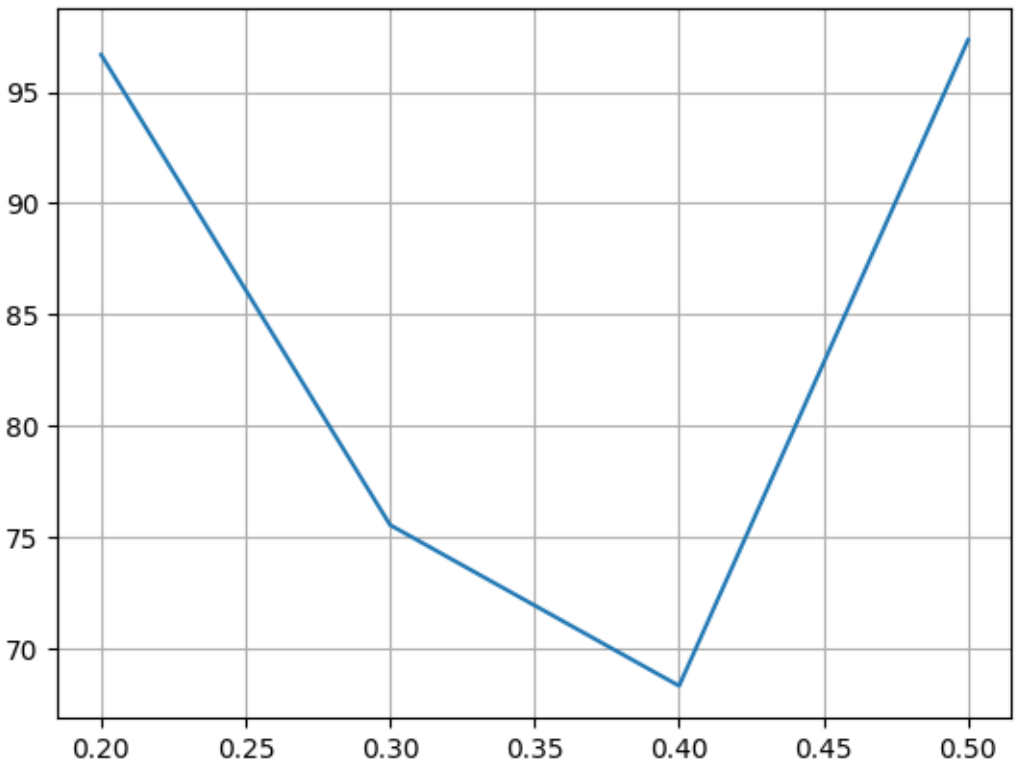



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        21
           1       0.00      0.00      0.00        33
           2       0.39      1.00      0.56        21

    accuracy                           0.56        75
   macro avg       0.46      0.67      0.52        75
weighted avg       0.39      0.56      0.44        75

Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        21
           1       0.00      0.00      0.00        33
           2       0.39      1.00      0.56        21

    accuracy                           0.56        75
   macro avg       0.46      0.67      0.52        75
weighted avg       0.39      0.56      0.44        75
```

[15]:
```
x_points = [float(key) for key in dict_mnb]
y_points = [i*100 for i in dict_mnb.values()]
plt.plot(x_points, y_points)
plt.grid(True)
plt.show()
```

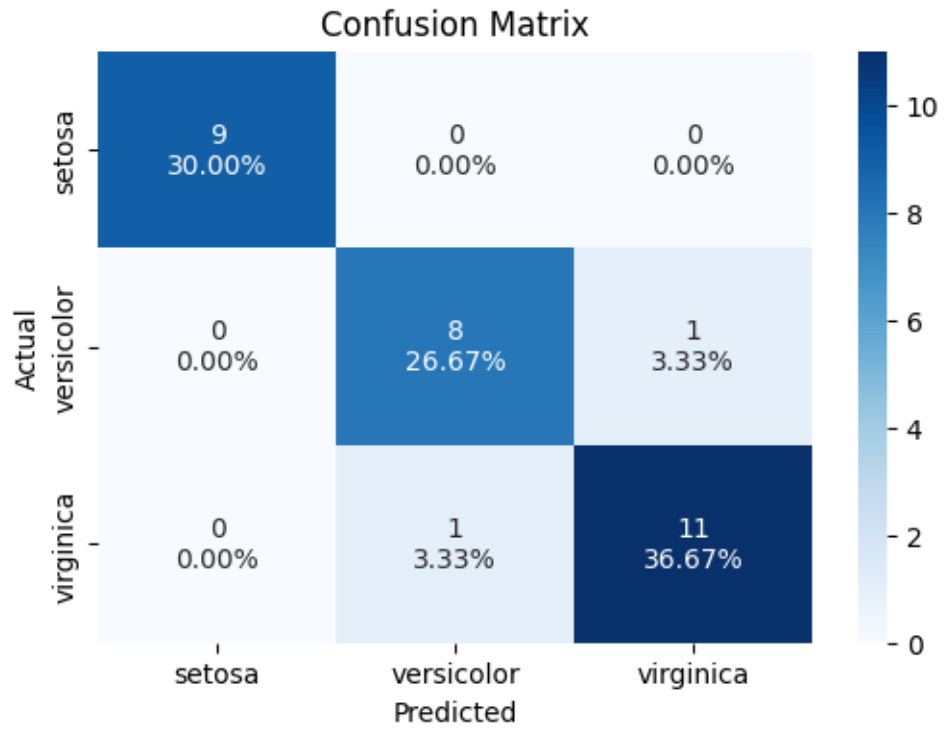


0.1.1 Classification using Guassian Naive Bayes

[16]:
```
def FGaussian(split):
  from sklearn.naive_bayes import GaussianNB
  from sklearn.model_selection import train_test_split
  from sklearn.metrics import accuracy_score
  from sklearn.preprocessing import StandardScaler
  scaler = StandardScaler()
  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = split,␣
 ↪random_state=44)
  scaler.fit_transform(X_train)
  scaler.transform(X_test)
  classifier = GaussianNB()
  classifier.fit(X_train, y_train)
  y_pred = classifier.predict(X_test)
  print("Train-test split: " + str(split))
  print("**********************************************")
  reports(y_test, y_pred)
  accuracy = accuracy_score(y_test, y_pred)
  if(str(split) in dict_gnb):
    dict_gnb[str(split)] = max(accuracy, dict_gnb[str(split)])
    if(str(split) == '0.3' and accuracy > dict_bnb[str(split)]):
```

```
      RocAucgnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
  else:
    dict_gnb[str(split)] = accuracy
    RocAucgnb['max'] = {'y_test': y_test, 'y_pred': y_pred}

## Train-Test split 0.2
FGaussian(0.2)
# 94, 97, 94, 96,
```

```
Train-test split: 0.2
**********************************************
Confusion Matrix :
```

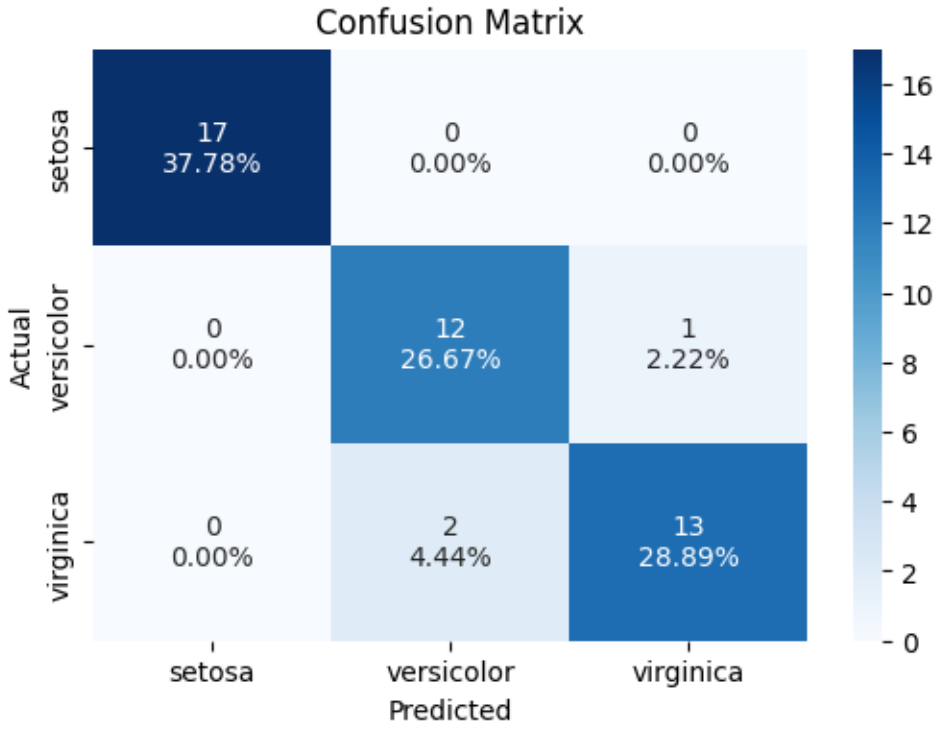



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         9
           1       0.89      0.89      0.89         9
           2       0.92      0.92      0.92        12

    accuracy                           0.93        30
   macro avg       0.94      0.94      0.94        30
```

```
weighted avg       0.93      0.93      0.93        30
```

[17]:
```
## Train-Test split 0.3
FGaussian(0.3)
```

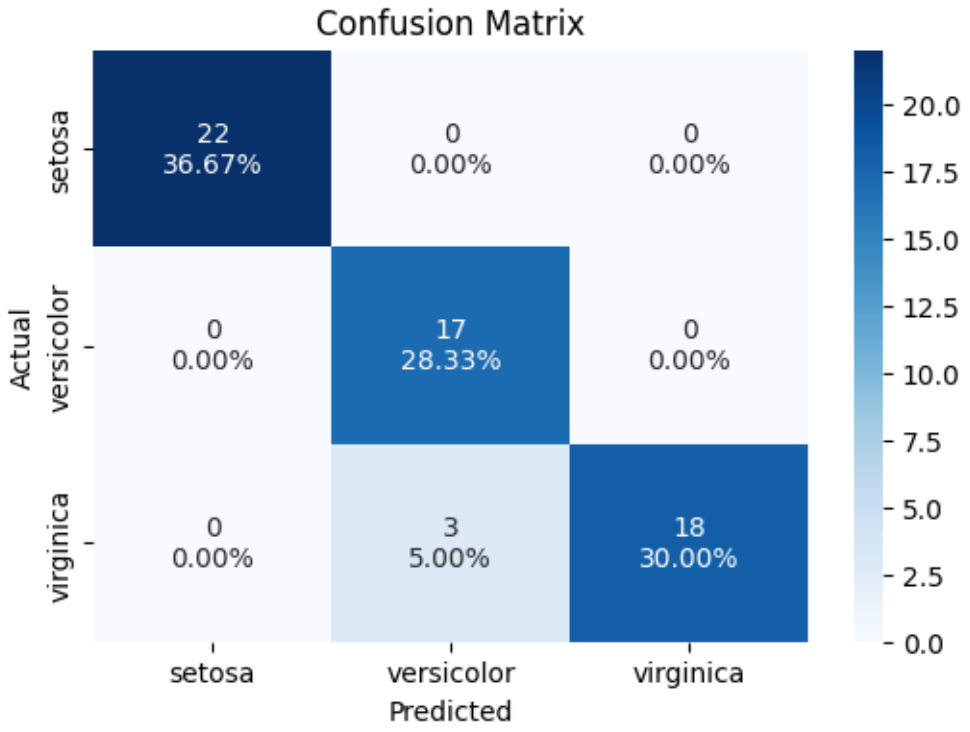
```
Train-test split: 0.3
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        17
           1       0.86      0.92      0.89        13
           2       0.93      0.87      0.90        15

    accuracy                           0.93        45
   macro avg       0.93      0.93      0.93        45
weighted avg       0.93      0.93      0.93        45
```

[18]:
```
## Train-Test split 0.4
FGaussian(0.4)
```

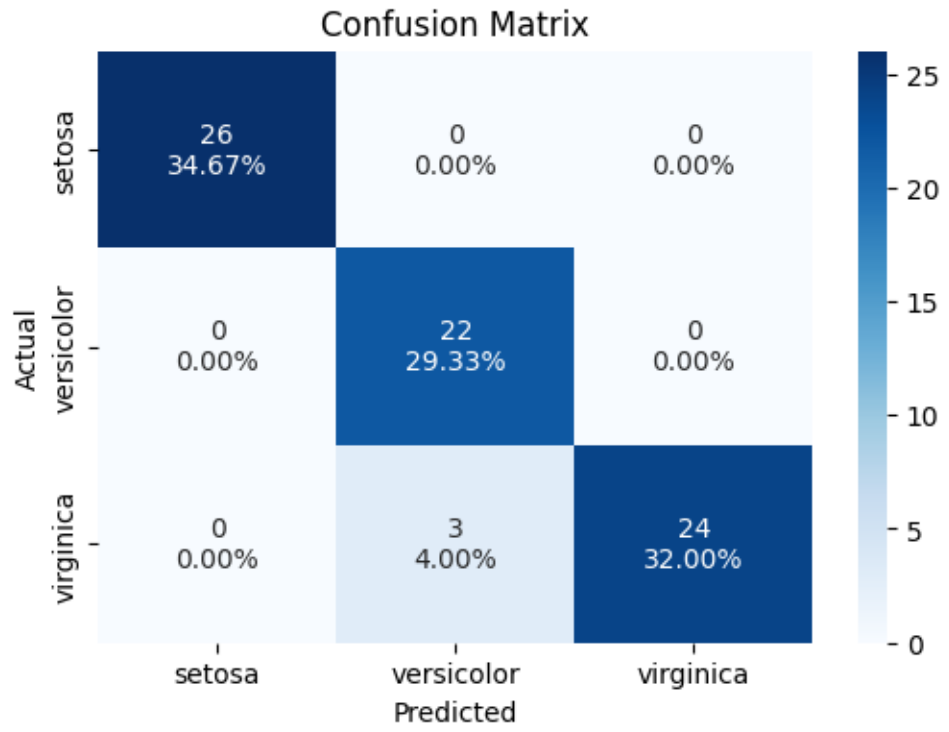
```
Train-test split: 0.4
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        22
           1       0.85      1.00      0.92        17
           2       1.00      0.86      0.92        21

    accuracy                           0.95        60
   macro avg       0.95      0.95      0.95        60
weighted avg       0.96      0.95      0.95        60
```

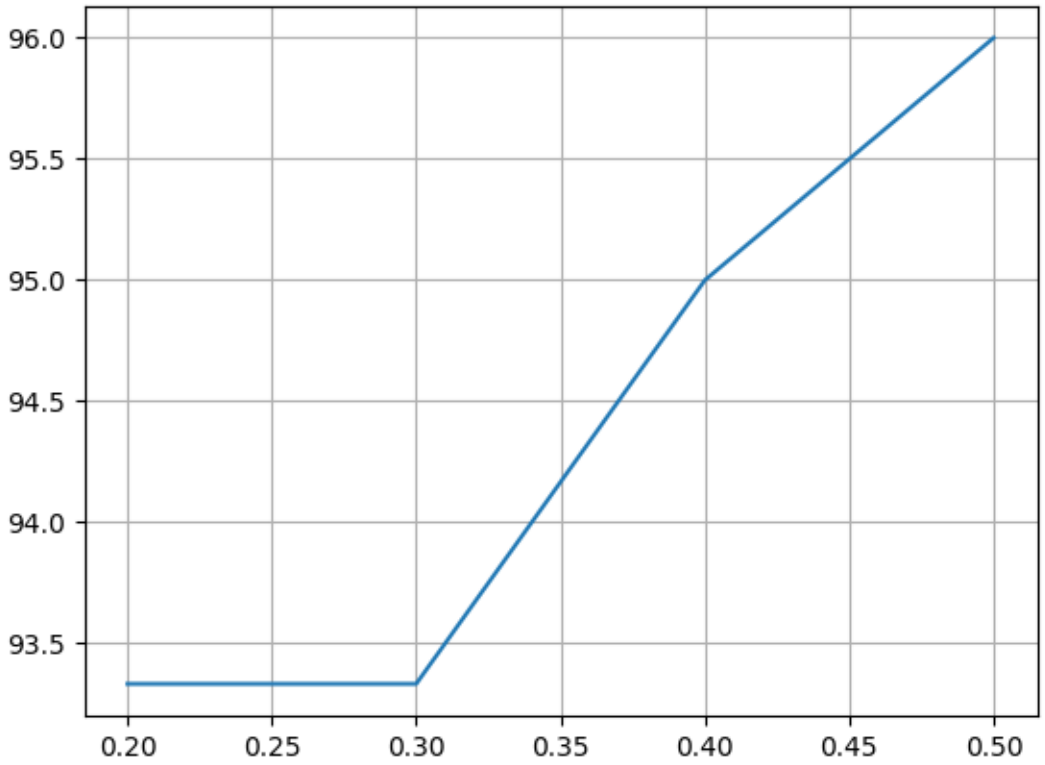
[19]:
```
## Train-Test split 0.5
FGaussian(0.5)
```

```
Train-test split: 0.5
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        26
           1       0.88      1.00      0.94        22
           2       1.00      0.89      0.94        27

    accuracy                           0.96        75
   macro avg       0.96      0.96      0.96        75
weighted avg       0.96      0.96      0.96        75
```

[20]:
```
x_points = [float(key) for key in dict_gnb]
y_points = [i*100 for i in dict_gnb.values()]
plt.plot(x_points, y_points)
plt.grid(True)
plt.show()
```



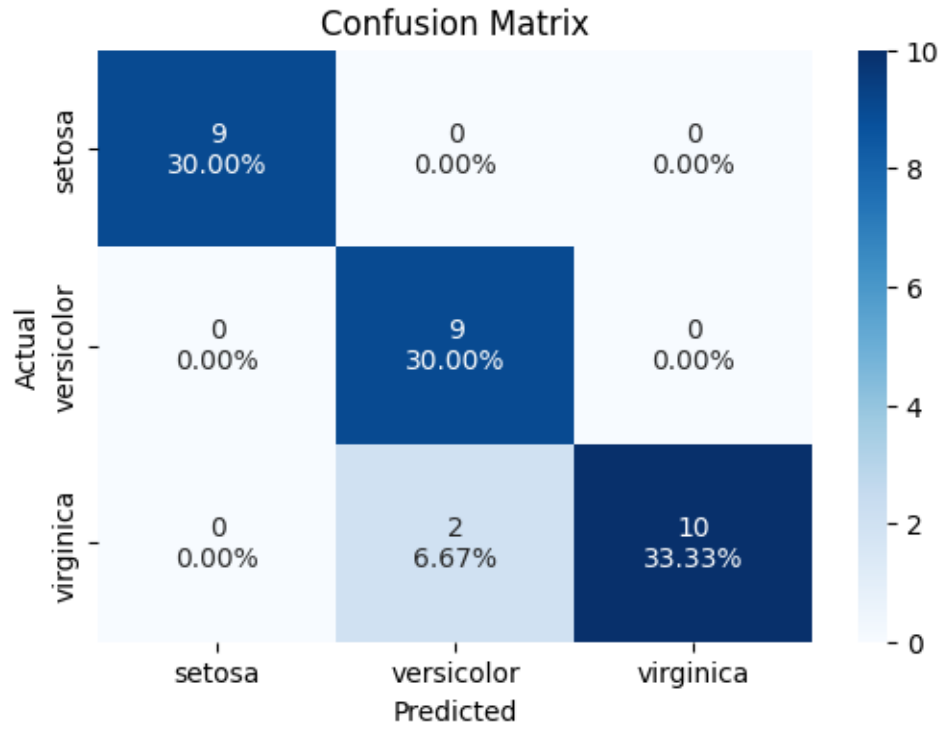
0.1.2 Classification using Decision Tree

[21]:
```
def decision_tree(split, criterion_value):
  from sklearn.model_selection import train_test_split
  from sklearn.tree import DecisionTreeClassifier
  from sklearn import tree
  from sklearn.metrics import accuracy_score
  from sklearn.preprocessing import StandardScaler
  scaler = StandardScaler()
  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = split,␣
 ↪random_state=44)
  scaler.fit_transform(X_train)
  scaler.transform(X_test)

  classifier = DecisionTreeClassifier(criterion = criterion_value)
  classifier.fit(X_train, y_train)
  y_pred = classifier.predict(X_test)
  print("Train-test split: " + str(split))
  print("Value: Entropy: " + criterion_value)
  print("********************************************")
  reports(y_test, y_pred)
  accuracy = accuracy_score(y_test, y_pred)
```

```
  if(str(split) in dict_dtr):
    dict_dtr[str(split)] = max(accuracy, dict_dtr[str(split)])
    if(str(split) == '0.3' and accuracy > dict_bnb[str(split)]):
      RocAucdtr['max'] = {'y_test': y_test, 'y_pred': y_pred}
  else:
    dict_dtr[str(split)] = accuracy
    if(str(split) == '0.3'):
      RocAucdtr['max'] = {'y_test': y_test, 'y_pred': y_pred}

  fig = plt.figure(figsize=(12,8))
  _ = tree.plot_tree(classifier,
                     feature_names=column_names,
                     class_names=['outcome1', 'outcome2', 'outcome3'],
                     filled=True)
```

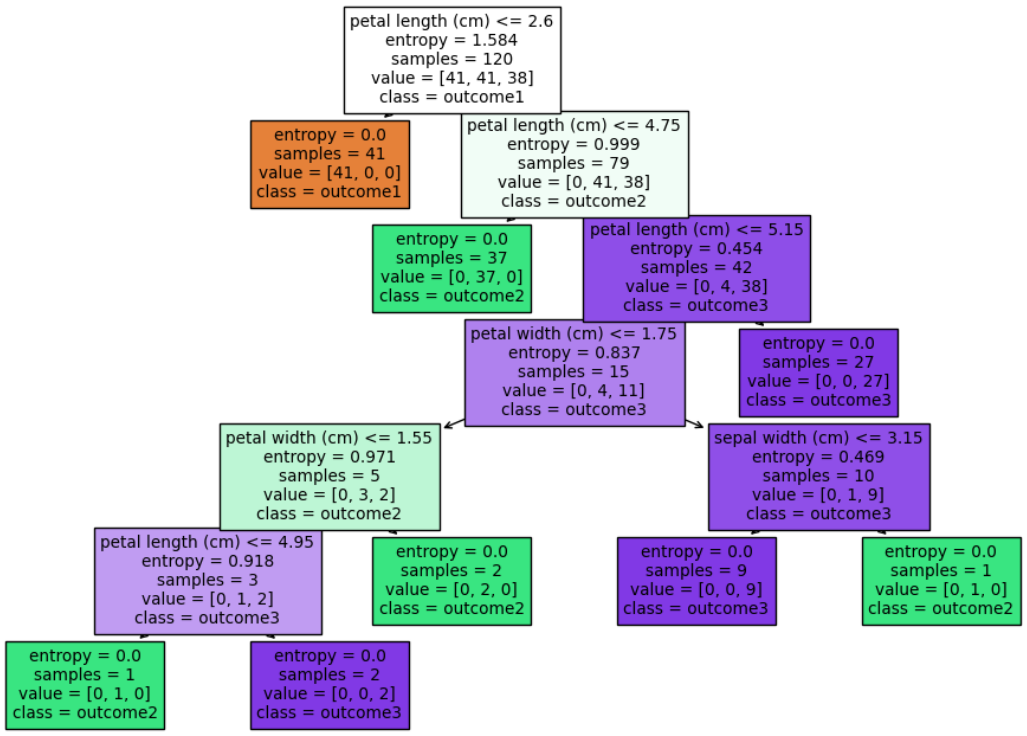
[22]:
```
decision_tree(0.2, 'entropy')
```

```
Train-test split: 0.2
Value: Entropy: entropy
*********************************************
Confusion Matrix :
```



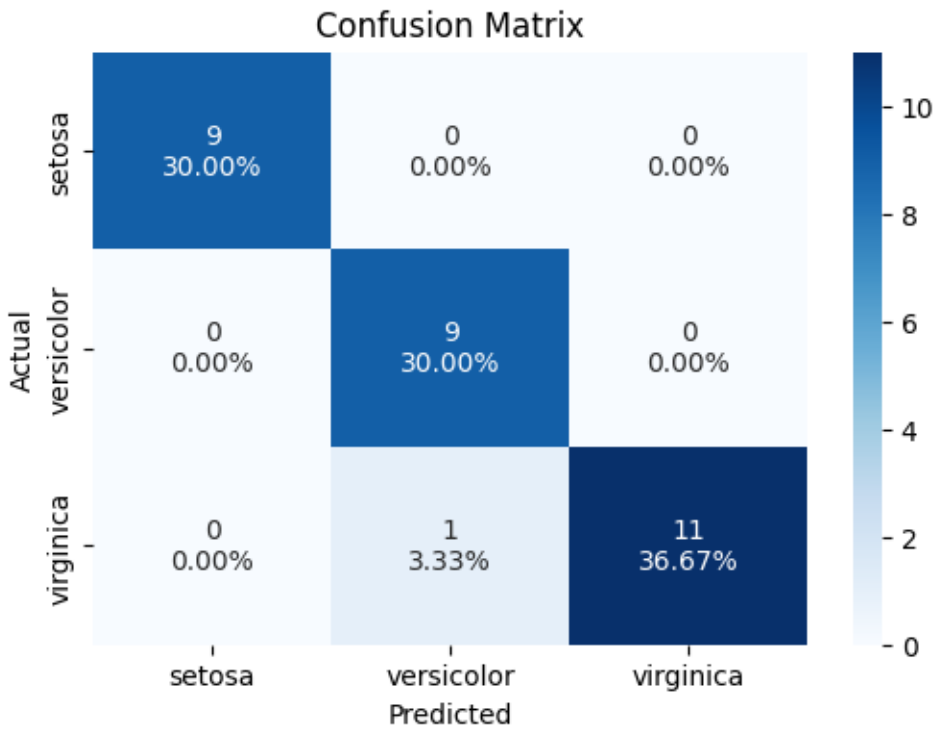

```
*********************************************
```

```
*******************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         9
           1       0.82      1.00      0.90         9
           2       1.00      0.83      0.91        12

    accuracy                           0.93        30
   macro avg       0.94      0.94      0.94        30
weighted avg       0.95      0.93      0.93        30
```

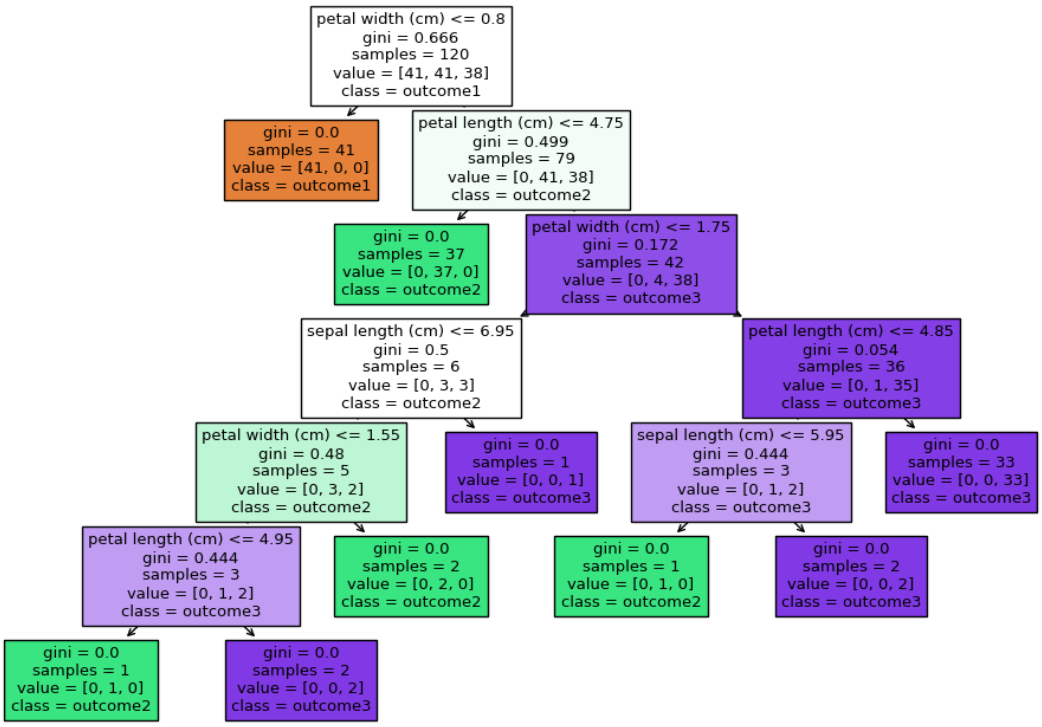


[23]:
```
decision_tree(0.2, 'gini')
```

```
Train-test split: 0.2
Value: Entropy: gini
*******************************************
Confusion Matrix :
```

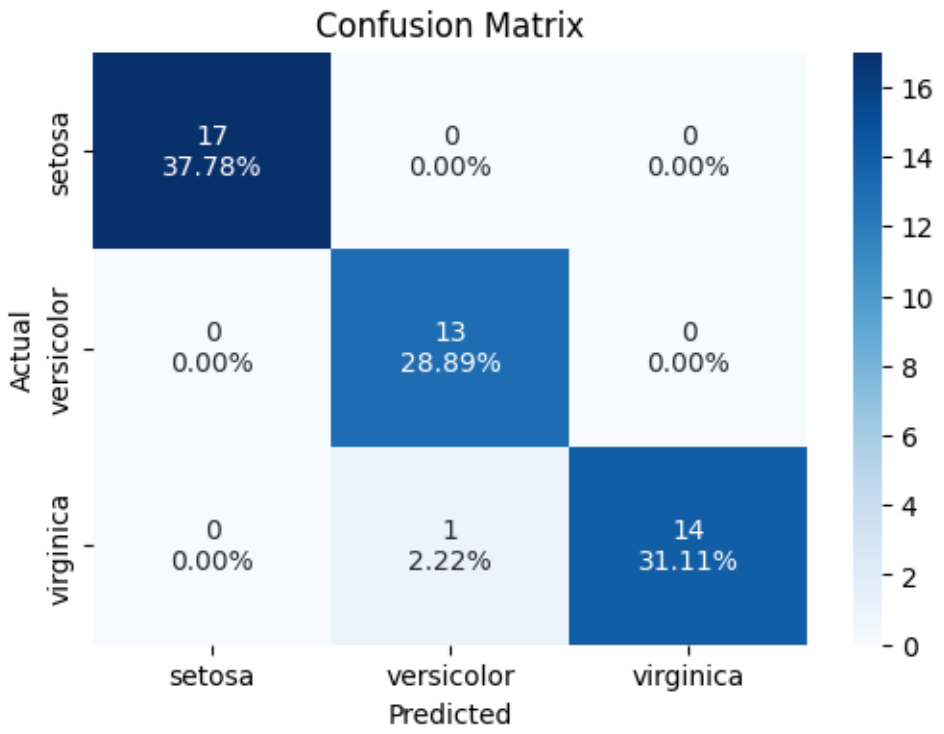



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         9
           1       0.90      1.00      0.95         9
           2       1.00      0.92      0.96        12

    accuracy                           0.97        30
   macro avg       0.97      0.97      0.97        30
weighted avg       0.97      0.97      0.97        30
```

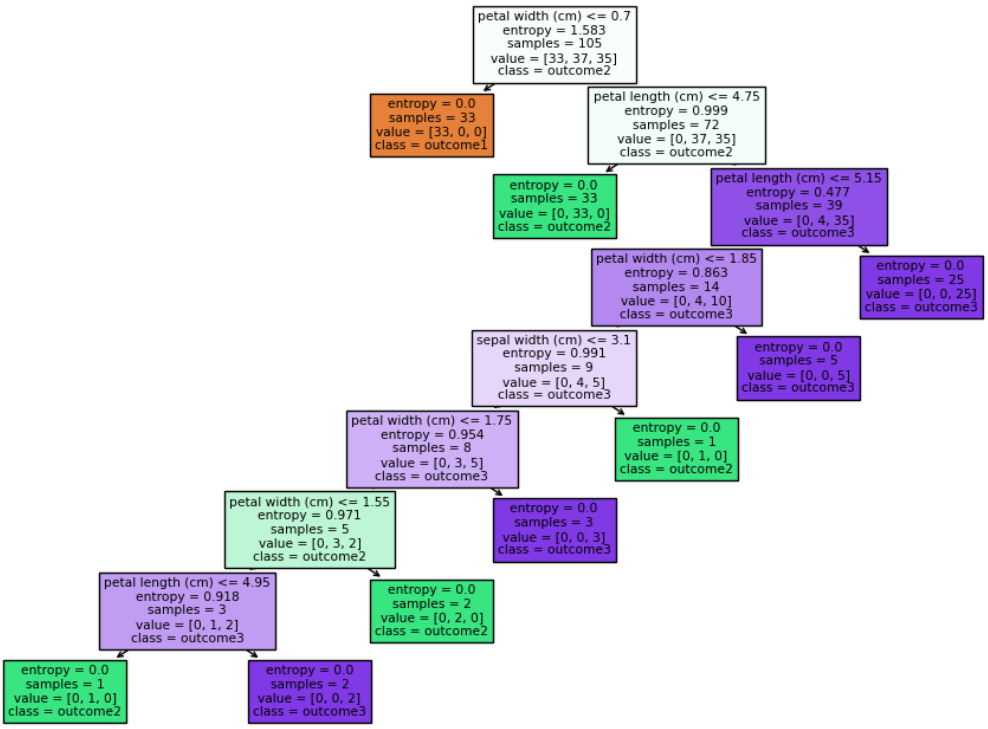


[24]:
```
decision_tree(0.3, 'entropy')
```

```
Train-test split: 0.3
Value: Entropy: entropy
*********************************************
Confusion Matrix :
```

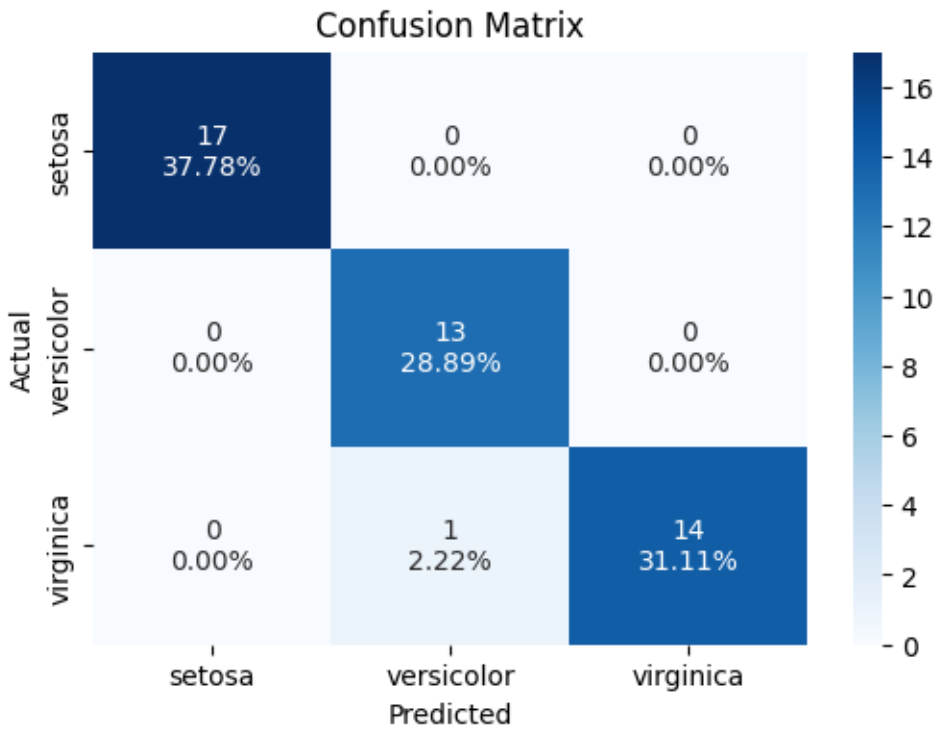



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        17
           1       0.93      1.00      0.96        13
           2       1.00      0.93      0.97        15

    accuracy                           0.98        45
   macro avg       0.98      0.98      0.98        45
weighted avg       0.98      0.98      0.98        45
```

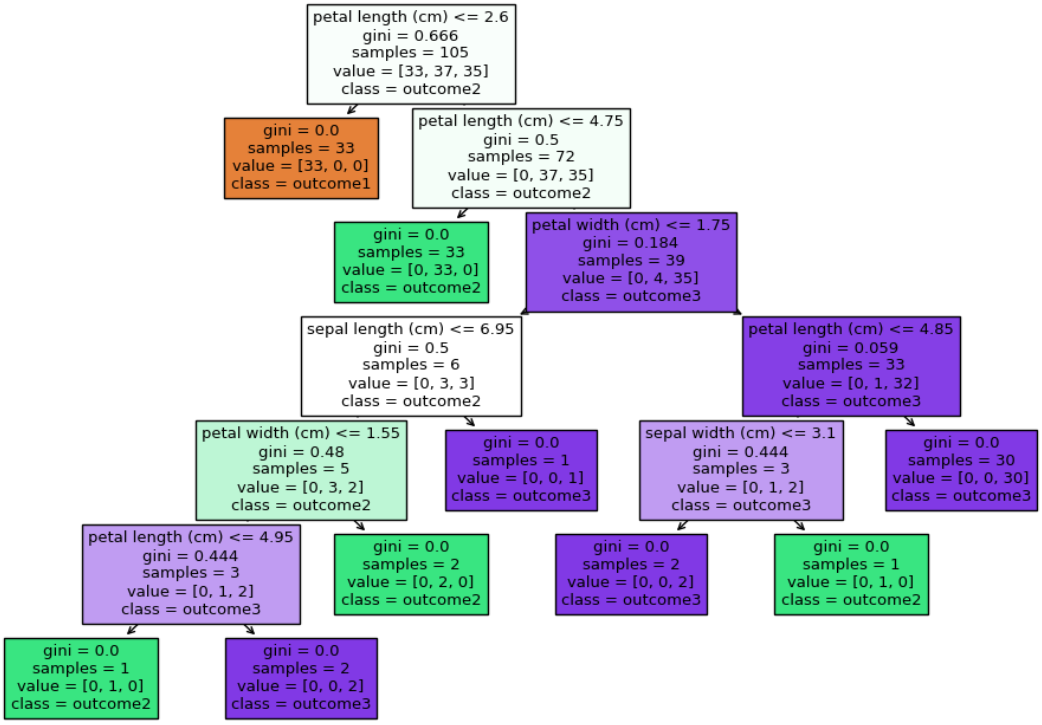


[25]:
```
decision_tree(0.3, 'gini')
```

```
Train-test split: 0.3
Value: Entropy: gini
*********************************************
Confusion Matrix :
```

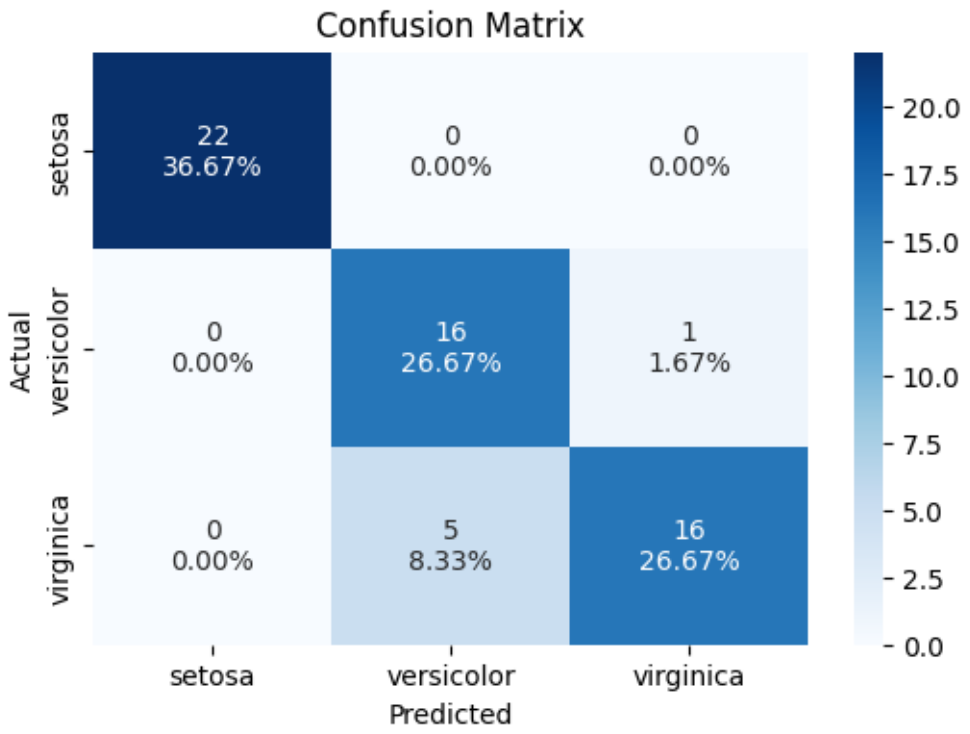



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        17
           1       0.93      1.00      0.96        13
           2       1.00      0.93      0.97        15

    accuracy                           0.98        45
   macro avg       0.98      0.98      0.98        45
weighted avg       0.98      0.98      0.98        45
```

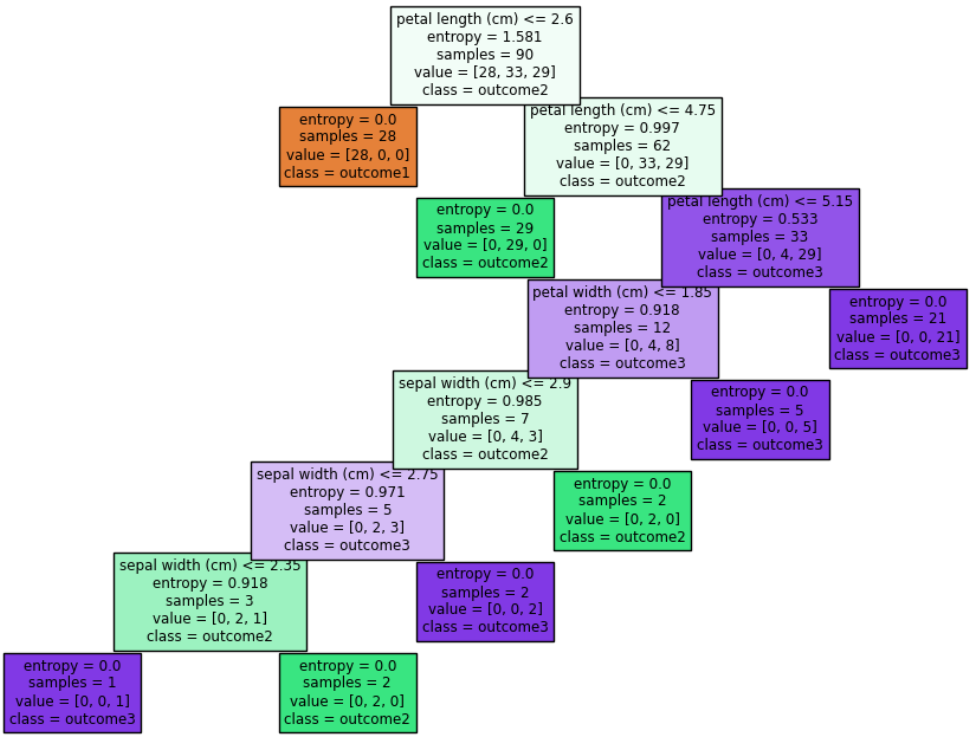


[26]:
```
decision_tree(0.4, 'entropy')
```

```
Train-test split: 0.4
Value: Entropy: entropy
*********************************************
Confusion Matrix :
```

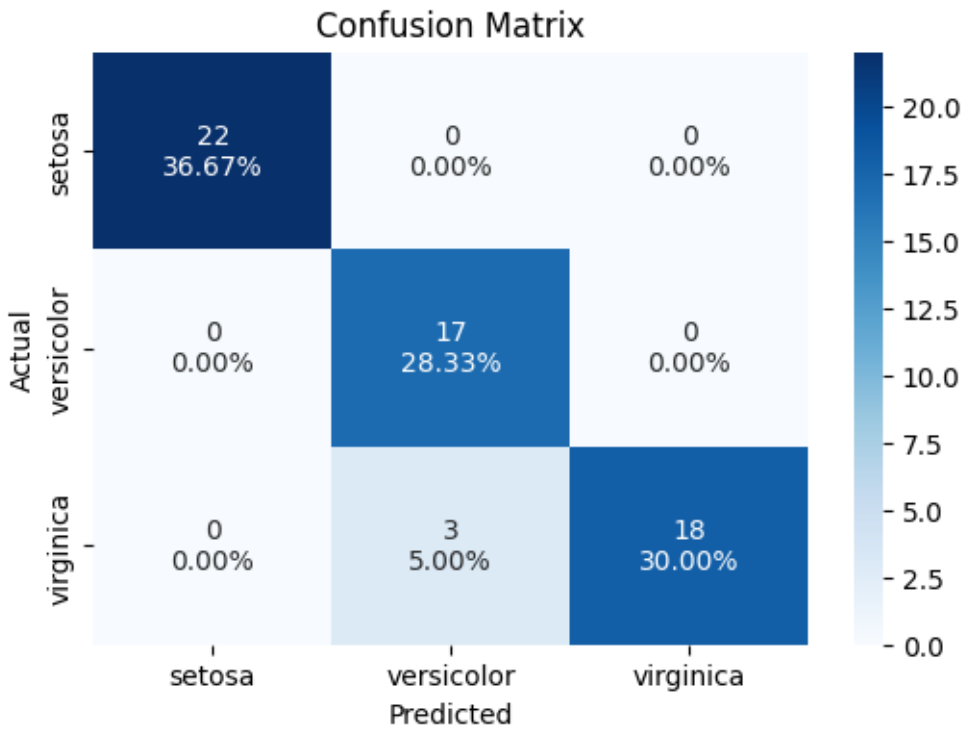



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        22
           1       0.76      0.94      0.84        17
           2       0.94      0.76      0.84        21

    accuracy                           0.90        60
   macro avg       0.90      0.90      0.89        60
weighted avg       0.91      0.90      0.90        60
```

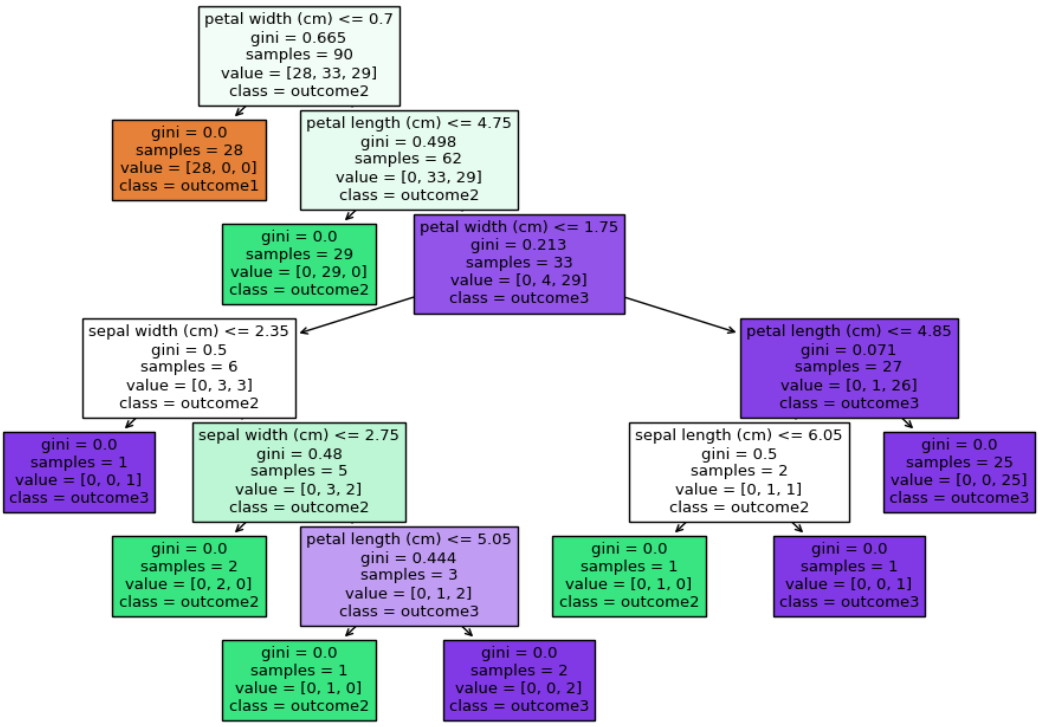


[27]:
```
decision_tree(0.4, 'gini')
```

```
Train-test split: 0.4
Value: Entropy: gini
*********************************************
Confusion Matrix :
```

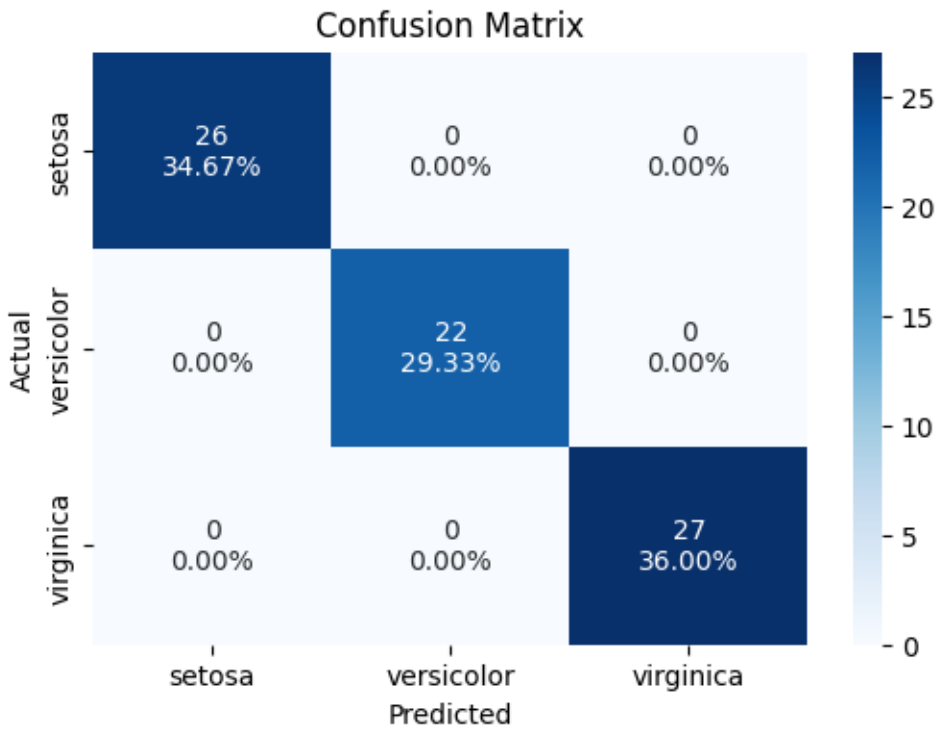



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        22
           1       0.85      1.00      0.92        17
           2       1.00      0.86      0.92        21

    accuracy                           0.95        60
   macro avg       0.95      0.95      0.95        60
weighted avg       0.96      0.95      0.95        60
```

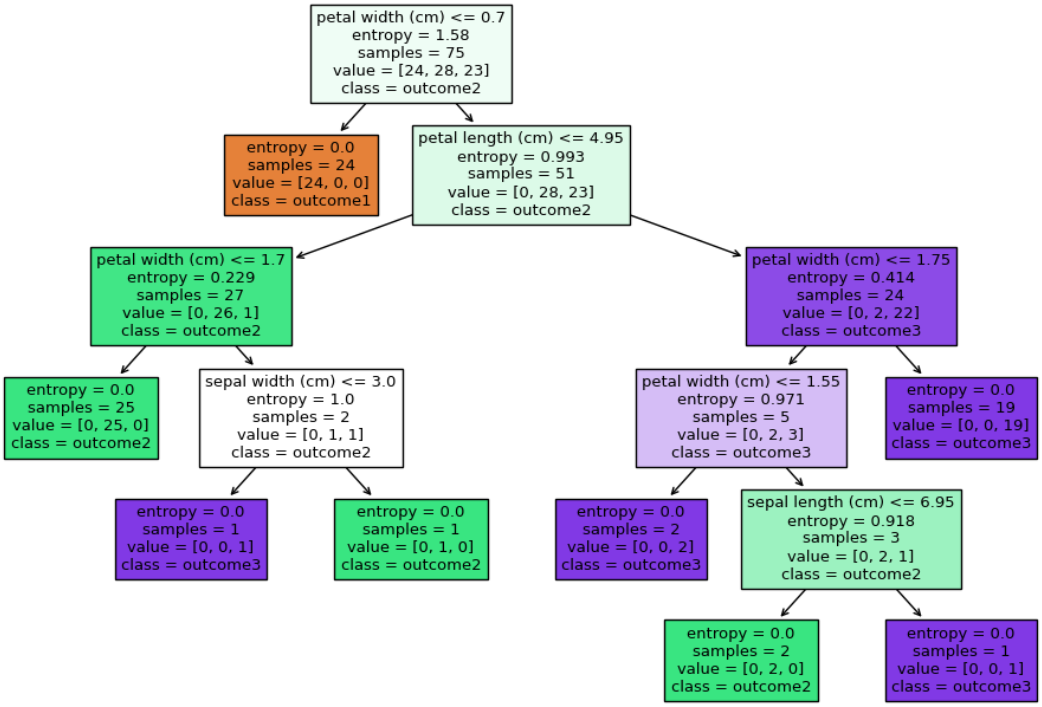


[28]:
```
decision_tree(0.5, 'entropy')
```

```
Train-test split: 0.5
Value: Entropy: entropy
*********************************************
Confusion Matrix :
```

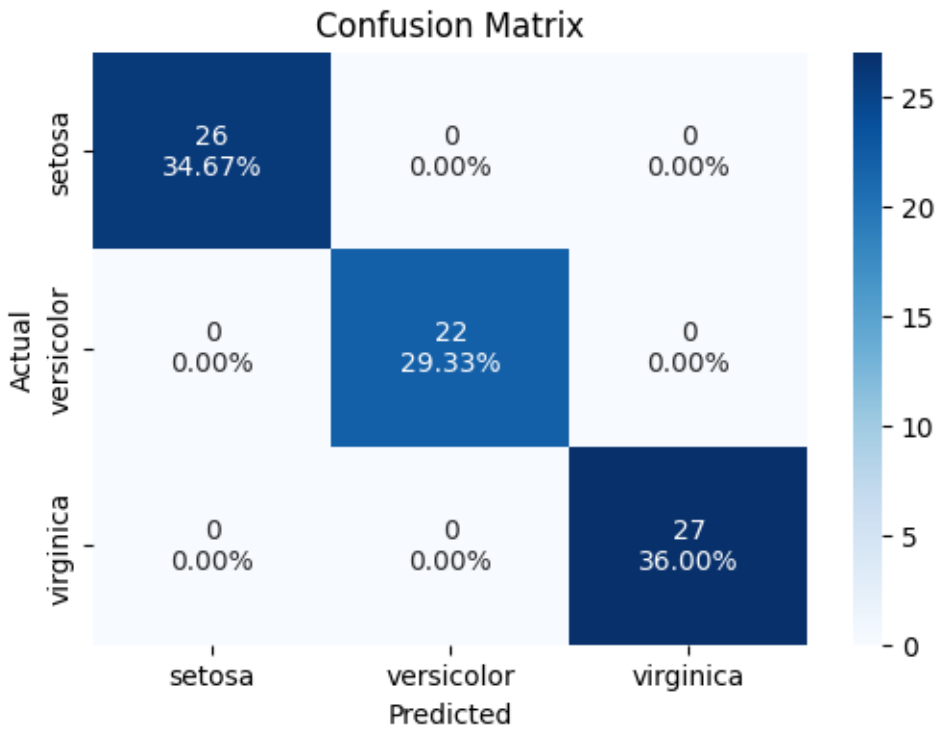



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        26
           1       1.00      1.00      1.00        22
           2       1.00      1.00      1.00        27

    accuracy                           1.00        75
   macro avg       1.00      1.00      1.00        75
weighted avg       1.00      1.00      1.00        75
```

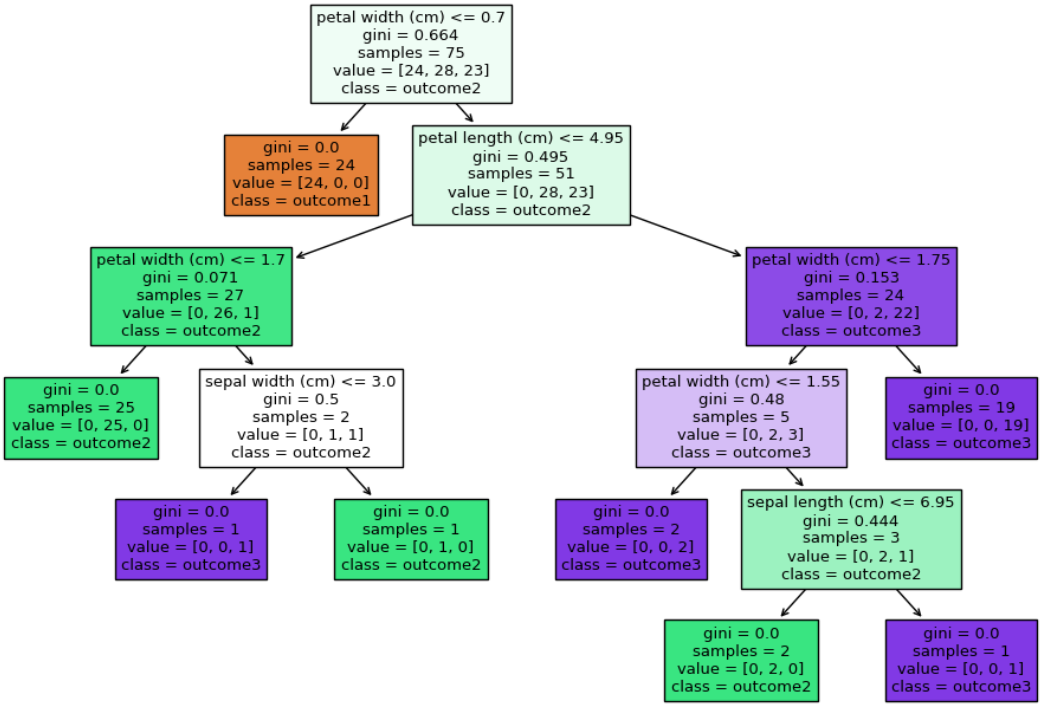


[29]:
```
decision_tree(0.5, 'gini')
```

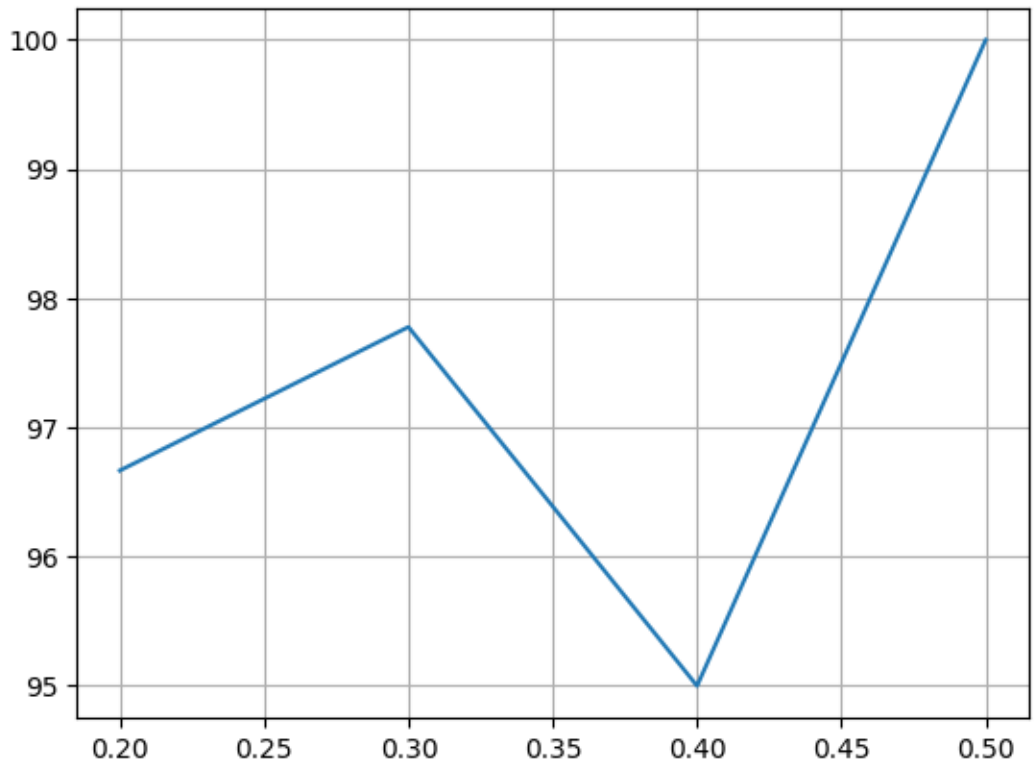
```
Train-test split: 0.5
Value: Entropy: gini
*********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        26
           1       1.00      1.00      1.00        22
           2       1.00      1.00      1.00        27

    accuracy                           1.00        75
   macro avg       1.00      1.00      1.00        75
weighted avg       1.00      1.00      1.00        75
```

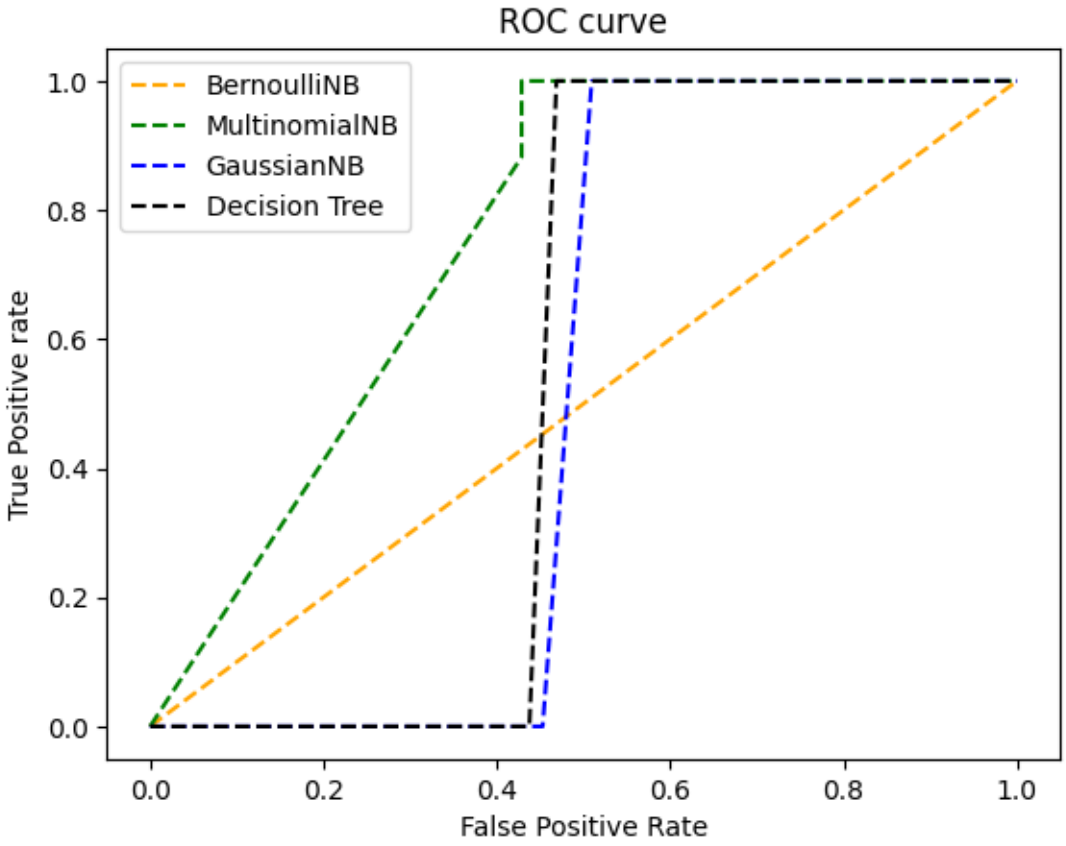


[30]:
```
x_points = [float(key) for key in dict_dtr]
y_points = [i*100 for i in dict_dtr.values()]
plt.plot(x_points, y_points)
plt.grid(True)
plt.show()
```



[31]:
```
from sklearn import metrics
def auc_roc():
    fpr1, tpr1, _1 = metrics.roc_curve(RocAucbnb['max']['y_test'],␣
 ↪RocAucbnb['max']['y_pred'], pos_label=1)
    fpr4, tpr4, _3 = metrics.roc_curve(RocAucmnb['max']['y_test'],␣
 ↪RocAucmnb['max']['y_pred'], pos_label=1)
    fpr2, tpr2, _2 = metrics.roc_curve(RocAucgnb['max']['y_test'],␣
 ↪RocAucgnb['max']['y_pred'], pos_label=1)
    fpr3, tpr3, _3 = metrics.roc_curve(RocAucdtr['max']['y_test'],␣
 ↪RocAucdtr['max']['y_pred'], pos_label=1)
    plt.plot(fpr1, tpr1, linestyle='--',color='orange', label='BernoulliNB')
    plt.plot(fpr4, tpr4, linestyle='--',color='green', label='MultinomialNB')
    plt.plot(fpr2, tpr2, linestyle='--', color='blue', label= 'GaussianNB')
    plt.plot(fpr3, tpr3, linestyle='--', color='black', label= 'Decision Tree')
    plt.title('ROC curve')
    # x label
    plt.xlabel('False Positive Rate')
    # y label
    plt.ylabel('True Positive rate')

    plt.legend(loc='best')
    plt.savefig('ROC',dpi=300)
```

```
    plt.show()
auc_roc()
```



kilus-sayadat-037-diabetes-dataset

August 16, 2023

[ ]:

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

0.0.1 Loading dataset from github

[2]:
```
url = 'https://raw.githubusercontent.com/Aqeel-0/test/build/diabetes.csv'
df = pd.read_csv(url)
df.head()
column_names = list(df.columns.values)
```

[ ]:
```
X = df.iloc[:,:-1]
y = df["Outcome"]
dict_bnb = {}
dict_mnb = {}
dict_gnb = {}
dict_dtr = {}
RocAucbnb = {}
RocAucmnb = {}
RocAucgnb = {}
RocAucdtr = {}
y.info(), X.info()
```

[4]:
```
def plot(y_test, y_pred):
  from sklearn.metrics import confusion_matrix
  import seaborn as sns

  print("Confusion Matrix : ")
  cf_matrix = confusion_matrix(y_test, y_pred)
  group_names = ['True Pos','False Pos','False Neg','True neg']
  group_counts = ["{0:0.0f}".format(value) for value in
                  cf_matrix.flatten()]
  group_percentages = ["{0:.2%}".format(value) for value in
                       cf_matrix.flatten()/np.sum(cf_matrix)]
  labels = [f"{v1}\n{v2}\n{v3}" for v1, v2, v3 in
```

```
            zip(group_names,group_counts,group_percentages)]
  labels = np.asarray(labels).reshape(2,2)
  plt.figure(figsize=(6, 4))
  sns.heatmap(cf_matrix, annot=labels, fmt='', cmap='Blues',␣
↪xticklabels=['Benign', 'Malignant'], yticklabels=['Benign', 'Malignant'])
  plt.xlabel('Predicted')
  plt.ylabel('Actual')
  plt.title('Confusion Matrix')
  plt.show()
  print("***********************************************")
```

[5]:
```
def reports(y_test, y_pred):
  from sklearn.metrics import classification_report
  plot(y_test, y_pred)
  print("***********************************************")
  print("Classification Evaluation : ")
  print(classification_report(y_test, y_pred, zero_division = 0))
```

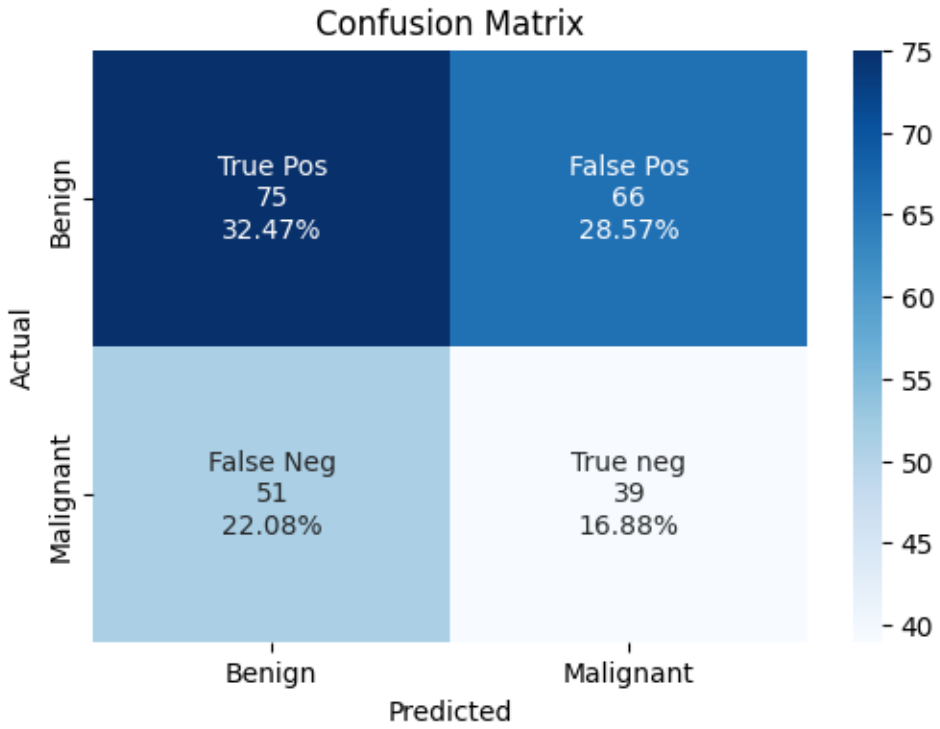
0.0.2 Classification using BernoulliNB Naive Bayes

[6]:
```
def FBouBernoulli(split, alpha_value = 1.0, binarize_value = 0.0,␣
↪fit_prior_value = False):
  from sklearn.naive_bayes import BernoulliNB
  from sklearn.metrics import accuracy_score
  from sklearn.model_selection import train_test_split
  from sklearn.preprocessing import StandardScaler
  #scaler = StandardScaler()
  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = split,␣
↪random_state=44)
  #scaler.fit_transform(X_train)
  #scaler.transform(X_test)
  classifier = BernoulliNB(alpha = alpha_value, binarize = binarize_value,␣
↪fit_prior = fit_prior_value)
  classifier.fit(X_train, y_train)
  y_pred = classifier.predict(X_test)
  print("Train-test split: " + str(split))
  print("value: alpha: "+str(alpha_value) + " binarize: " + str(binarize_value)␣
↪+ " fit_prior: " +str(fit_prior_value))
  print("***********************************************")
  accuracy = accuracy_score(y_test, y_pred)
  if str(split) in dict_bnb:
    dict_bnb[str(split)] = max(accuracy, dict_bnb[str(split)])
    if str(split) == '0.3' and accuracy > dict_bnb[str(split)]:
      RocAucbnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
  else:
    dict_bnb[str(split)] = accuracy
```

```
    if str(split) == '0.3':
      RocAucbnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
  reports(y_test, y_pred)
```

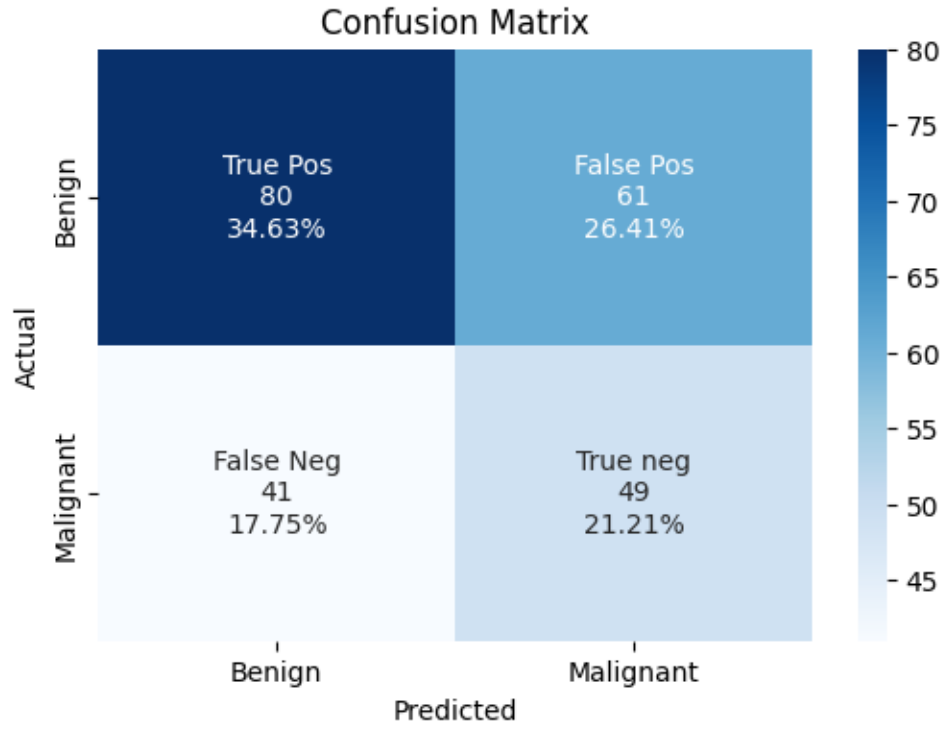
[7]:
```
## Train-Test split 0.3
FBouBernoulli(0.3)
FBouBernoulli(0.3, 1.0, 1.5)
FBouBernoulli(0.3, 1.0, 1.5, True)
```

```
Train-test split: 0.3
value: alpha: 1.0 binarize: 0.0 fit_prior: False
**********************************************
Confusion Matrix :
```

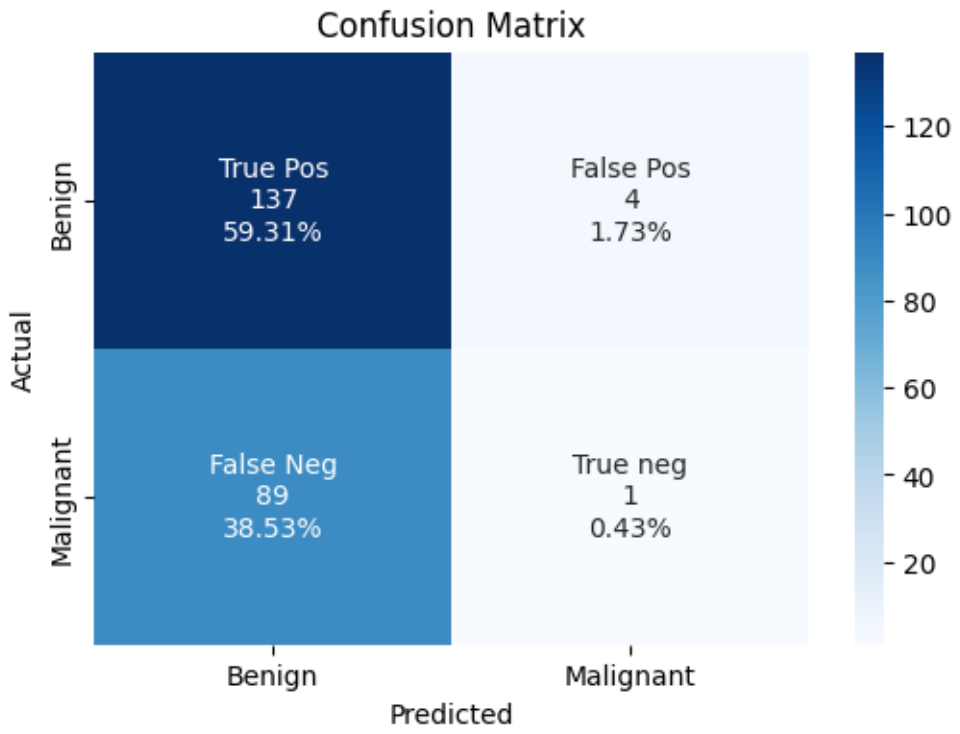



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.60      0.53      0.56       141
           1       0.37      0.43      0.40        90

    accuracy                           0.49       231
   macro avg       0.48      0.48      0.48       231
```

```
weighted avg       0.51      0.49      0.50       231

Train-test split: 0.3
value: alpha: 1.0 binarize: 1.5 fit_prior: False
**********************************************
Confusion Matrix :
```



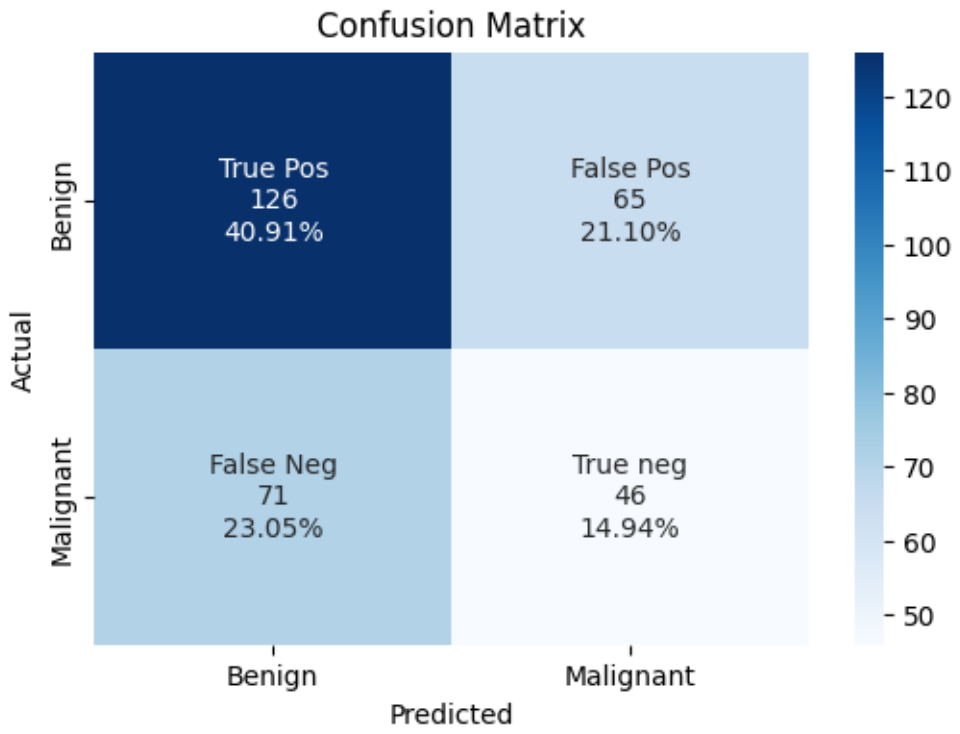

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.66      0.57      0.61       141
           1       0.45      0.54      0.49        90

    accuracy                           0.56       231
   macro avg       0.55      0.56      0.55       231
weighted avg       0.58      0.56      0.56       231

Train-test split: 0.3
value: alpha: 1.0 binarize: 1.5 fit_prior: True
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.61      0.97      0.75       141
           1       0.20      0.01      0.02        90

    accuracy                           0.60       231
   macro avg       0.40      0.49      0.38       231
weighted avg       0.45      0.60      0.46       231
```

[8]:
```
## Train-Test split 0.4
FBouBernoulli(0.4)
```

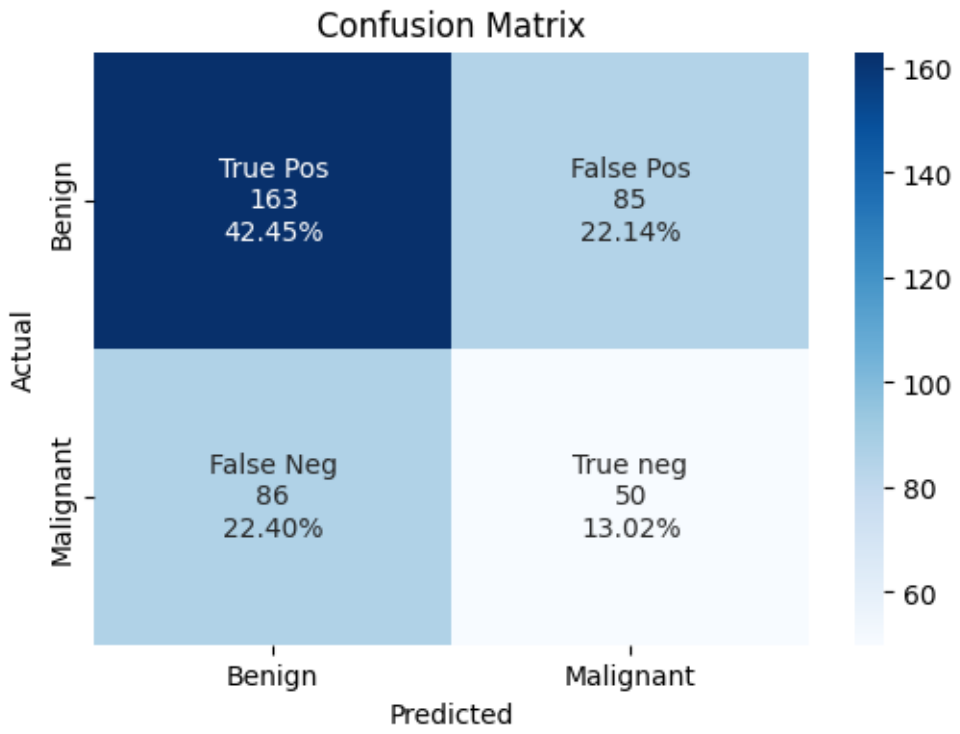
```
Train-test split: 0.4
value: alpha: 1.0 binarize: 0.0 fit_prior: False
**********************************************
Confusion Matrix :
```



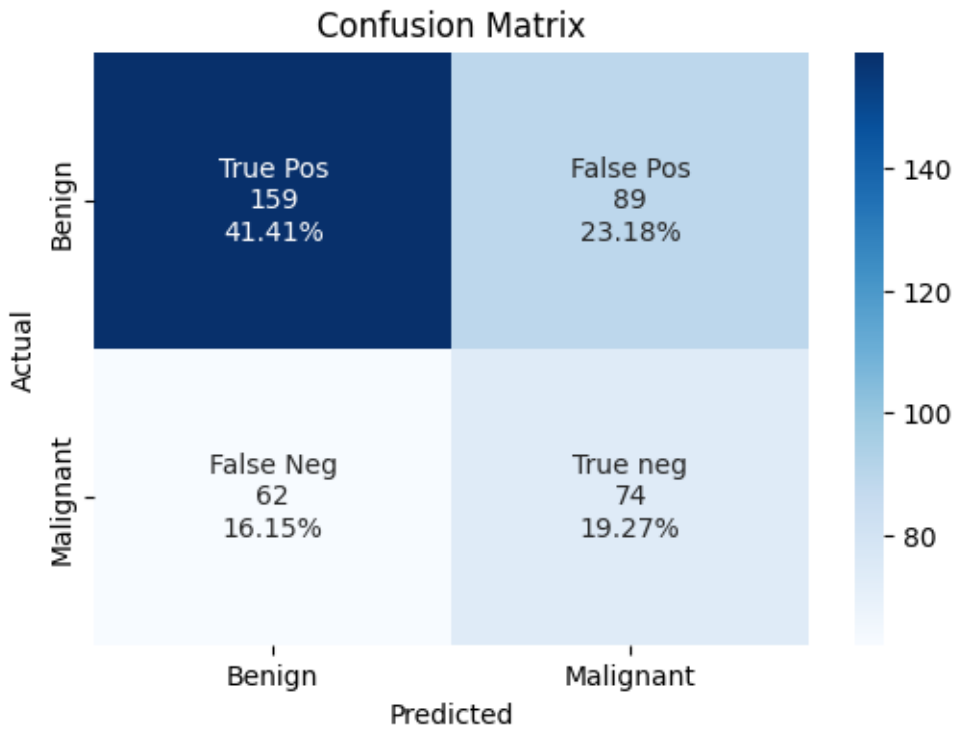

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.64      0.66      0.65       191
           1       0.41      0.39      0.40       117

    accuracy                           0.56       308
   macro avg       0.53      0.53      0.53       308
weighted avg       0.55      0.56      0.56       308
```

[9]:
```
## Train-Test split 0.5
FBouBernoulli(0.5)
FBouBernoulli(0.5, 1.0, 7.9)
FBouBernoulli(0.5, 1.0, 7.9, True)
```

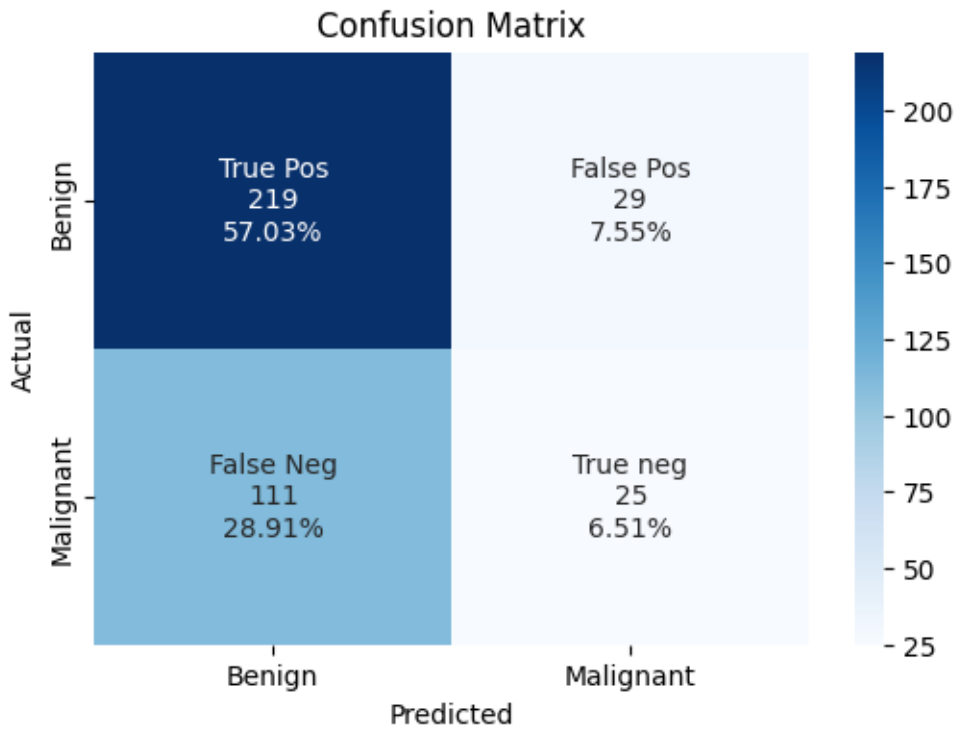
```
Train-test split: 0.5
value: alpha: 1.0 binarize: 0.0 fit_prior: False
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.65      0.66      0.66       248
           1       0.37      0.37      0.37       136

    accuracy                           0.55       384
   macro avg       0.51      0.51      0.51       384
weighted avg       0.55      0.55      0.55       384

Train-test split: 0.5
value: alpha: 1.0 binarize: 7.9 fit_prior: False
**********************************************
Confusion Matrix :
```

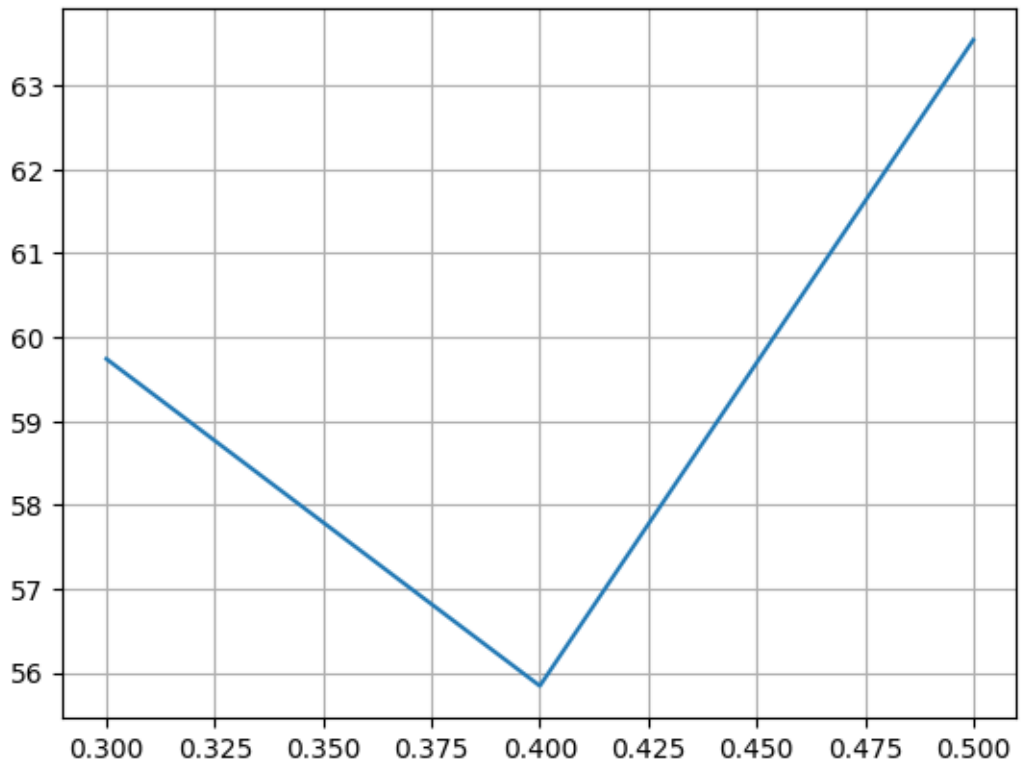



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.72      0.64      0.68       248
           1       0.45      0.54      0.49       136

    accuracy                           0.61       384
   macro avg       0.59      0.59      0.59       384
weighted avg       0.63      0.61      0.61       384

Train-test split: 0.5
value: alpha: 1.0 binarize: 7.9 fit_prior: True
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.66      0.88      0.76       248
           1       0.46      0.18      0.26       136

    accuracy                           0.64       384
   macro avg       0.56      0.53      0.51       384
weighted avg       0.59      0.64      0.58       384
```

[10]:
```
x_points = [float(key) for key in dict_bnb]
y_points = [i*100 for i in dict_bnb.values()]
plt.plot(x_points, y_points)
plt.grid(True)
plt.show()
```

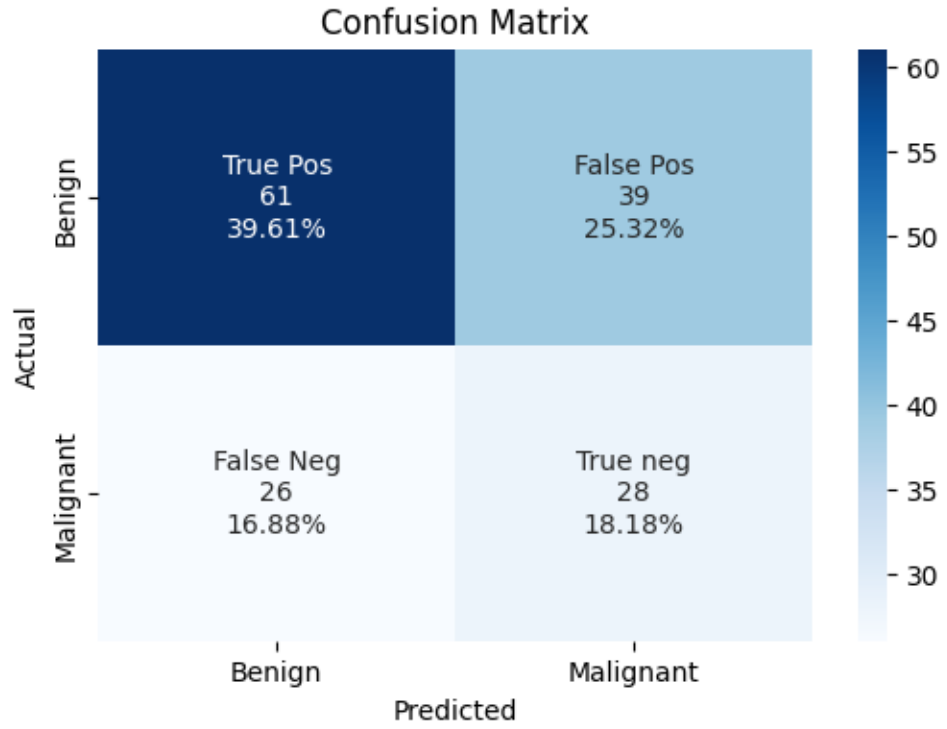


0.1 Classification using Multinomial Naive Bayes

[11]:
```
def FMultinomial(split, alpha_value = 1.0):
  from sklearn.naive_bayes import MultinomialNB
  from sklearn.metrics import accuracy_score
  from sklearn.model_selection import train_test_split
  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = split)
  classifier = MultinomialNB(alpha = alpha_value)
  classifier.fit(X_train, y_train)
  y_pred = classifier.predict(X_test)
  print("Train-test split: " + str(split))
  print("value: alpha: "+str(alpha_value))
  print("*********************************************")
  accuracy = accuracy_score(y_test, y_pred)
  if str(split) in dict_mnb:
    dict_mnb[str(split)] = max(accuracy, dict_mnb[str(split)])
    if str(split) == '0.3' and accuracy > dict_mnb[str(split)]:
      RocAucmnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
  else:
    dict_mnb[str(split)] = accuracy
    if str(split) == '0.3':
      RocAucmnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
```

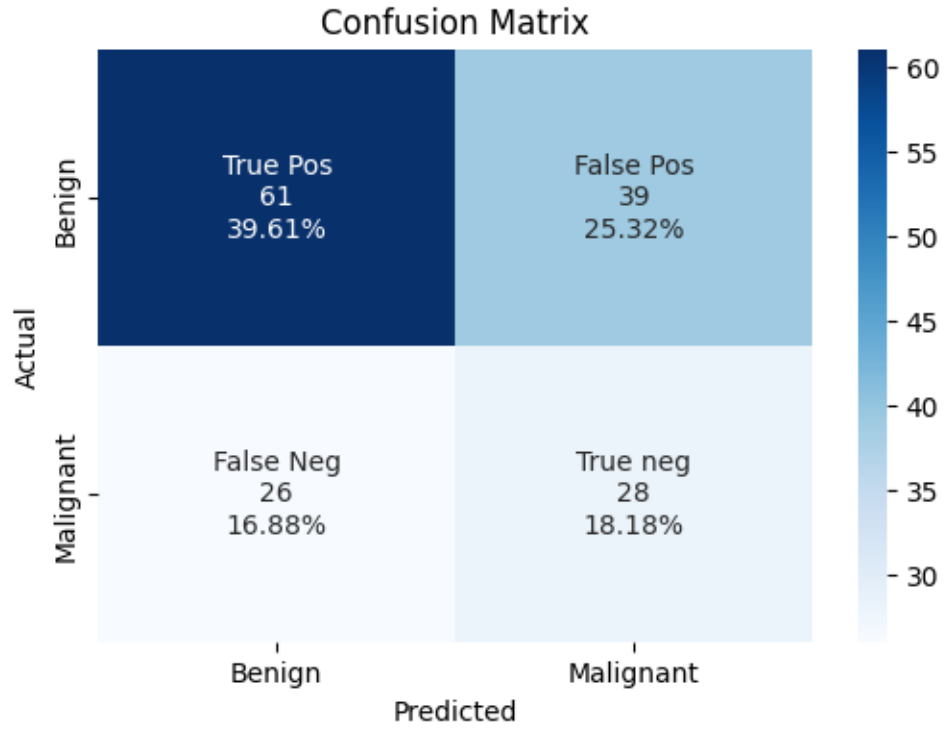
```
  reports(y_test, y_pred)
  reports(y_test, y_pred)

## Train-Test split 0.2
FMultinomial(0.2)
FMultinomial(0.2, 1.8)
```

```
Train-test split: 0.2
value: alpha: 1.0
**********************************************
Confusion Matrix :
```

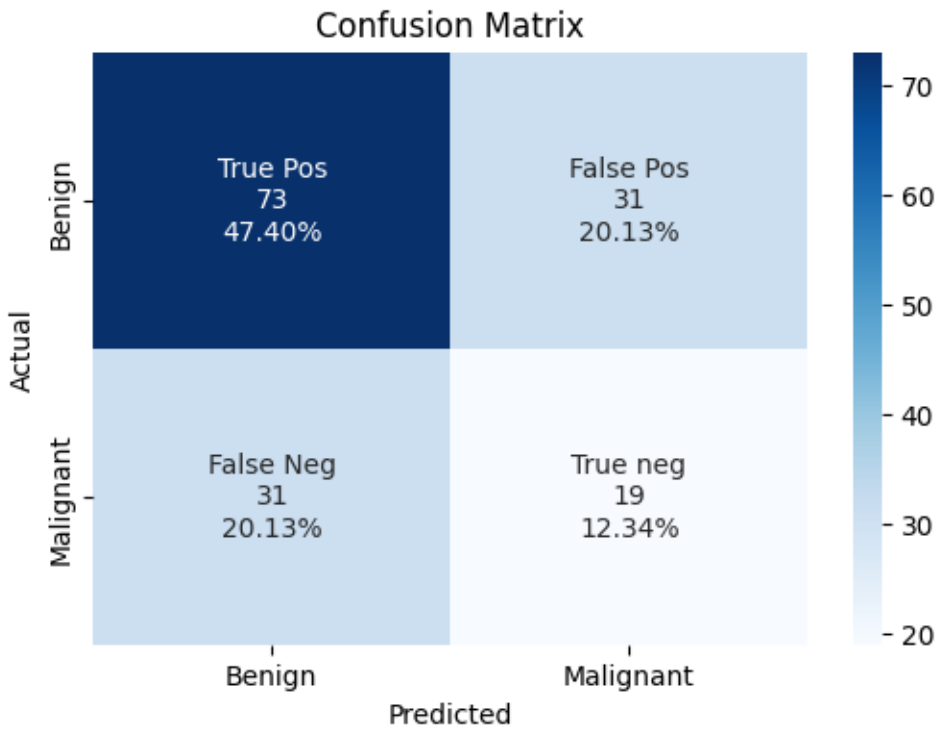



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.70      0.61      0.65       100
           1       0.42      0.52      0.46        54

    accuracy                           0.58       154
   macro avg       0.56      0.56      0.56       154
weighted avg       0.60      0.58      0.59       154
```

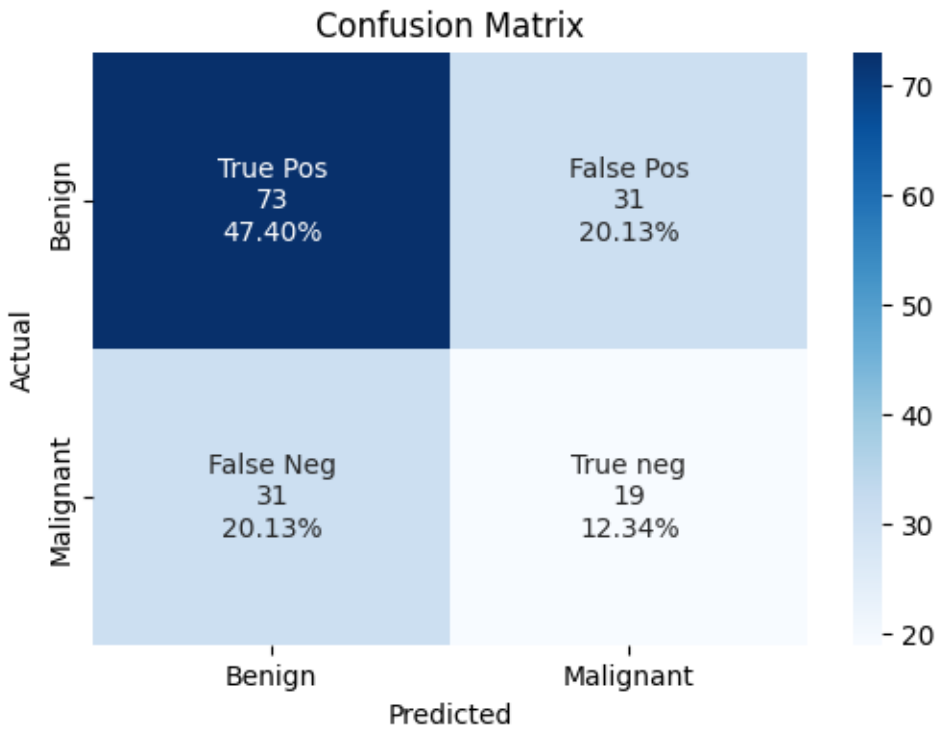
Confusion Matrix :




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.70      0.61      0.65       100
           1       0.42      0.52      0.46        54

    accuracy                           0.58       154
   macro avg       0.56      0.56      0.56       154
weighted avg       0.60      0.58      0.59       154

Train-test split: 0.2
value: alpha: 1.8
**********************************************
Confusion Matrix :
```



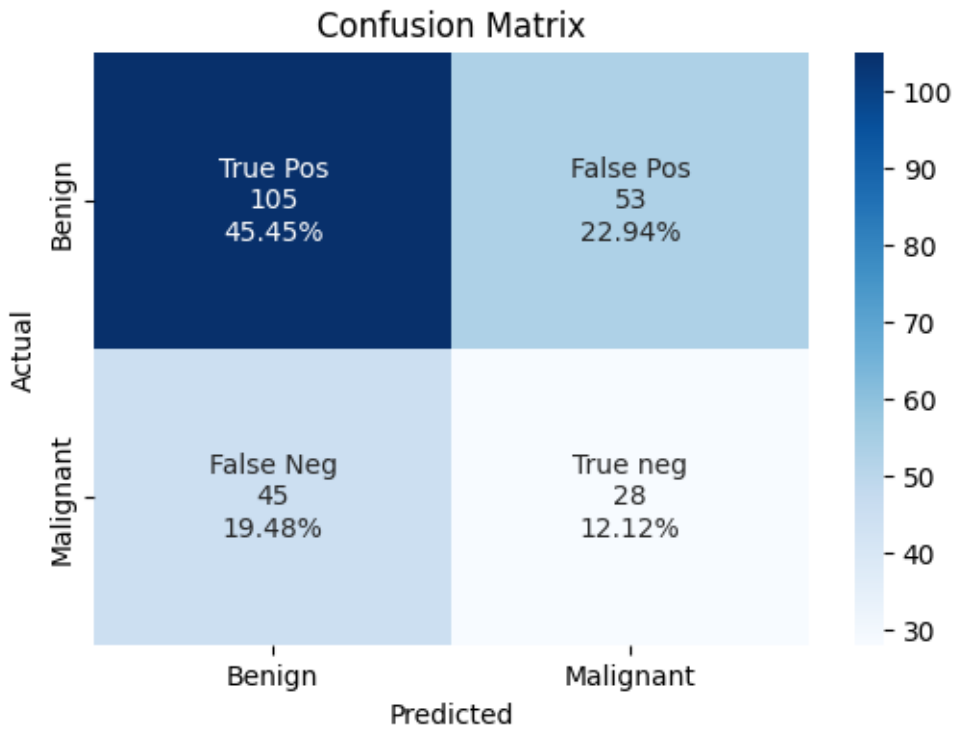

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.70      0.70      0.70       104
           1       0.38      0.38      0.38        50

    accuracy                           0.60       154
   macro avg       0.54      0.54      0.54       154
weighted avg       0.60      0.60      0.60       154

Confusion Matrix :
```



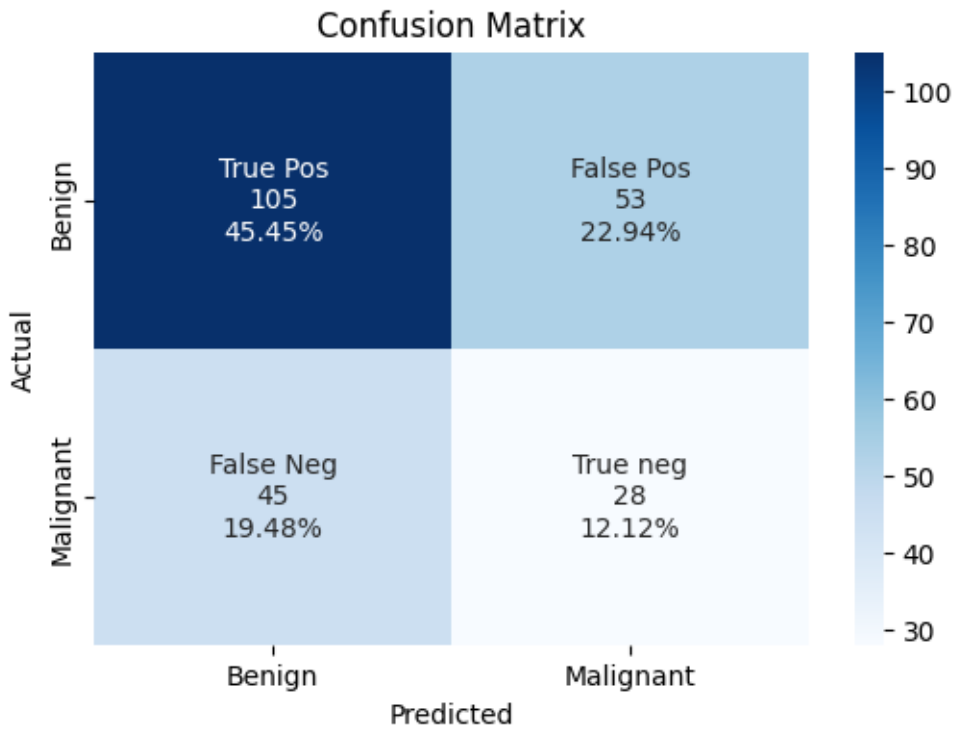

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.70      0.70      0.70       104
           1       0.38      0.38      0.38        50

    accuracy                           0.60       154
   macro avg       0.54      0.54      0.54       154
weighted avg       0.60      0.60      0.60       154
```

[12]:
```
## Train-Test split 0.3
FMultinomial(0.3)
FMultinomial(0.3, 2.9)
```

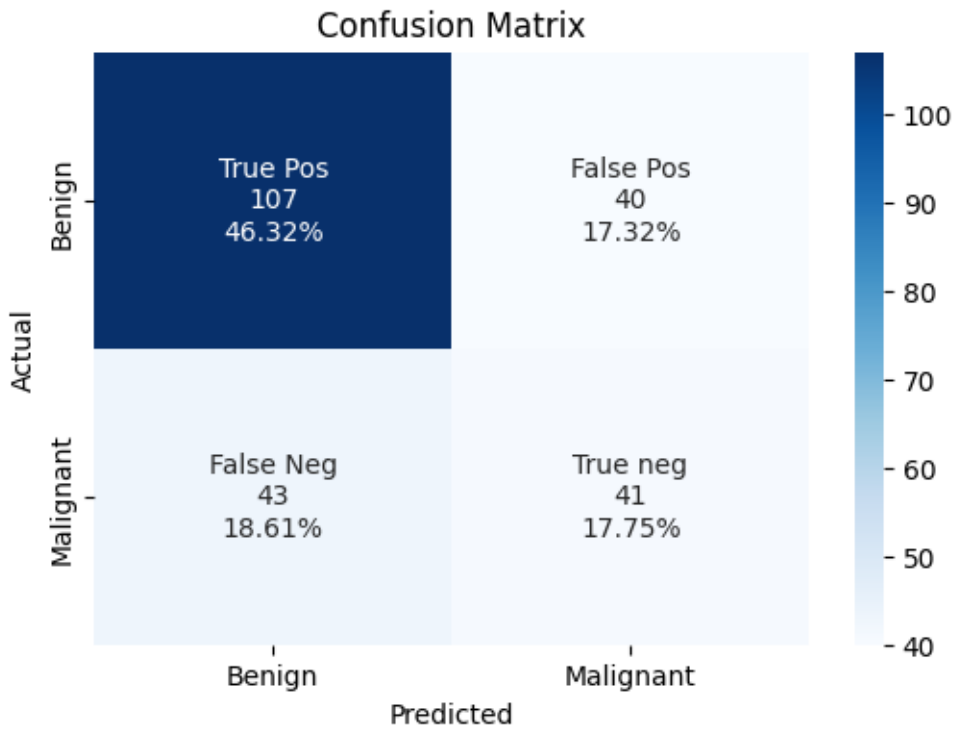
```
Train-test split: 0.3
value: alpha: 1.0
**********************************************
Confusion Matrix :
```



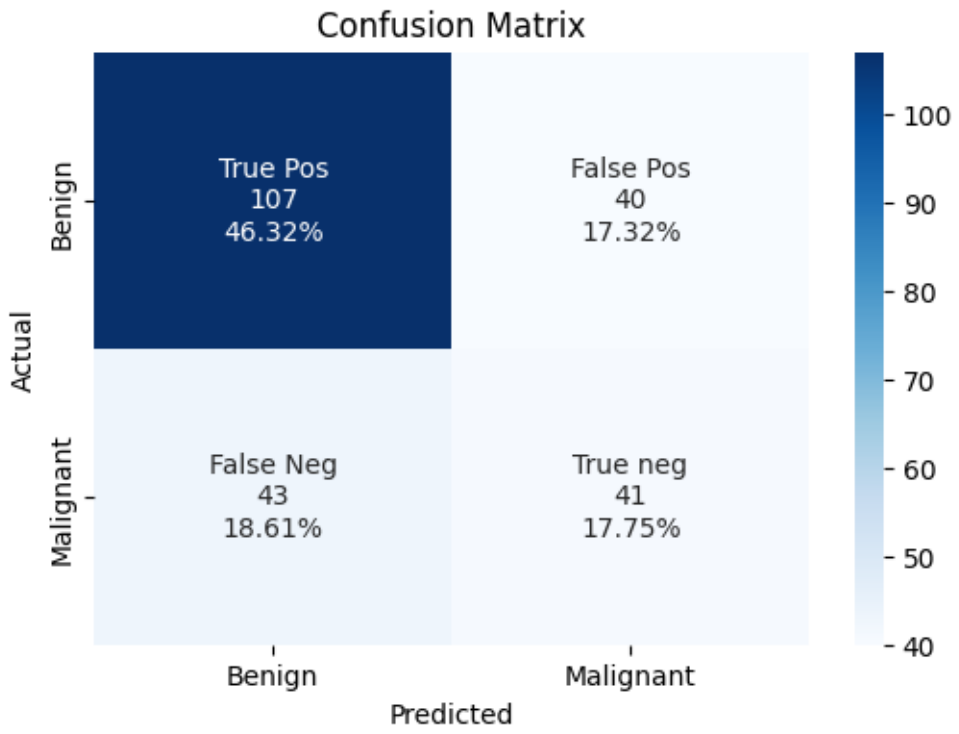

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.70      0.66      0.68       158
           1       0.35      0.38      0.36        73

    accuracy                           0.58       231
   macro avg       0.52      0.52      0.52       231
weighted avg       0.59      0.58      0.58       231

Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.70      0.66      0.68       158
           1       0.35      0.38      0.36        73

    accuracy                           0.58       231
   macro avg       0.52      0.52      0.52       231
weighted avg       0.59      0.58      0.58       231

Train-test split: 0.3
value: alpha: 2.9
**********************************************
Confusion Matrix :
```



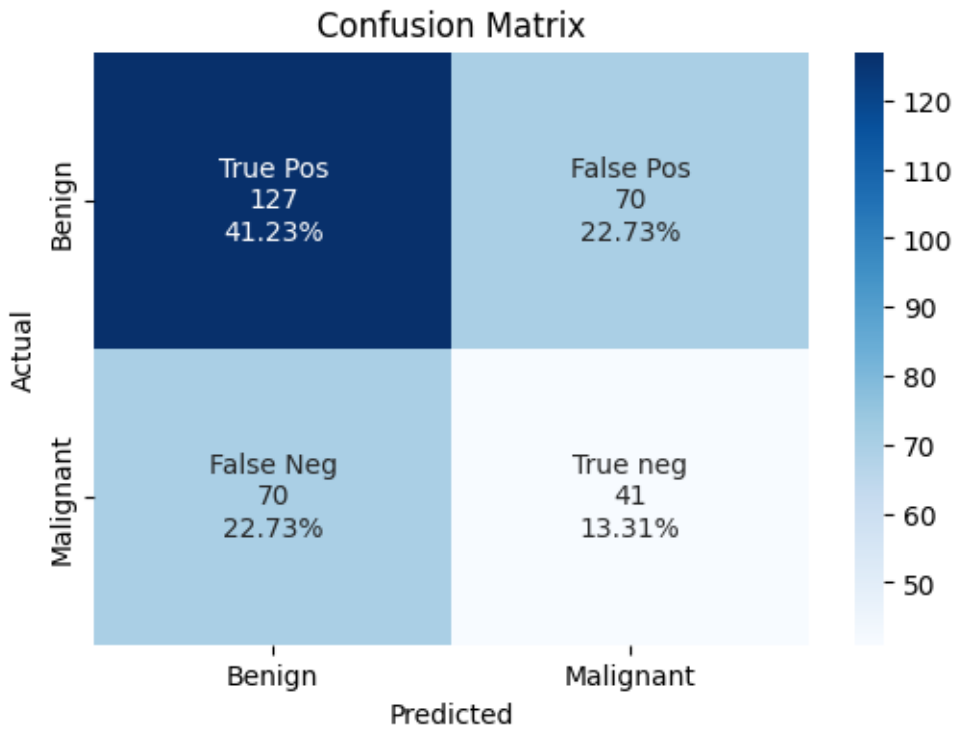

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.71      0.73      0.72       147
           1       0.51      0.49      0.50        84

    accuracy                           0.64       231
   macro avg       0.61      0.61      0.61       231
weighted avg       0.64      0.64      0.64       231

Confusion Matrix :
```



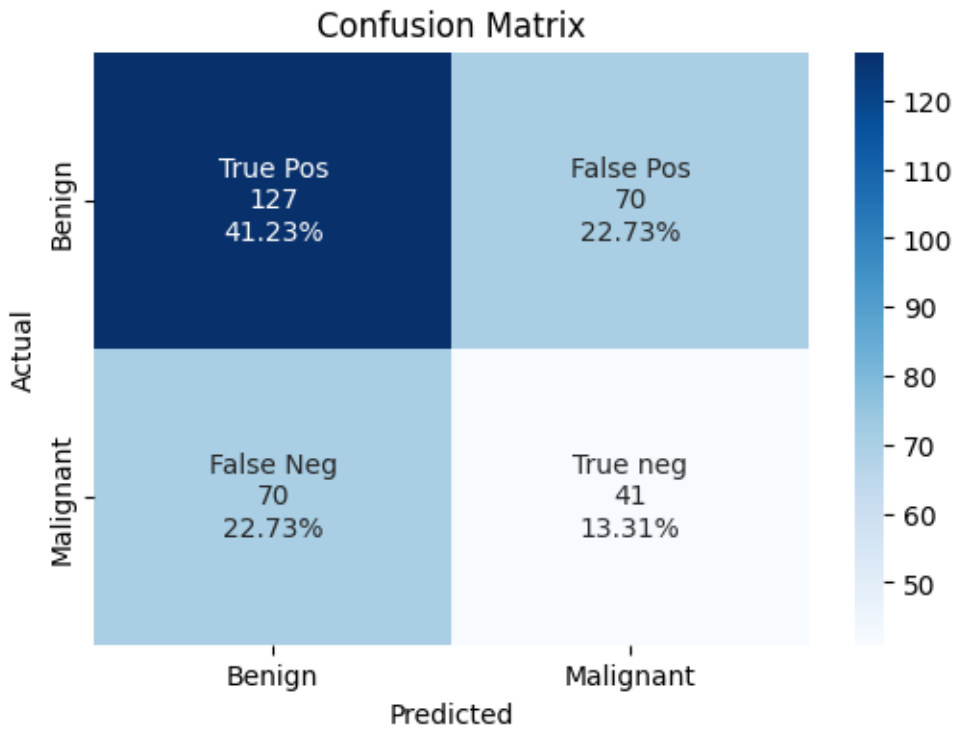

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.71      0.73      0.72       147
           1       0.51      0.49      0.50        84

    accuracy                           0.64       231
   macro avg       0.61      0.61      0.61       231
weighted avg       0.64      0.64      0.64       231
```

[13]:
```
## Train-Test split 0.4
FMultinomial(0.4)
FMultinomial(0.4, 1.1)
```

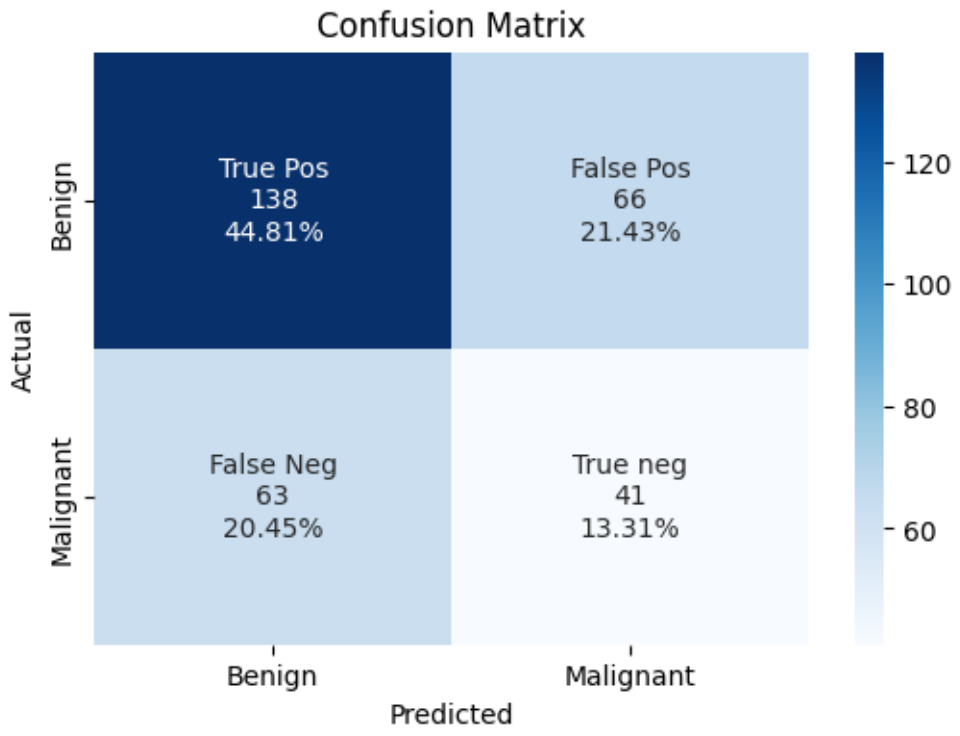
```
Train-test split: 0.4
value: alpha: 1.0
**********************************************
Confusion Matrix :
```



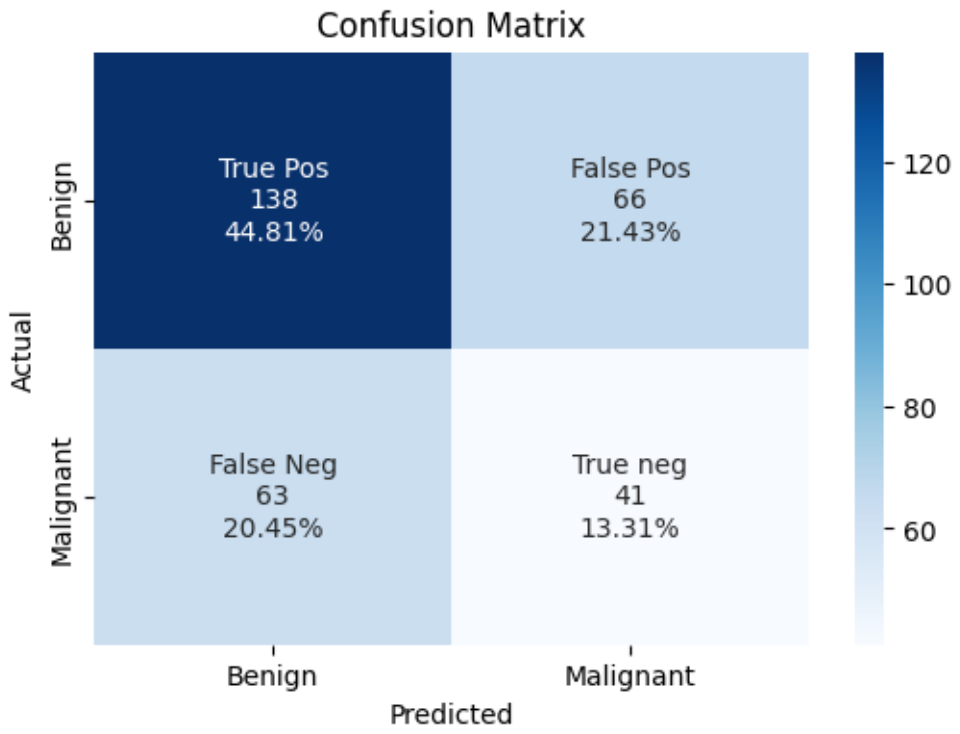

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.64      0.64      0.64       197
           1       0.37      0.37      0.37       111

    accuracy                           0.55       308
   macro avg       0.51      0.51      0.51       308
weighted avg       0.55      0.55      0.55       308

Confusion Matrix :
```




```
*********************************************
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.64      0.64      0.64       197
           1       0.37      0.37      0.37       111

    accuracy                           0.55       308
   macro avg       0.51      0.51      0.51       308
weighted avg       0.55      0.55      0.55       308

Train-test split: 0.4
value: alpha: 1.1
*********************************************
Confusion Matrix :
```



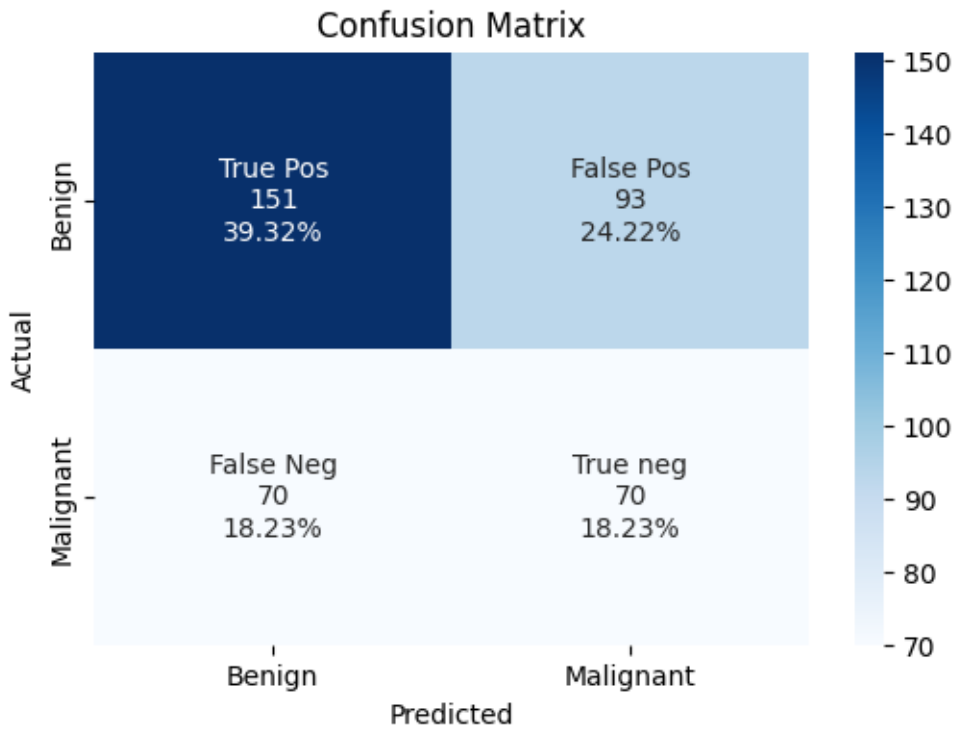

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.69      0.68      0.68       204
           1       0.38      0.39      0.39       104

    accuracy                           0.58       308
   macro avg       0.53      0.54      0.54       308
weighted avg       0.58      0.58      0.58       308

Confusion Matrix :
```



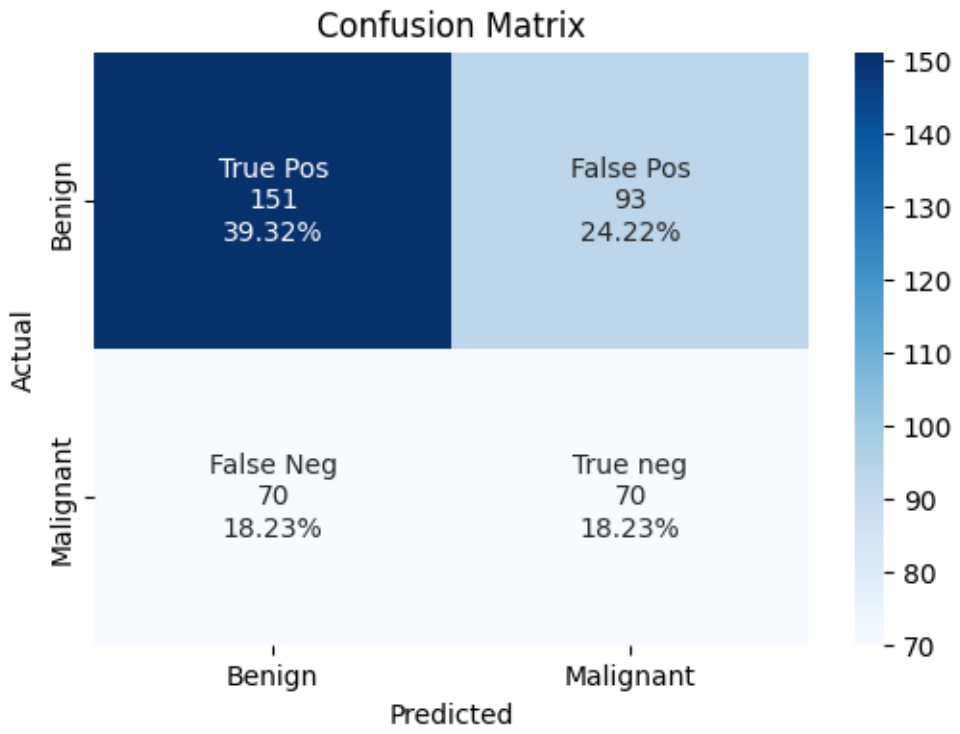

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.69      0.68      0.68       204
           1       0.38      0.39      0.39       104

    accuracy                           0.58       308
   macro avg       0.53      0.54      0.54       308
weighted avg       0.58      0.58      0.58       308
```

[14]:
```
## Train-Test split 0.5
FMultinomial(0.5)
FMultinomial(0.5, 4.8)
```

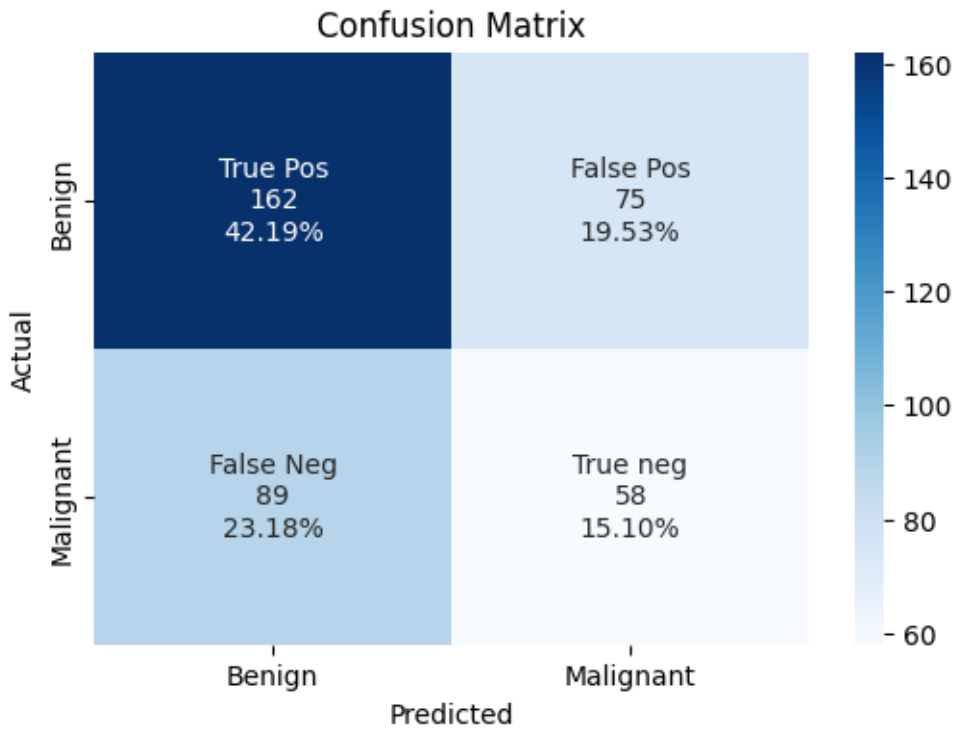
```
Train-test split: 0.5
value: alpha: 1.0
**********************************************
Confusion Matrix :
```



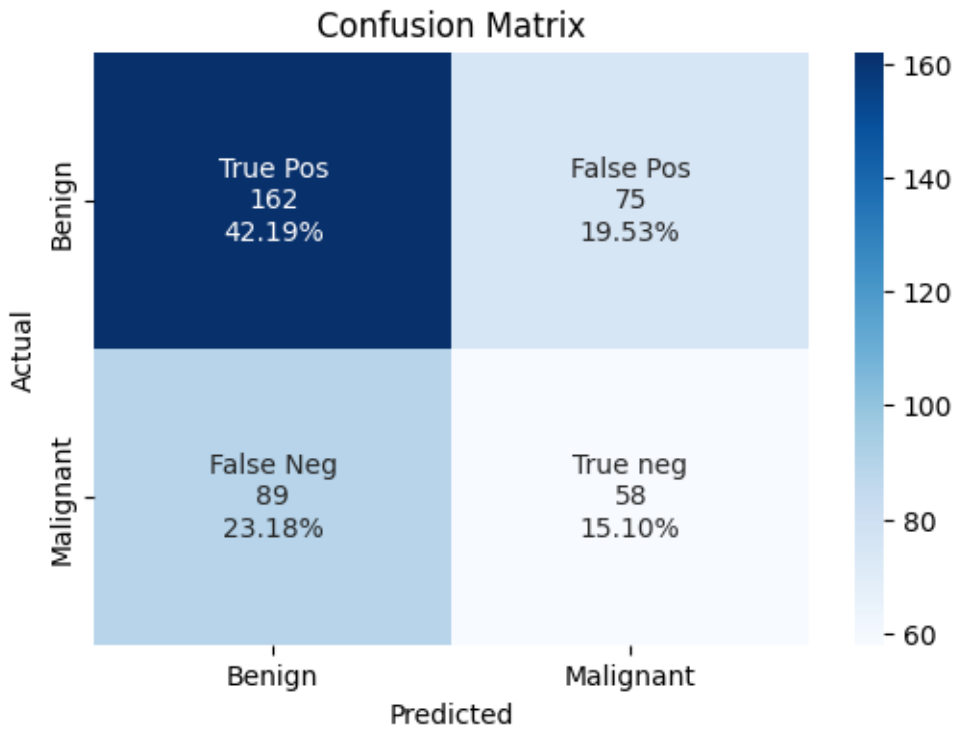

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.68      0.62      0.65       244
           1       0.43      0.50      0.46       140

    accuracy                           0.58       384
   macro avg       0.56      0.56      0.56       384
weighted avg       0.59      0.58      0.58       384

Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.68      0.62      0.65       244
           1       0.43      0.50      0.46       140

    accuracy                           0.58       384
   macro avg       0.56      0.56      0.56       384
weighted avg       0.59      0.58      0.58       384

Train-test split: 0.5
value: alpha: 4.8
**********************************************
Confusion Matrix :
```

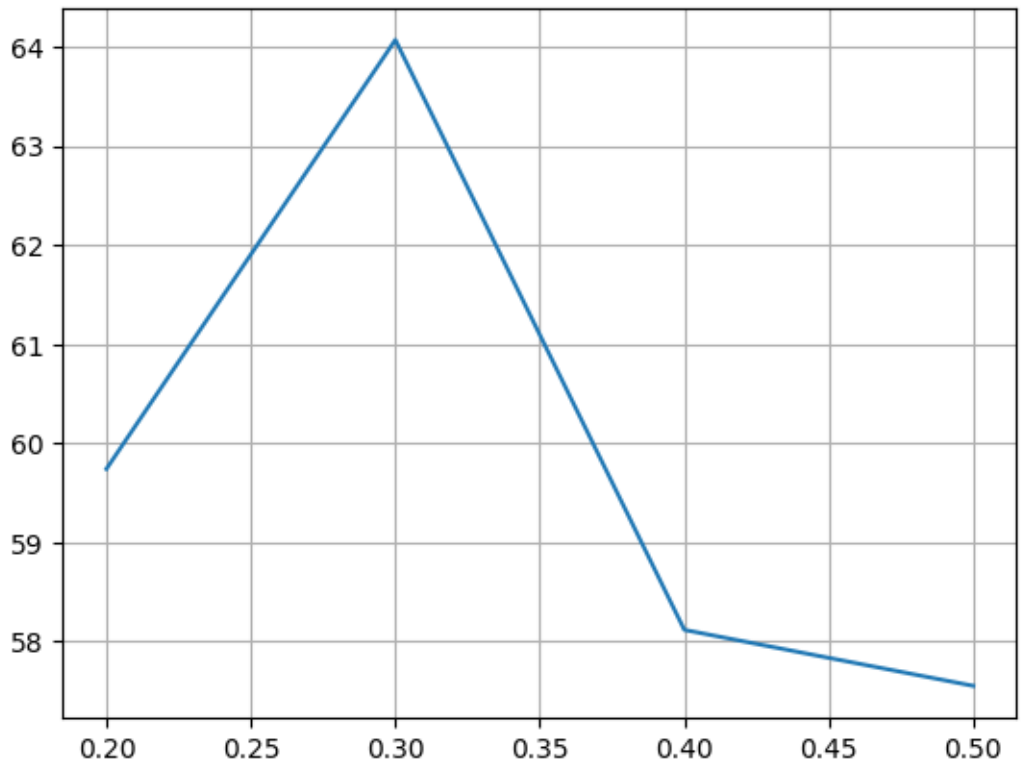



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.65      0.68      0.66       237
           1       0.44      0.39      0.41       147

    accuracy                           0.57       384
   macro avg       0.54      0.54      0.54       384
weighted avg       0.57      0.57      0.57       384

Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.65      0.68      0.66       237
           1       0.44      0.39      0.41       147

    accuracy                           0.57       384
   macro avg       0.54      0.54      0.54       384
weighted avg       0.57      0.57      0.57       384
```

[15]:
```
x_points = [float(key) for key in dict_mnb]
y_points = [i*100 for i in dict_mnb.values()]
plt.plot(x_points, y_points)
plt.grid(True)
plt.show()
```

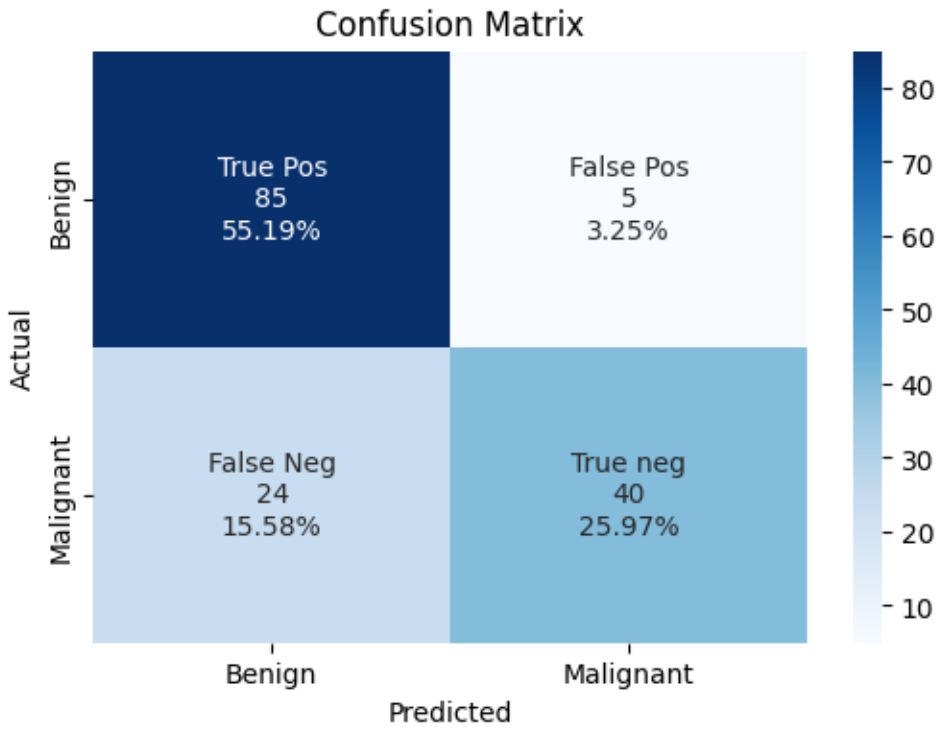


0.1.1 Classification using Guassian Naive Bayes

[16]:
```
def FGaussian(split):
  from sklearn.naive_bayes import GaussianNB
  from sklearn.model_selection import train_test_split
  from sklearn.metrics import accuracy_score
  from sklearn.preprocessing import StandardScaler
  scaler = StandardScaler()
  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = split,␣
 ↪random_state=44)
  scaler.fit_transform(X_train)
  scaler.transform(X_test)
  classifier = GaussianNB()
  classifier.fit(X_train, y_train)
  y_pred = classifier.predict(X_test)
  print("Train-test split: " + str(split))
  print("**********************************************")
  reports(y_test, y_pred)
  accuracy = accuracy_score(y_test, y_pred)
  if(str(split) in dict_gnb):
    dict_gnb[str(split)] = max(accuracy, dict_gnb[str(split)])
    if(str(split) == '0.3' and accuracy > dict_bnb[str(split)]):
```

```
      RocAucgnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
  else:
    dict_gnb[str(split)] = accuracy
    RocAucgnb['max'] = {'y_test': y_test, 'y_pred': y_pred}

## Train-Test split 0.2
FGaussian(0.2)
```

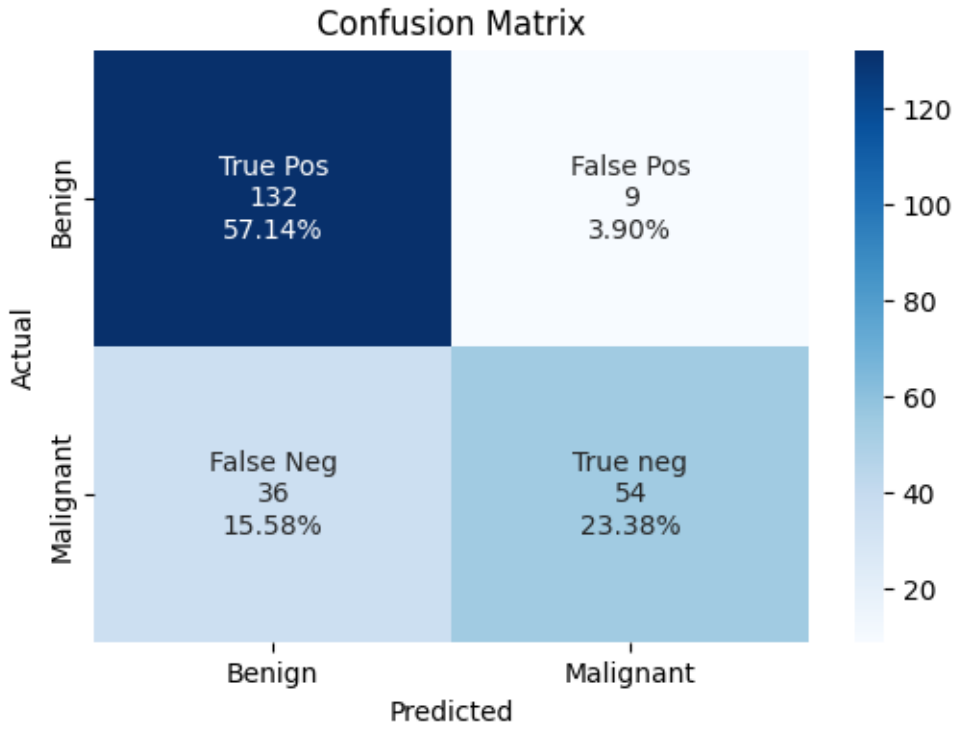
```
Train-test split: 0.2
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.78      0.94      0.85        90
           1       0.89      0.62      0.73        64

    accuracy                           0.81       154
   macro avg       0.83      0.78      0.79       154
weighted avg       0.83      0.81      0.80       154
```

[17]:
```
## Train-Test split 0.3
FGaussian(0.3)
```

```
Train-test split: 0.3
**********************************************
Confusion Matrix :
```

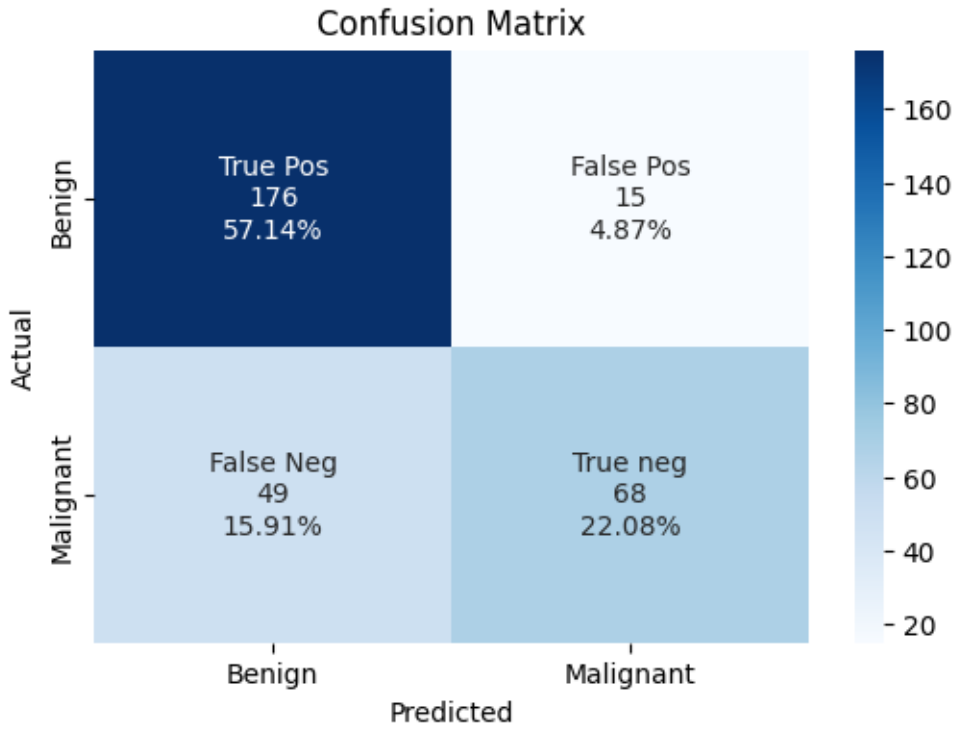



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.79      0.94      0.85       141
           1       0.86      0.60      0.71        90

    accuracy                           0.81       231
   macro avg       0.82      0.77      0.78       231
weighted avg       0.81      0.81      0.80       231
```

[18]:
```
## Train-Test split 0.4
FGaussian(0.4)
```

```
Train-test split: 0.4
```

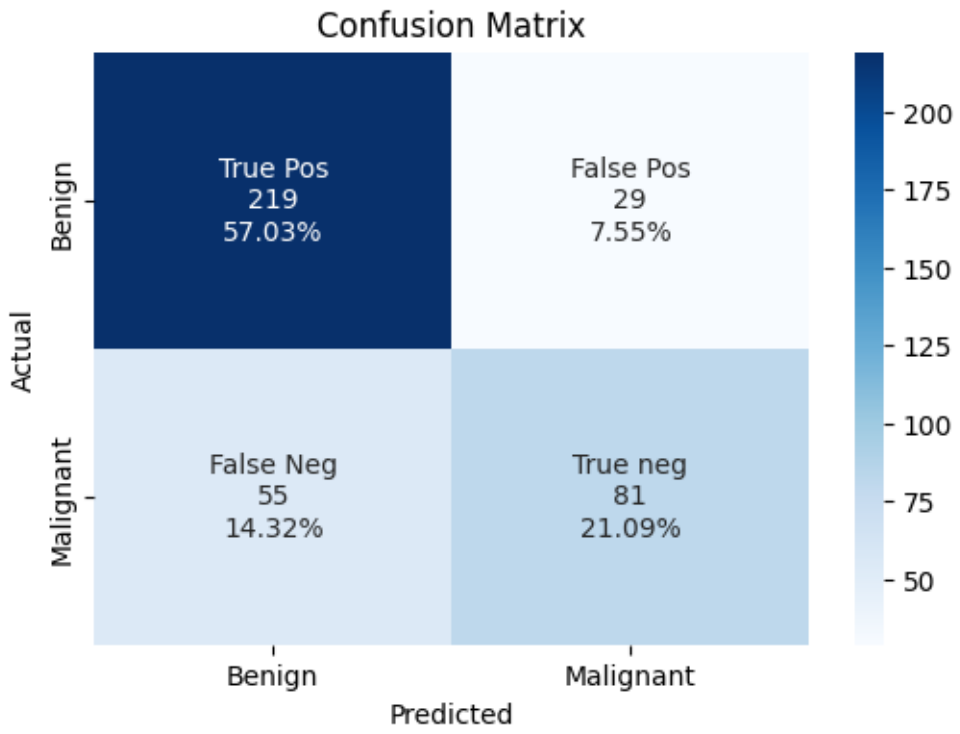
```
*********************************************
Confusion Matrix :
```




```
*********************************************
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.78      0.92      0.85       191
           1       0.82      0.58      0.68       117

    accuracy                           0.79       308
   macro avg       0.80      0.75      0.76       308
weighted avg       0.80      0.79      0.78       308
```

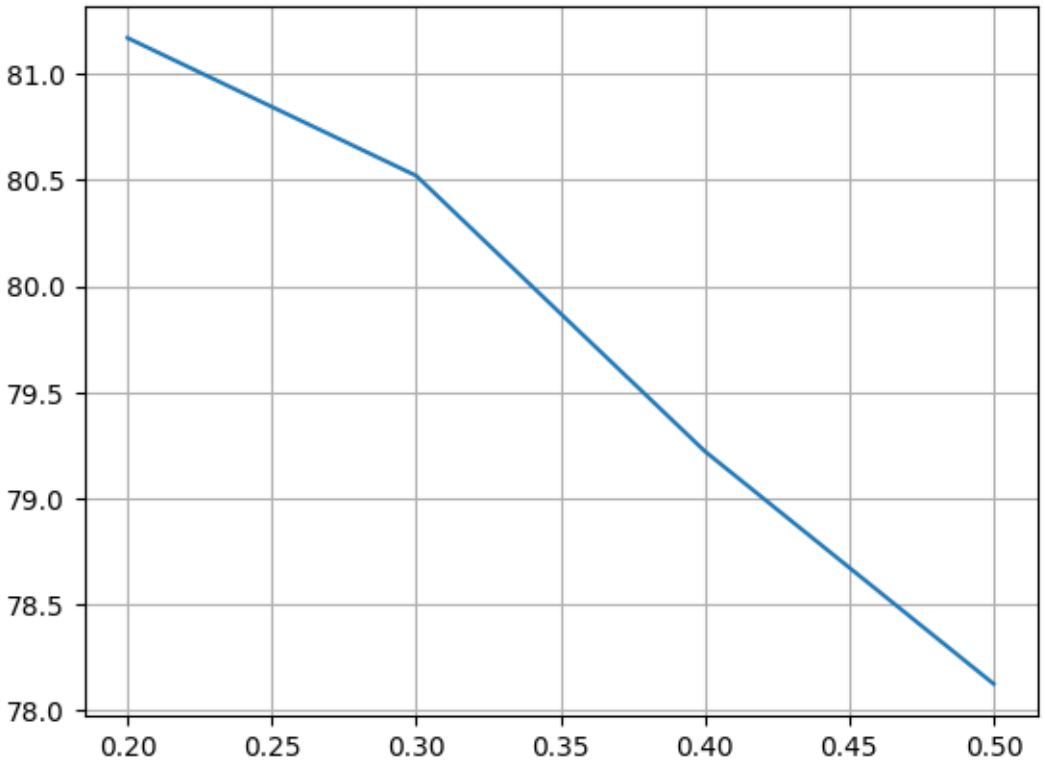
[19]:
```
## Train-Test split 0.5
FGaussian(0.5)
```

```
Train-test split: 0.5
*********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.80      0.88      0.84       248
           1       0.74      0.60      0.66       136

    accuracy                           0.78       384
   macro avg       0.77      0.74      0.75       384
weighted avg       0.78      0.78      0.78       384
```

[20]:
```
x_points = [float(key) for key in dict_gnb]
y_points = [i*100 for i in dict_gnb.values()]
plt.plot(x_points, y_points)
plt.grid(True)
plt.show()
```



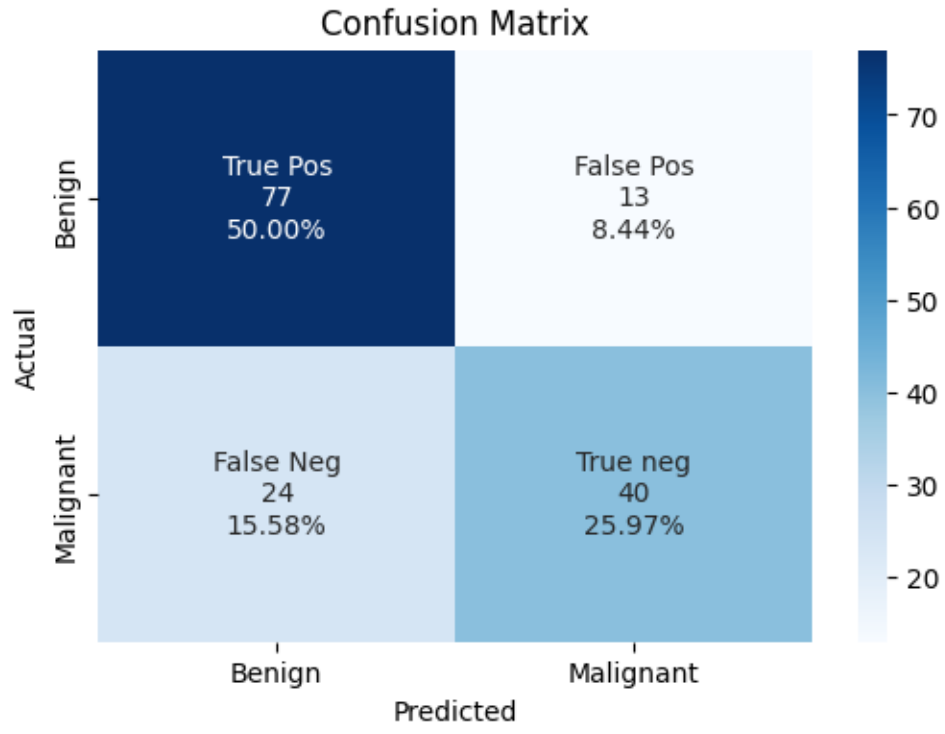
0.1.2 Classification using Decision Tree

[21]:
```
def decision_tree(split, criterion_value):
  from sklearn.model_selection import train_test_split
  from sklearn.tree import DecisionTreeClassifier
  from sklearn import tree
  from sklearn.metrics import accuracy_score
  from sklearn.preprocessing import StandardScaler
  scaler = StandardScaler()
  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = split,␣
 ↪random_state=44)
  scaler.fit_transform(X_train)
  scaler.transform(X_test)

  classifier = DecisionTreeClassifier(criterion = criterion_value)
  classifier.fit(X_train, y_train)
  y_pred = classifier.predict(X_test)
  print("Train-test split: " + str(split))
  print("Value: Entropy: " + criterion_value)
  print("********************************************")
  reports(y_test, y_pred)
  accuracy = accuracy_score(y_test, y_pred)
```

```
if(str(split) in dict_dtr):
  dict_dtr[str(split)] = max(accuracy, dict_dtr[str(split)])
  if(str(split) == '0.3' and accuracy > dict_bnb[str(split)]):
    RocAucdtr['max'] = {'y_test': y_test, 'y_pred': y_pred}
else:
  dict_dtr[str(split)] = accuracy
  if(str(split) == '0.3'):
    RocAucdtr['max'] = {'y_test': y_test, 'y_pred': y_pred}

fig = plt.figure(figsize=(12,8))
_ = tree.plot_tree(classifier,
                   feature_names=column_names,
                   class_names=['outcome1', 'outcome2'],
                   filled=True)
```

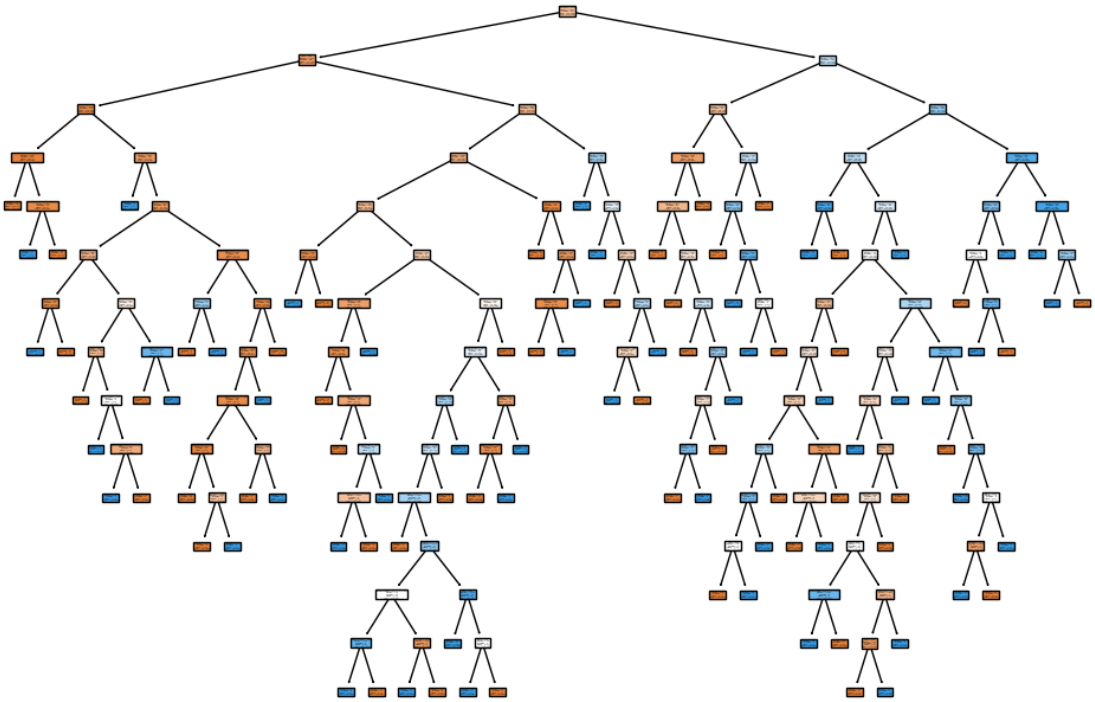
[22]:
```
decision_tree(0.2, 'entropy')
```

```
Train-test split: 0.2
Value: Entropy: entropy
*********************************************
Confusion Matrix :
```



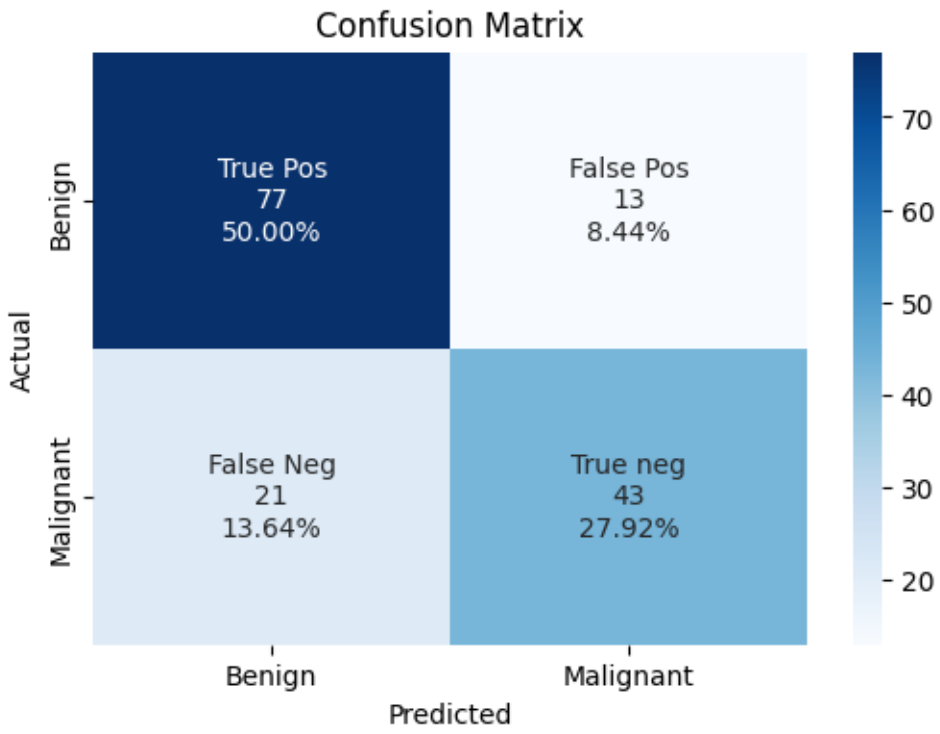

```
*********************************************
```

```
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.76      0.86      0.81        90
           1       0.75      0.62      0.68        64

    accuracy                           0.76       154
   macro avg       0.76      0.74      0.75       154
weighted avg       0.76      0.76      0.76       154
```

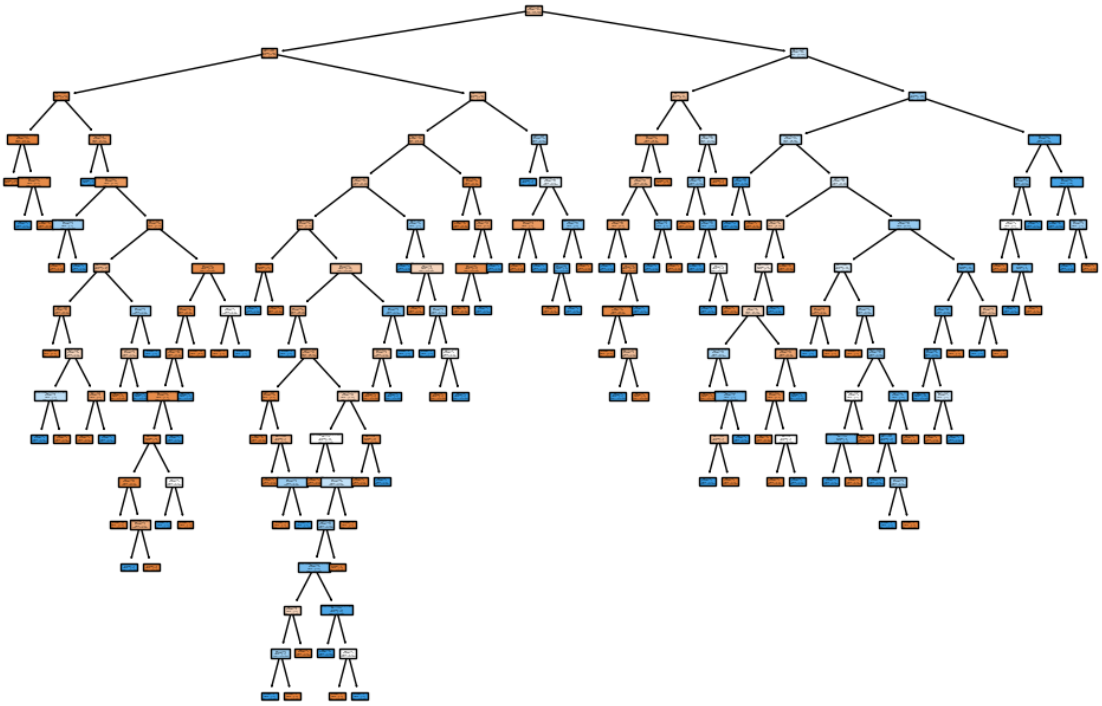


[23]:
```
decision_tree(0.2, 'gini')
```

```
Train-test split: 0.2
Value: Entropy: gini
*********************************************
Confusion Matrix :
```

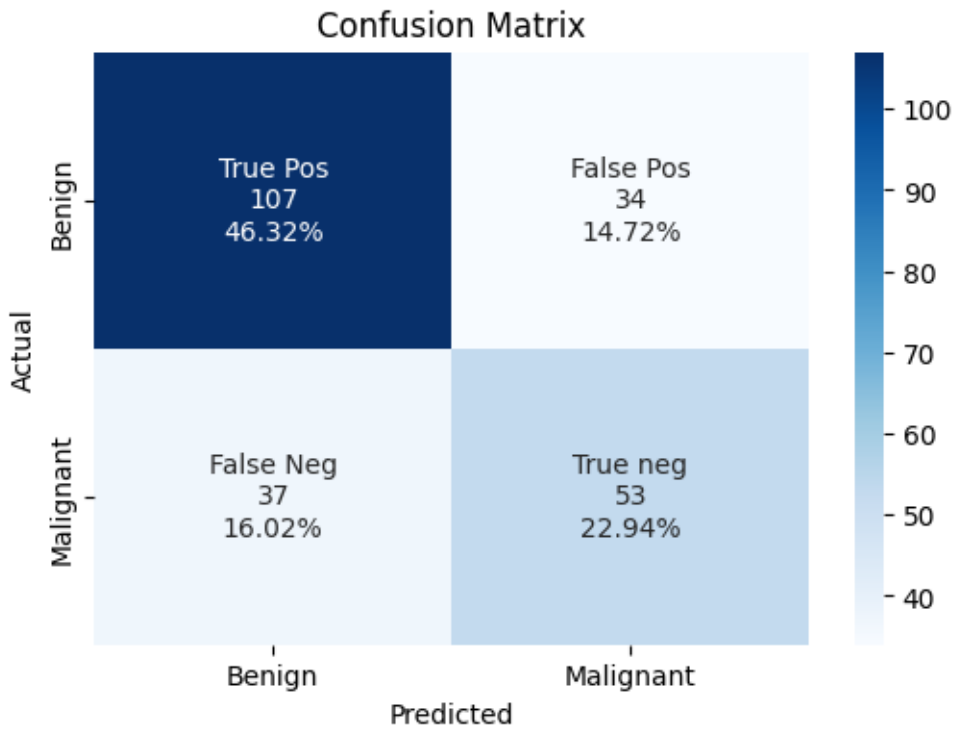



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.79      0.86      0.82        90
           1       0.77      0.67      0.72        64

    accuracy                           0.78       154
   macro avg       0.78      0.76      0.77       154
weighted avg       0.78      0.78      0.78       154
```

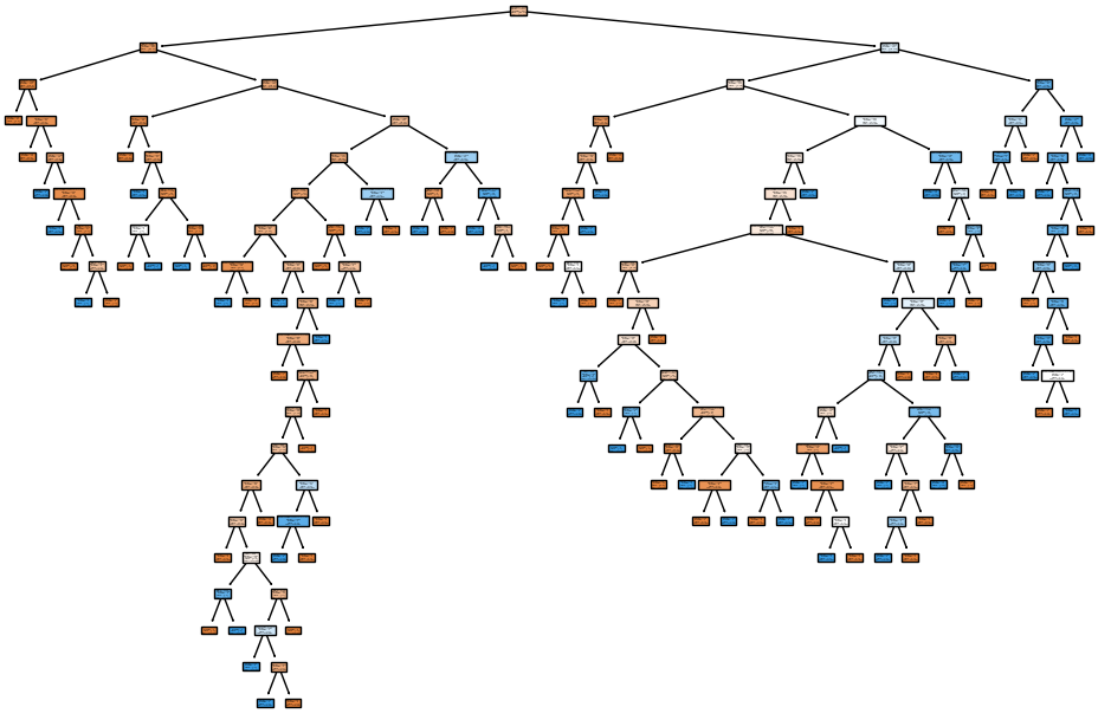


[24]:
```
decision_tree(0.3, 'entropy')
```

```
Train-test split: 0.3
Value: Entropy: entropy
*********************************************
Confusion Matrix :
```

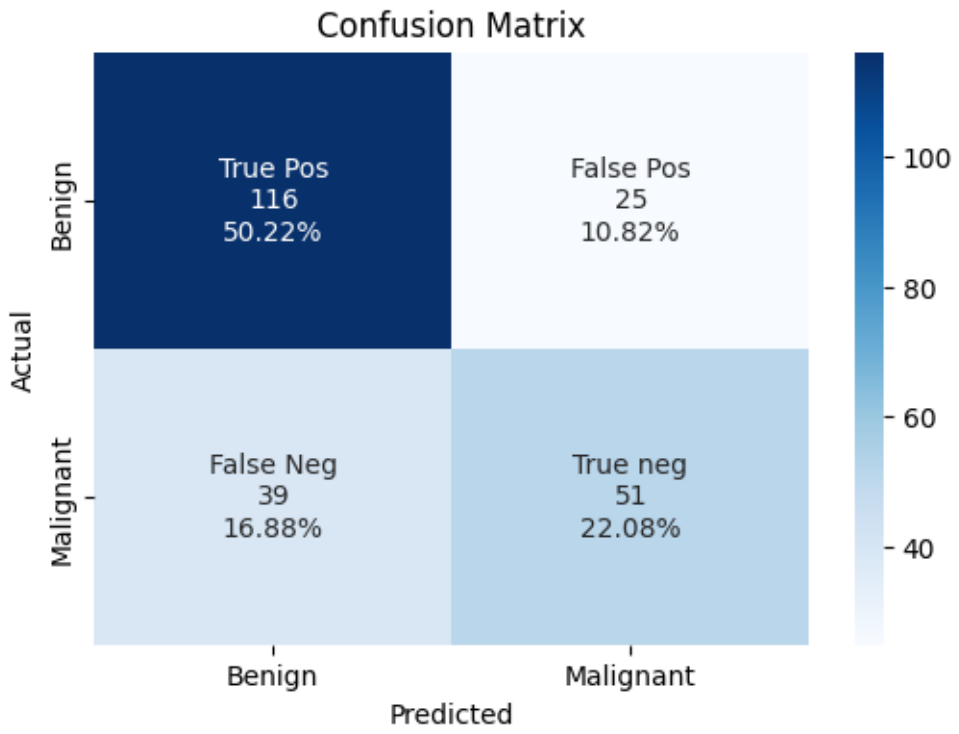



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.74      0.76      0.75       141
           1       0.61      0.59      0.60        90

    accuracy                           0.69       231
   macro avg       0.68      0.67      0.67       231
weighted avg       0.69      0.69      0.69       231
```

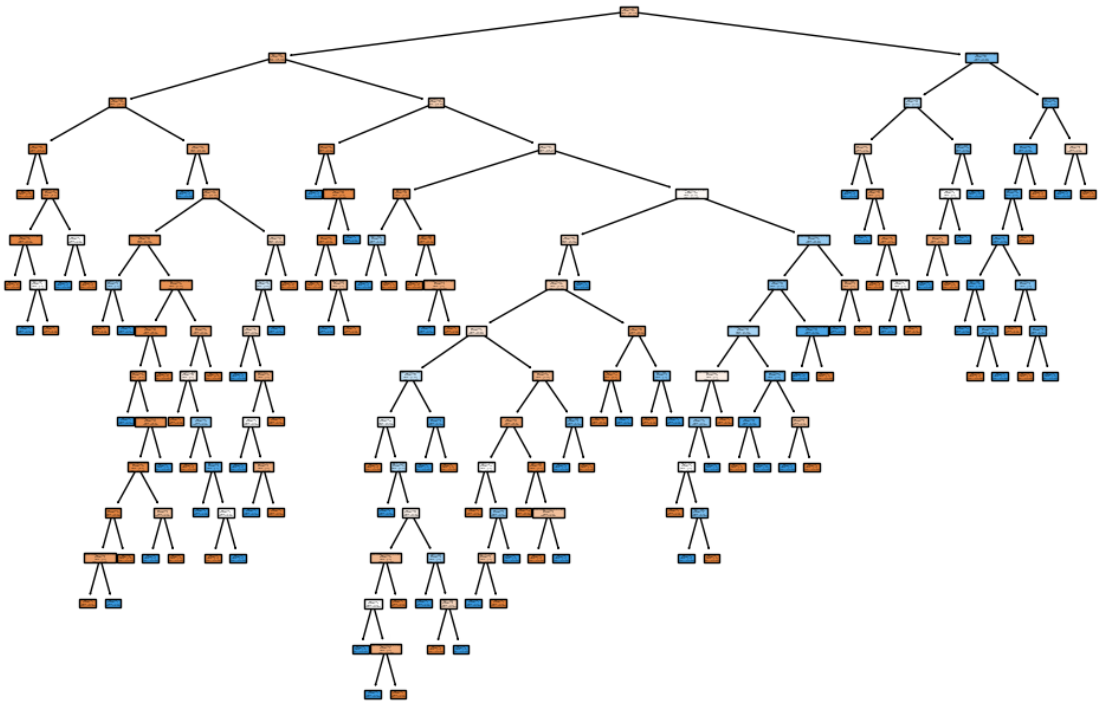


[25]:
```
decision_tree(0.3, 'gini')
```

```
Train-test split: 0.3
Value: Entropy: gini
*********************************************
Confusion Matrix :
```

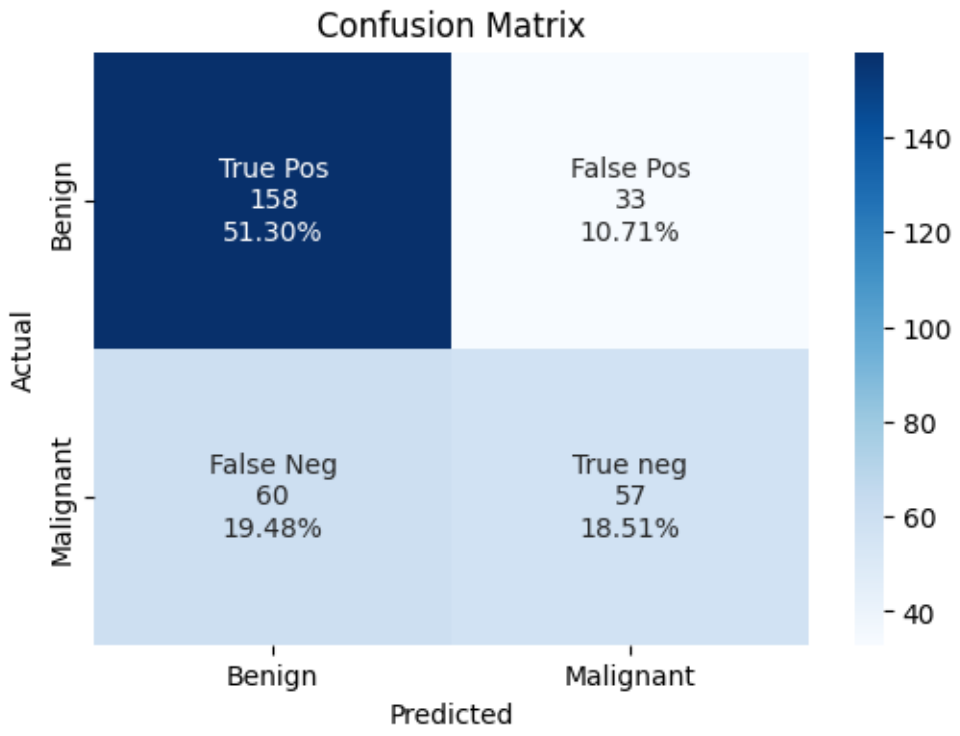



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.75      0.82      0.78       141
           1       0.67      0.57      0.61        90

    accuracy                           0.72       231
   macro avg       0.71      0.69      0.70       231
weighted avg       0.72      0.72      0.72       231
```

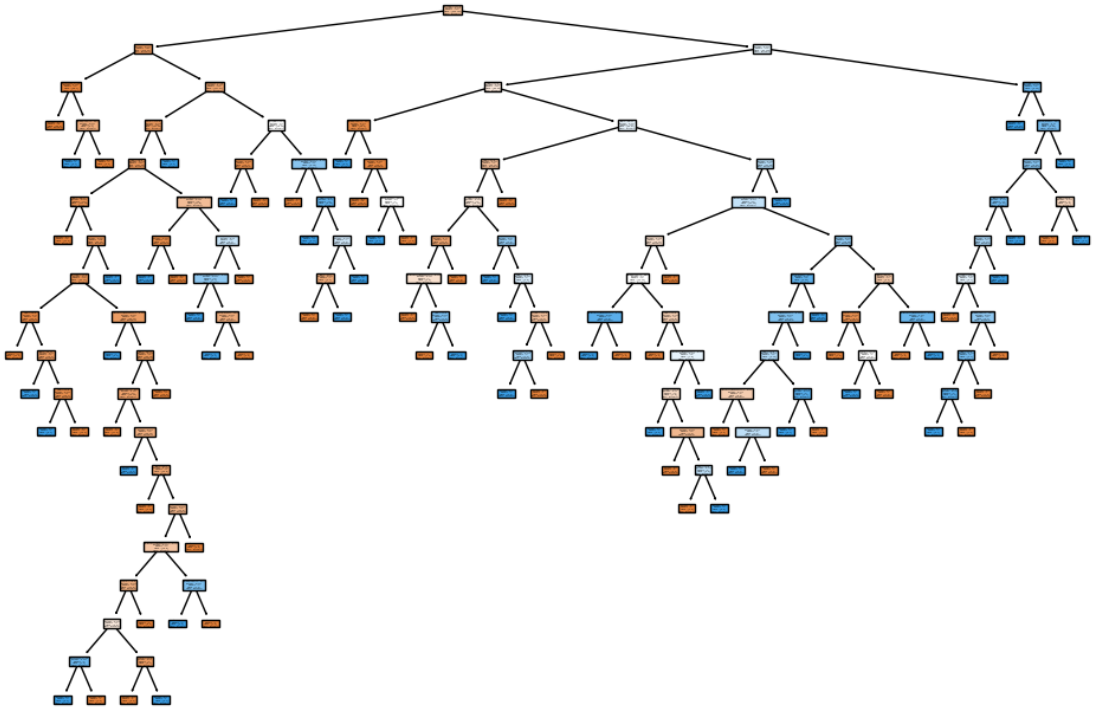


[26]:
```
decision_tree(0.4, 'entropy')
```

```
Train-test split: 0.4
Value: Entropy: entropy
*********************************************
Confusion Matrix :
```

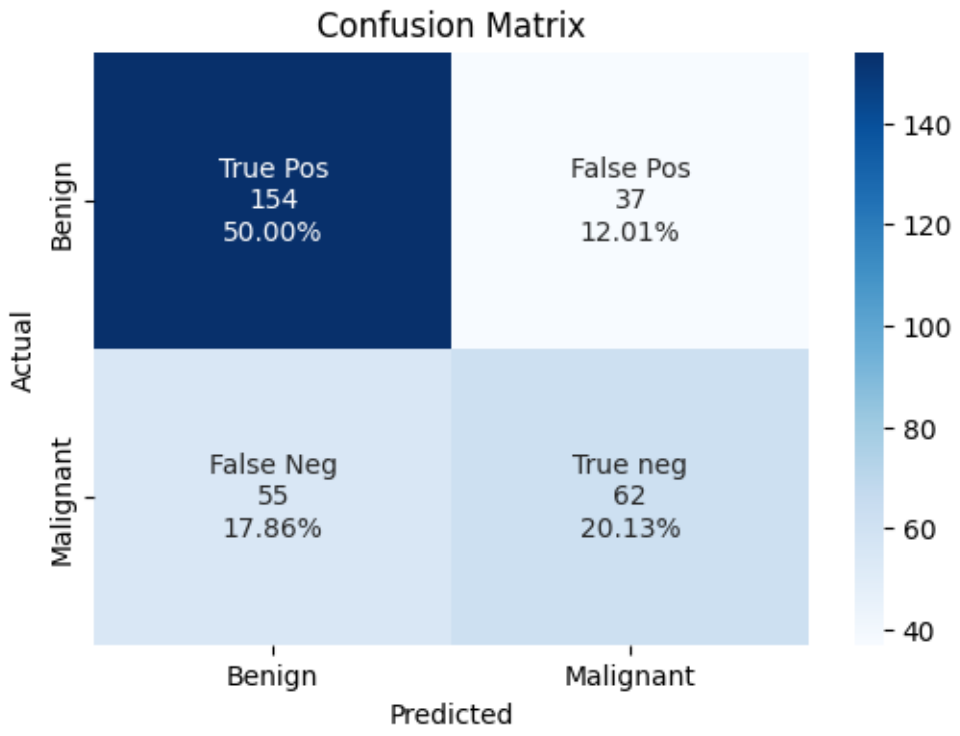



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.72      0.83      0.77       191
           1       0.63      0.49      0.55       117

    accuracy                           0.70       308
   macro avg       0.68      0.66      0.66       308
weighted avg       0.69      0.70      0.69       308
```

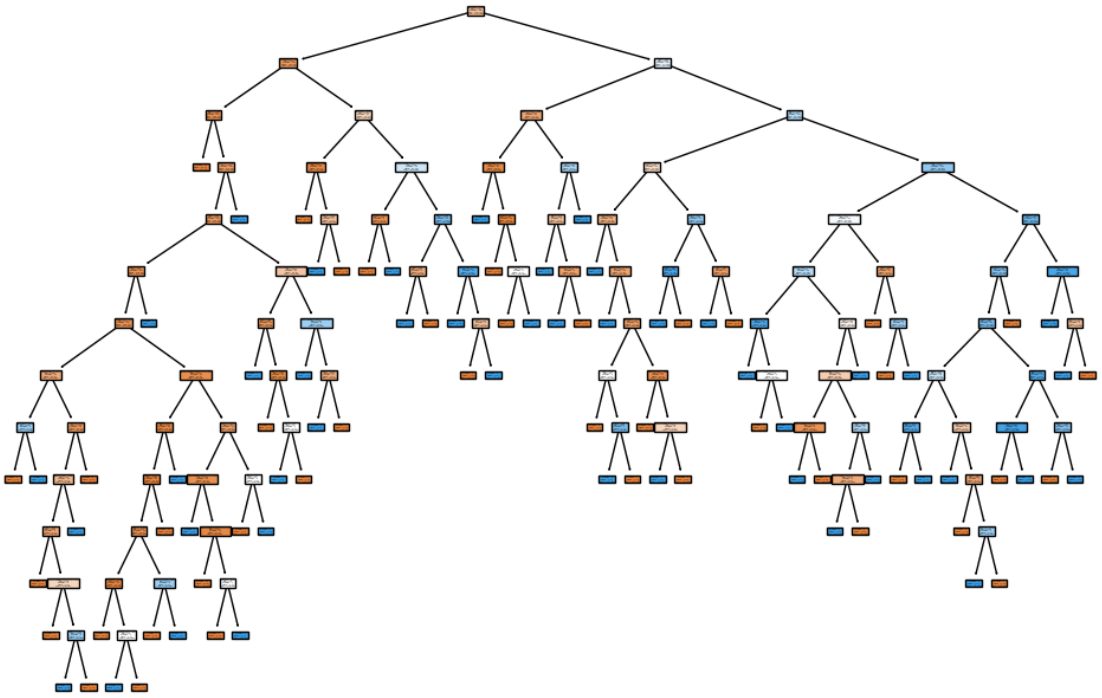


[27]:
```
decision_tree(0.4, 'gini')
```

```
Train-test split: 0.4
Value: Entropy: gini
*********************************************
Confusion Matrix :
```

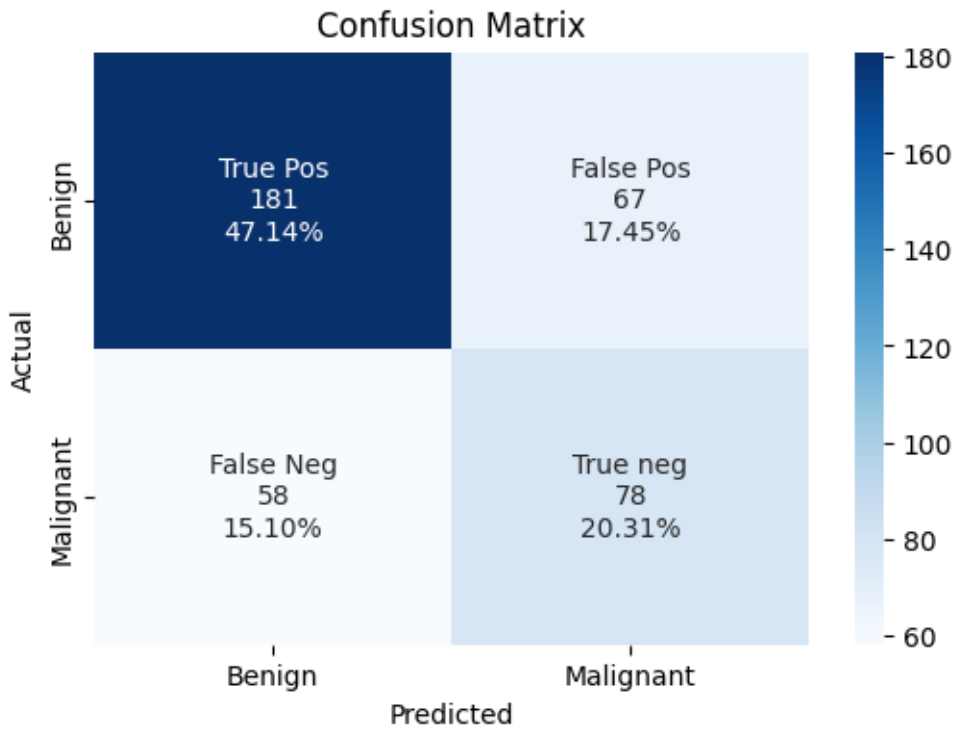



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.74      0.81      0.77       191
           1       0.63      0.53      0.57       117

    accuracy                           0.70       308
   macro avg       0.68      0.67      0.67       308
weighted avg       0.69      0.70      0.70       308
```

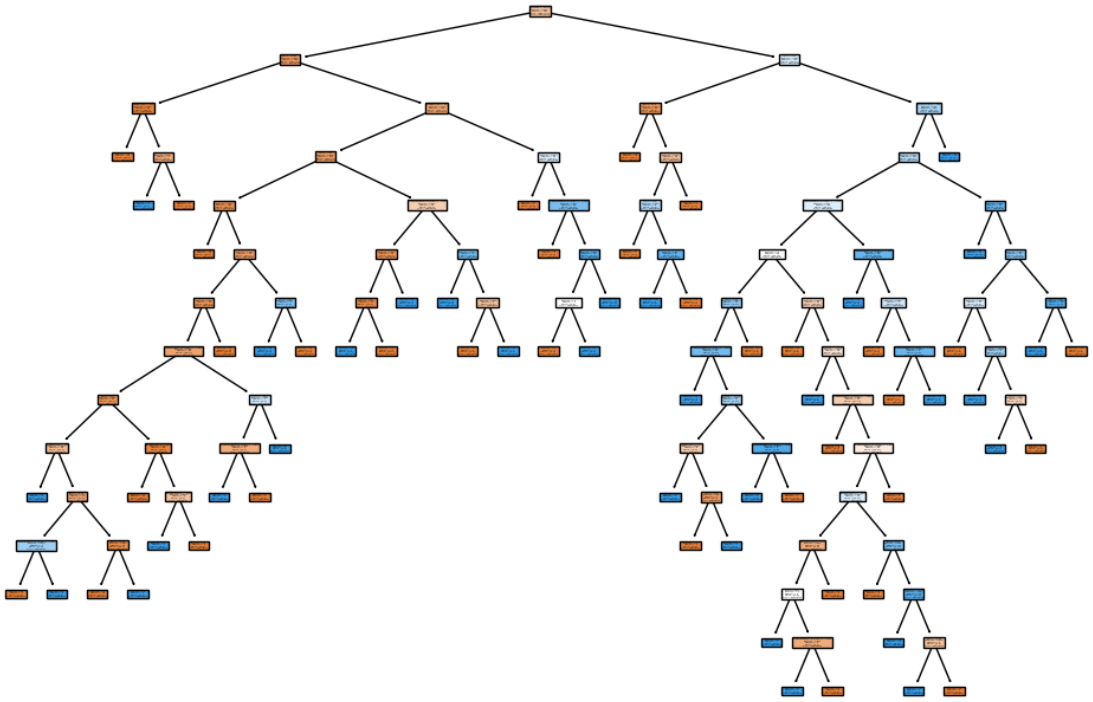


[28]:
```
decision_tree(0.5, 'entropy')
```

```
Train-test split: 0.5
Value: Entropy: entropy
*********************************************
Confusion Matrix :
```

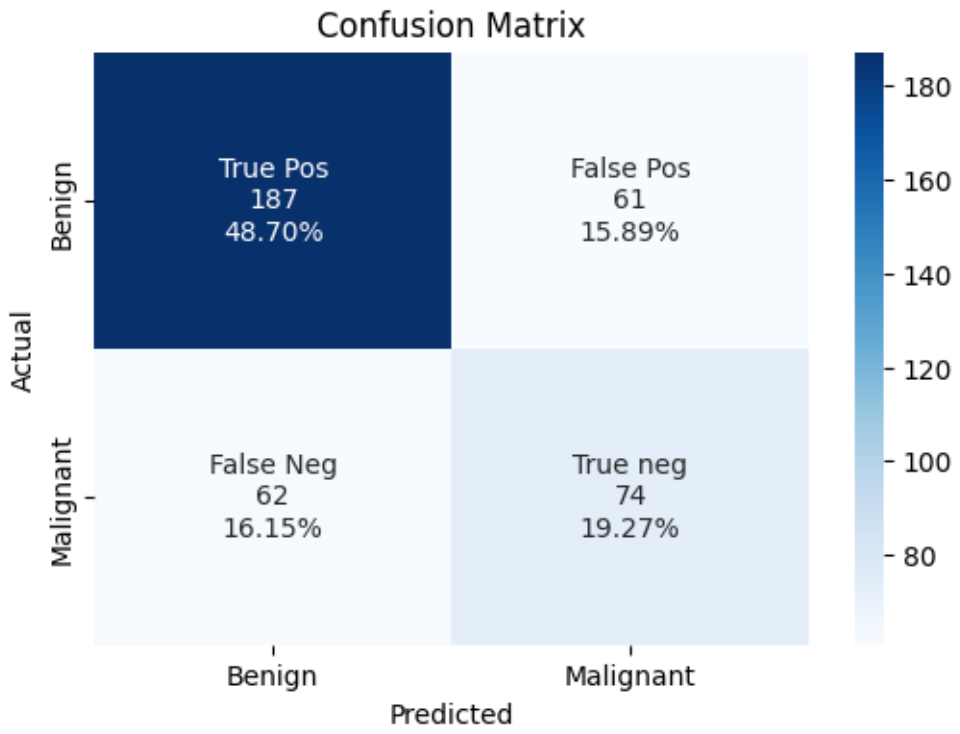



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.76      0.73      0.74       248
           1       0.54      0.57      0.56       136

    accuracy                           0.67       384
   macro avg       0.65      0.65      0.65       384
weighted avg       0.68      0.67      0.68       384
```

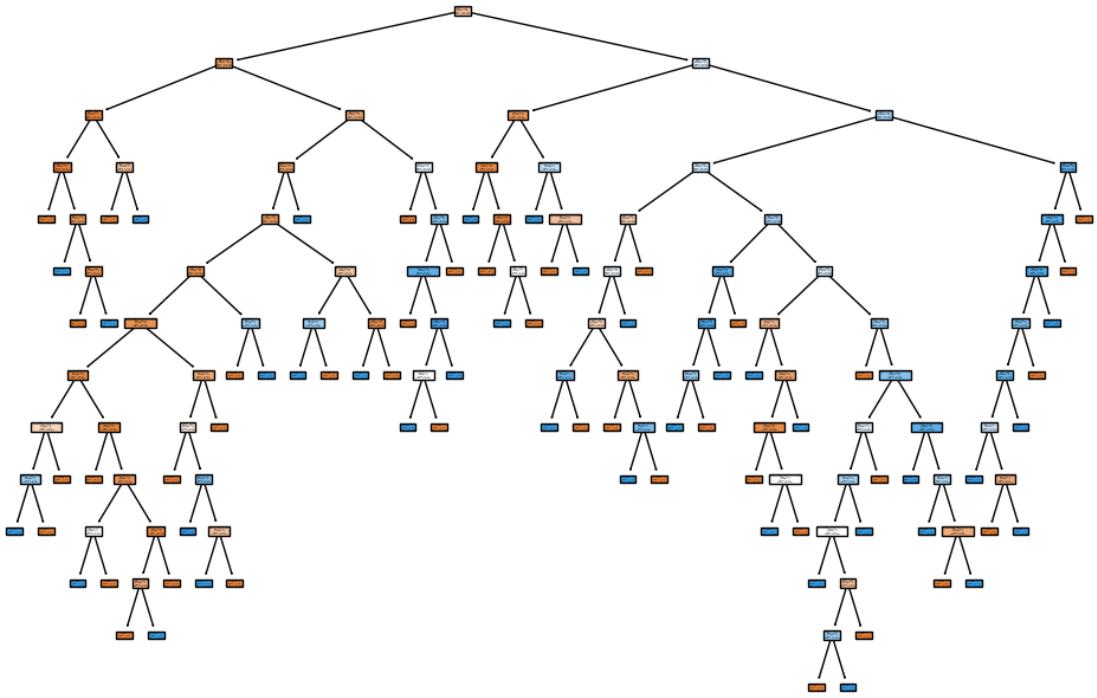


[29]:
```
decision_tree(0.5, 'gini')
```

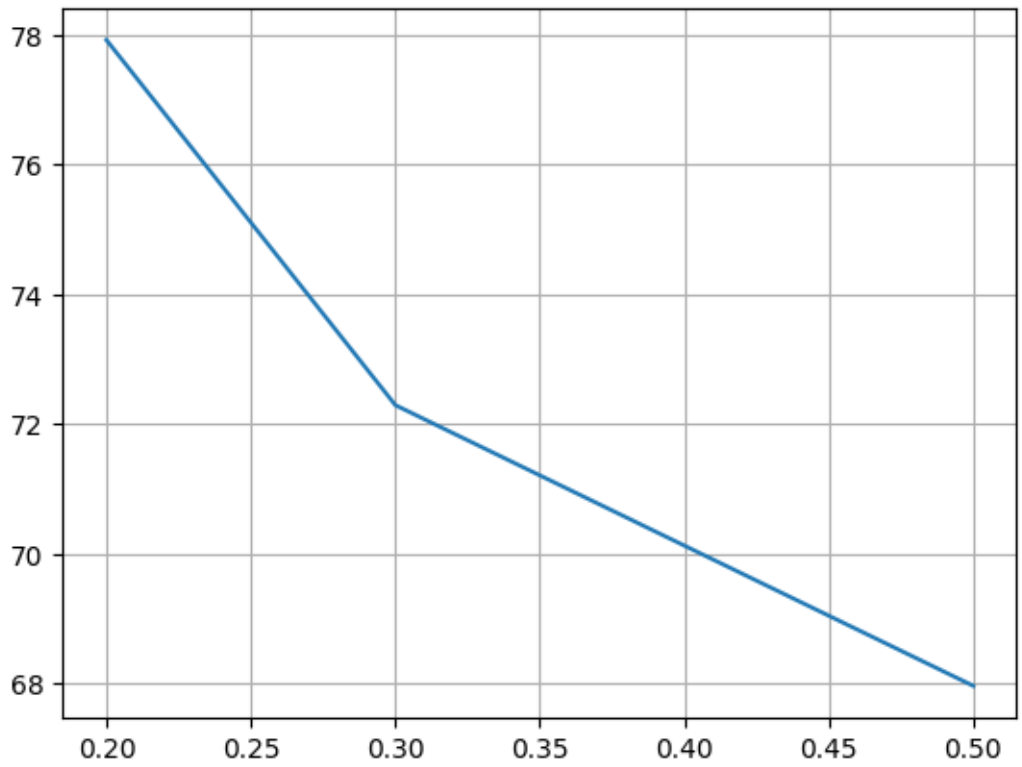
```
Train-test split: 0.5
Value: Entropy: gini
*********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.75      0.75      0.75       248
           1       0.55      0.54      0.55       136

    accuracy                           0.68       384
   macro avg       0.65      0.65      0.65       384
weighted avg       0.68      0.68      0.68       384
```

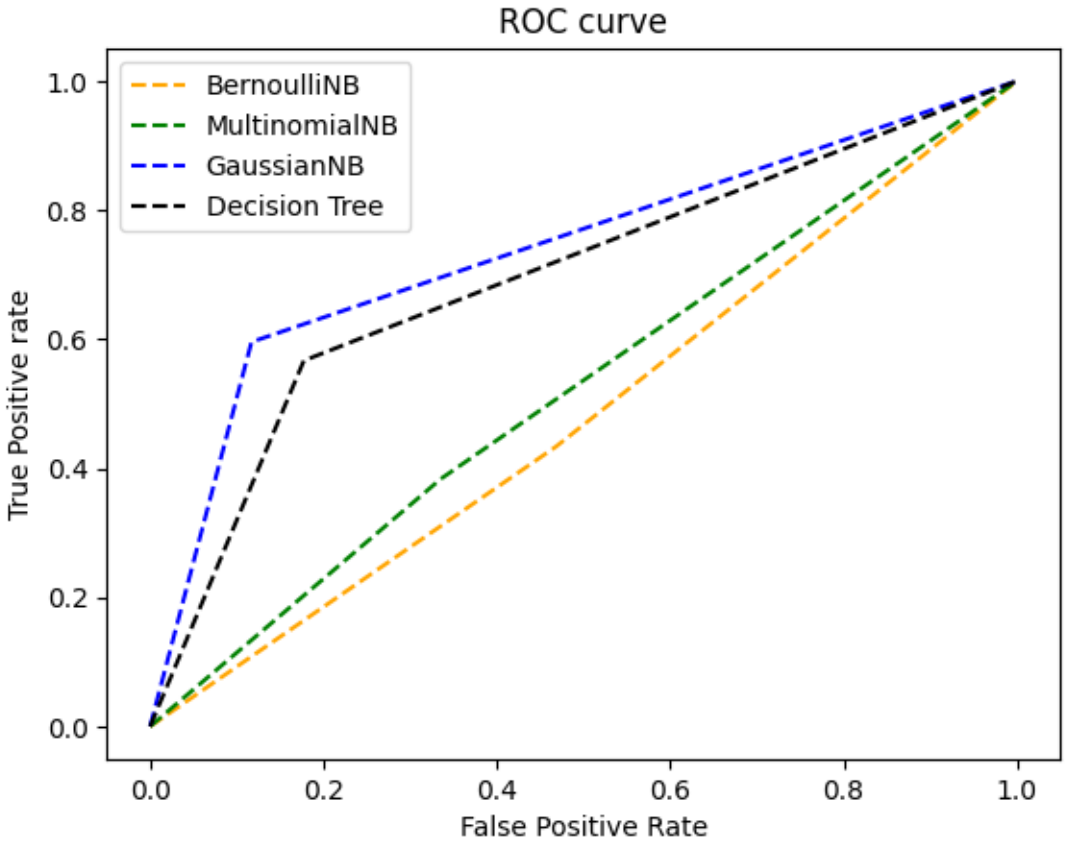


[30]:
```
x_points = [float(key) for key in dict_dtr]
y_points = [i*100 for i in dict_dtr.values()]
plt.plot(x_points, y_points)
plt.grid(True)
plt.show()
```



[31]:
```
from sklearn import metrics
def auc_roc():
    fpr1, tpr1, _1 = metrics.roc_curve(RocAucbnb['max']['y_test'],␣
 ↪RocAucbnb['max']['y_pred'], pos_label=1)
    fpr4, tpr4, _3 = metrics.roc_curve(RocAucmnb['max']['y_test'],␣
 ↪RocAucmnb['max']['y_pred'], pos_label=1)
    fpr2, tpr2, _2 = metrics.roc_curve(RocAucgnb['max']['y_test'],␣
 ↪RocAucgnb['max']['y_pred'], pos_label=1)
    fpr3, tpr3, _3 = metrics.roc_curve(RocAucdtr['max']['y_test'],␣
 ↪RocAucdtr['max']['y_pred'], pos_label=1)
    plt.plot(fpr1, tpr1, linestyle='--',color='orange', label='BernoulliNB')
    plt.plot(fpr4, tpr4, linestyle='--',color='green', label='MultinomialNB')
    plt.plot(fpr2, tpr2, linestyle='--', color='blue', label= 'GaussianNB')
    plt.plot(fpr3, tpr3, linestyle='--', color='black', label= 'Decision Tree')
    plt.title('ROC curve')
    # x label
    plt.xlabel('False Positive Rate')
    # y label
    plt.ylabel('True Positive rate')

    plt.legend(loc='best')
    plt.savefig('ROC',dpi=300)
```

```
    plt.show()
auc_roc()
```



sayadat-037-breast-cancer-dataset

August 16, 2023

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

0.0.1 Loading dataset from github

[ ]:
```
from sklearn.datasets import load_breast_cancer
breast_cancer = load_breast_cancer()
df = pd.DataFrame(breast_cancer.data, columns = breast_cancer.feature_names)
column_names = list(df.columns.values)
df.head()
```

[3]:
```
X = df.iloc[:][:]
y = breast_cancer["target"]
dict_bnb = {}
dict_mnb = {}
dict_gnb = {}
dict_dtr = {}
RocAucbnb = {}
RocAucmnb = {}
RocAucgnb = {}
RocAucdtr = {}
```

[4]:
```
def plot(y_test, y_pred):
  from sklearn.metrics import confusion_matrix
  import seaborn as sns

  print("Confusion Matrix : ")
  cf_matrix = confusion_matrix(y_test, y_pred)
  group_names = ['True Pos','False Pos','False Neg','True neg']
  group_counts = ["{0:0.0f}".format(value) for value in
                  cf_matrix.flatten()]
  group_percentages = ["{0:.2%}".format(value) for value in
                       cf_matrix.flatten()/np.sum(cf_matrix)]
  labels = [f"{v1}\n{v2}\n{v3}" for v1, v2, v3 in
            zip(group_names,group_counts,group_percentages)]
  labels = np.asarray(labels).reshape(2,2)
```

```
  plt.figure(figsize=(6, 4))
  sns.heatmap(cf_matrix, annot=labels, fmt='', cmap='Blues',␣
 ↪xticklabels=['Benign', 'Malignant'], yticklabels=['Benign', 'Malignant'])
  plt.xlabel('Predicted')
  plt.ylabel('Actual')
  plt.title('Confusion Matrix')
  plt.show()
  print("***********************************************")
```

[5]:
```
def reports(y_test, y_pred):
  from sklearn.metrics import classification_report
  plot(y_test, y_pred)
  print("***********************************************")
  print("Classification Evaluation : ")
  print(classification_report(y_test, y_pred, zero_division = 0))
```

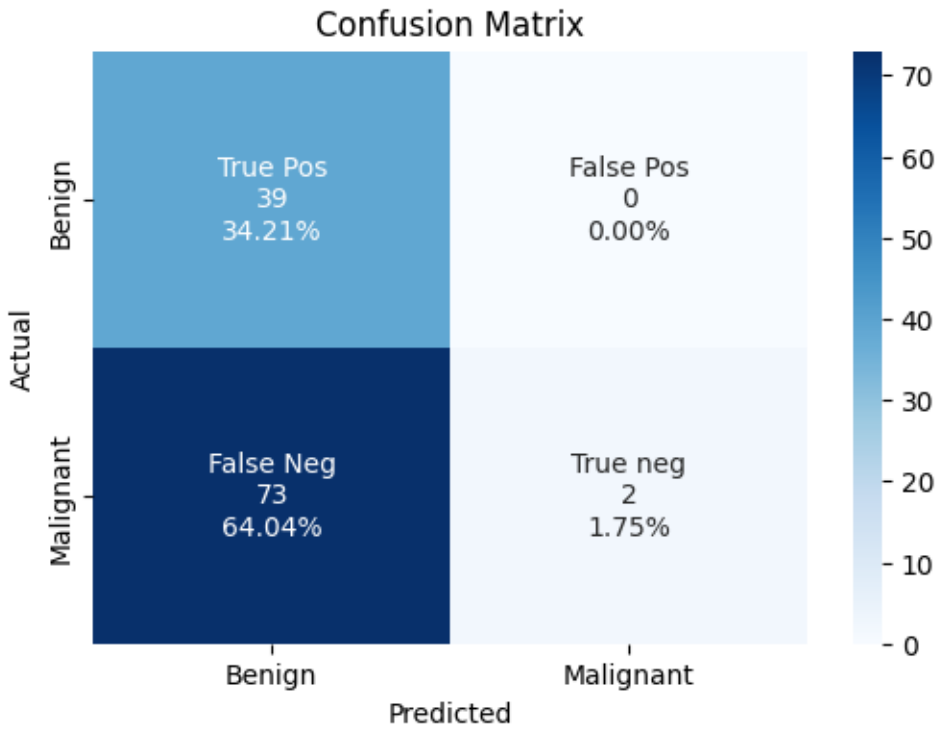
0.0.2 Classification using BernoulliNB Naive Bayes

[6]:
```
def FBouBernoulli(split, alpha_value = 1.0, binarize_value = 0.0,␣
 ↪fit_prior_value = False):
  from sklearn.naive_bayes import BernoulliNB
  from sklearn.metrics import accuracy_score
  from sklearn.model_selection import train_test_split
  from sklearn.preprocessing import StandardScaler
  #scaler = StandardScaler()
  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = split,␣
 ↪random_state=44)
  #scaler.fit_transform(X_train)
  #scaler.transform(X_test)
  classifier = BernoulliNB(alpha = alpha_value, binarize = binarize_value,␣
 ↪fit_prior = fit_prior_value)
  classifier.fit(X_train, y_train)
  y_pred = classifier.predict(X_test)
  print("Train-test split: " + str(split))
  print("value: alpha: "+str(alpha_value) + " binarize: " + str(binarize_value)␣
 ↪+ " fit_prior: " +str(fit_prior_value))
  print("***********************************************")
  accuracy = accuracy_score(y_test, y_pred)
  if str(split) in dict_bnb:
    dict_bnb[str(split)] = max(accuracy, dict_bnb[str(split)])
    if str(split) == '0.3' and accuracy > dict_bnb[str(split)]:
      RocAucbnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
  else:
    dict_bnb[str(split)] = accuracy
    if str(split) == '0.3':
      RocAucbnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
```

```
  reports(y_test, y_pred)
```

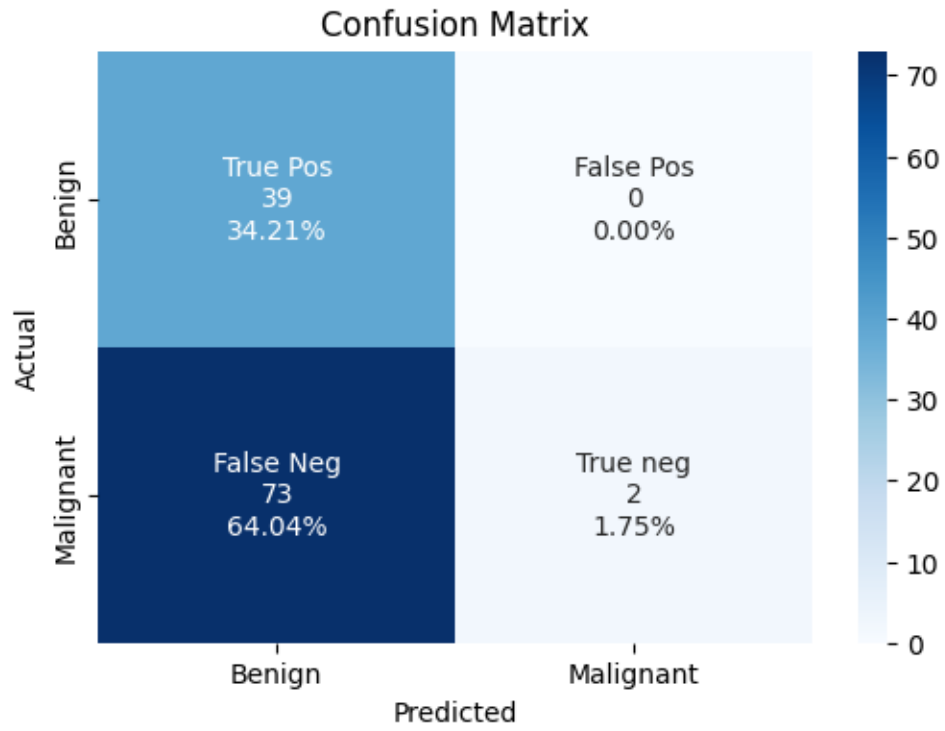
[7]:
```
## Train-Test split 0.2
FBouBernoulli(0.2)
FBouBernoulli(0.2, 2.8)
FBouBernoulli(0.2, 2.0, 2.8)
FBouBernoulli(0.2, 3.0, 3.3, True)
```

```
Train-test split: 0.2
value: alpha: 1.0 binarize: 0.0 fit_prior: False
**********************************************
Confusion Matrix :
```

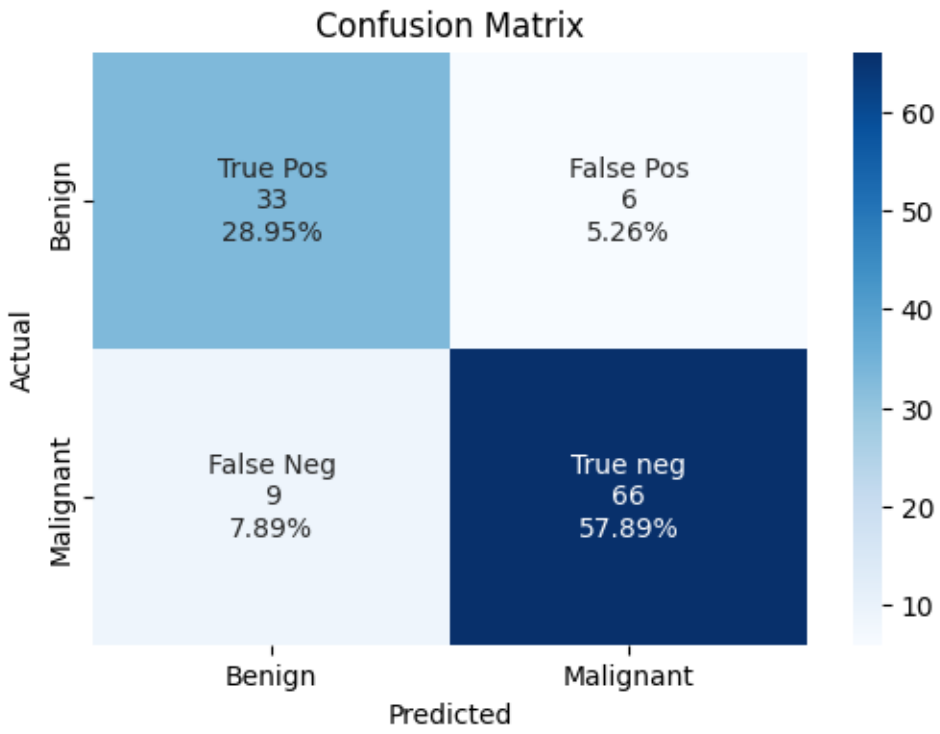



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.35      1.00      0.52        39
           1       1.00      0.03      0.05        75

    accuracy                           0.36       114
   macro avg       0.67      0.51      0.28       114
weighted avg       0.78      0.36      0.21       114
```

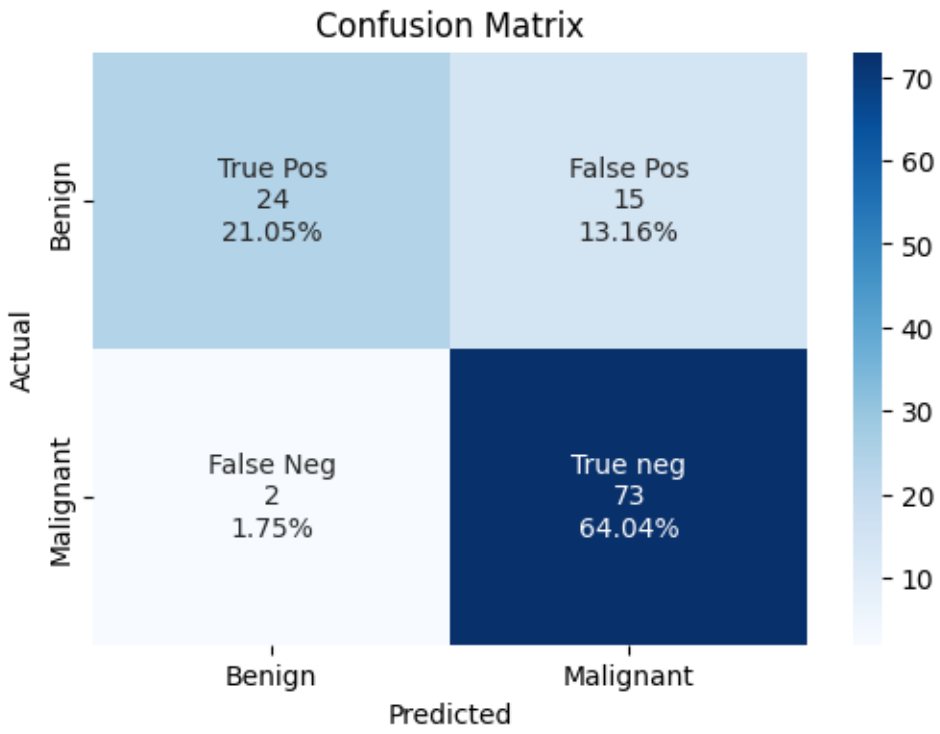
```
Train-test split: 0.2
value: alpha: 2.8 binarize: 0.0 fit_prior: False
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.35      1.00      0.52        39
           1       1.00      0.03      0.05        75

    accuracy                           0.36       114
   macro avg       0.67      0.51      0.28       114
weighted avg       0.78      0.36      0.21       114

Train-test split: 0.2
value: alpha: 2.0 binarize: 2.8 fit_prior: False
**********************************************
Confusion Matrix :
```



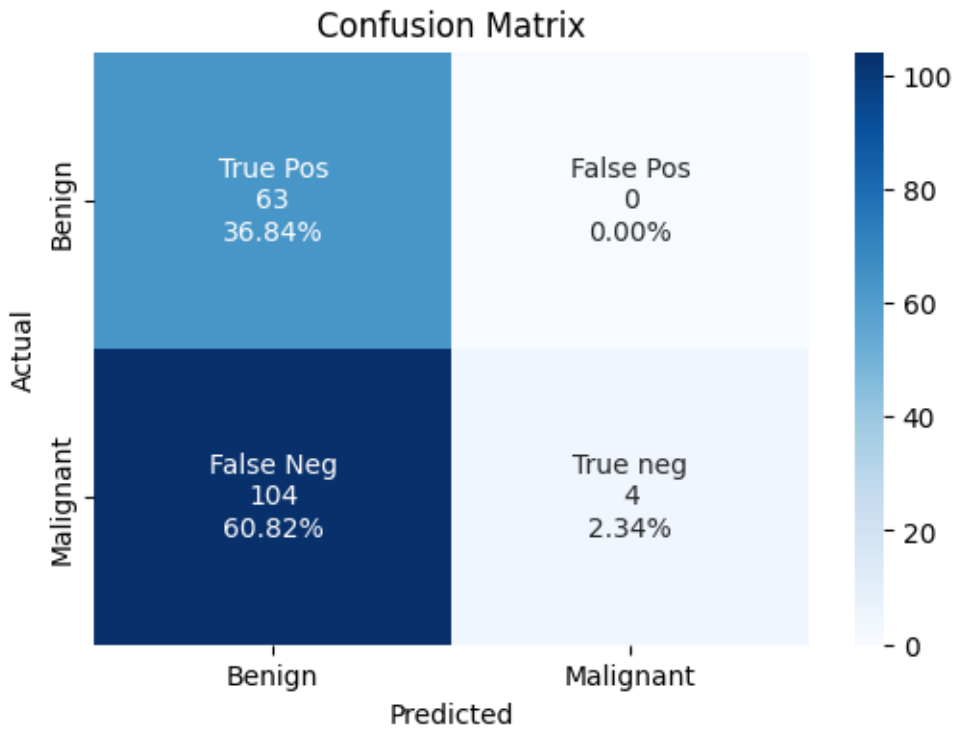

```
*********************************************
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.79      0.85      0.81        39
           1       0.92      0.88      0.90        75

    accuracy                           0.87       114
   macro avg       0.85      0.86      0.86       114
weighted avg       0.87      0.87      0.87       114

Train-test split: 0.2
value: alpha: 3.0 binarize: 3.3 fit_prior: True
*********************************************
Confusion Matrix :
```



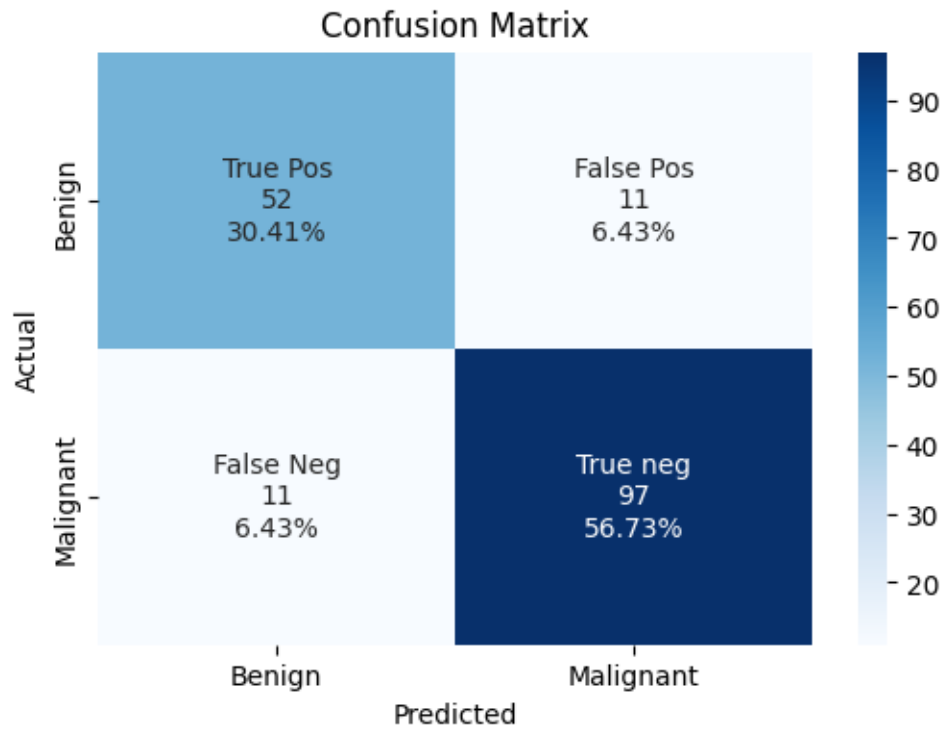

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.92      0.62      0.74        39
           1       0.83      0.97      0.90        75

    accuracy                           0.85       114
   macro avg       0.88      0.79      0.82       114
weighted avg       0.86      0.85      0.84       114
```

[8]:
```
## Train-Test split 0.3
FBouBernoulli(0.3)
FBouBernoulli(0.3, 1.0, 2.8)
FBouBernoulli(0.3, 1.0, 2.8, True)
```

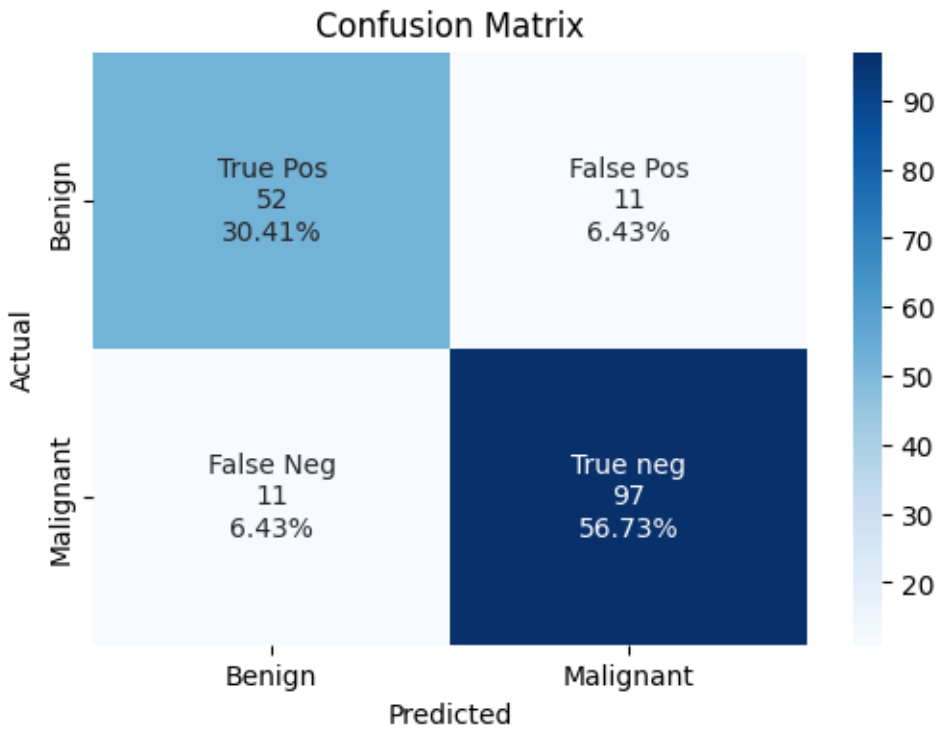
```
Train-test split: 0.3
value: alpha: 1.0 binarize: 0.0 fit_prior: False
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.38      1.00      0.55        63
           1       1.00      0.04      0.07       108

    accuracy                           0.39       171
   macro avg       0.69      0.52      0.31       171
weighted avg       0.77      0.39      0.25       171

Train-test split: 0.3
value: alpha: 1.0 binarize: 2.8 fit_prior: False
**********************************************
Confusion Matrix :
```



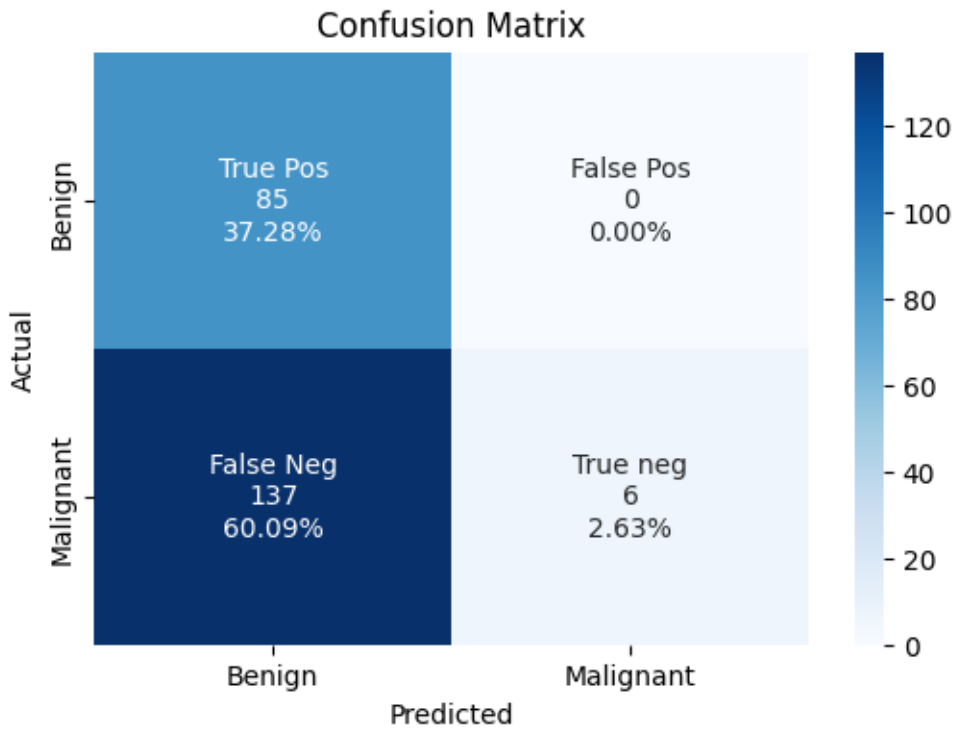

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.83      0.83      0.83        63
           1       0.90      0.90      0.90       108

    accuracy                           0.87       171
   macro avg       0.86      0.86      0.86       171
weighted avg       0.87      0.87      0.87       171

Train-test split: 0.3
value: alpha: 1.0 binarize: 2.8 fit_prior: True
**********************************************
Confusion Matrix :
```



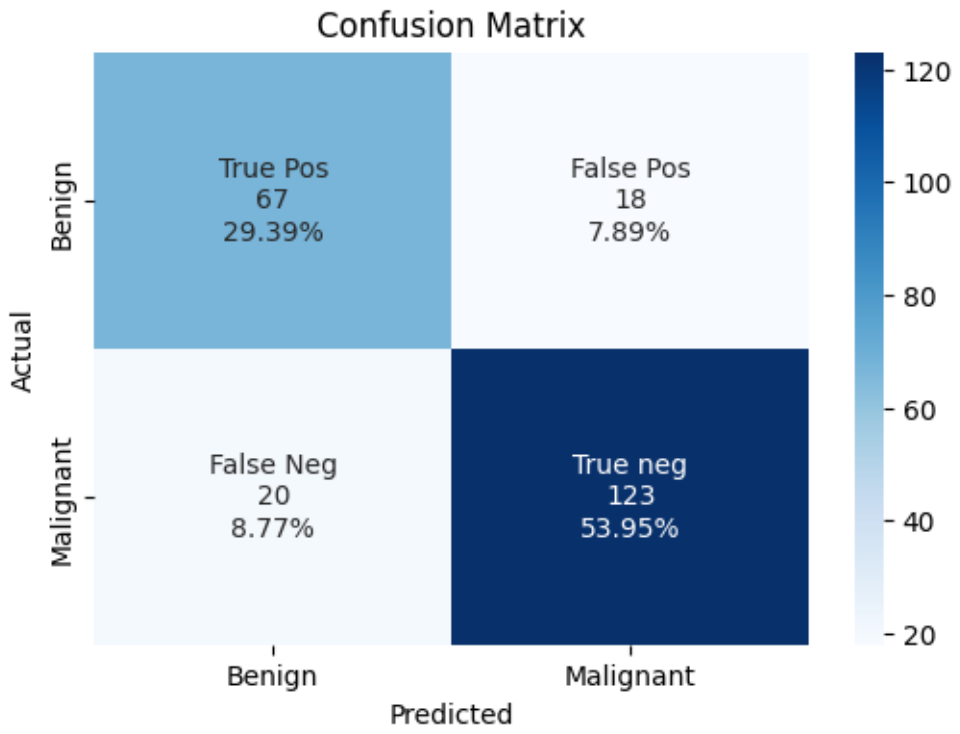

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.83      0.83      0.83        63
           1       0.90      0.90      0.90       108

    accuracy                           0.87       171
   macro avg       0.86      0.86      0.86       171
weighted avg       0.87      0.87      0.87       171
```

[9]:
```
## Train-Test split 0.4
FBouBernoulli(0.4)
FBouBernoulli(0.4, 1.0, 2.8)
FBouBernoulli(0.4, 1.0, 2.8, True)
```

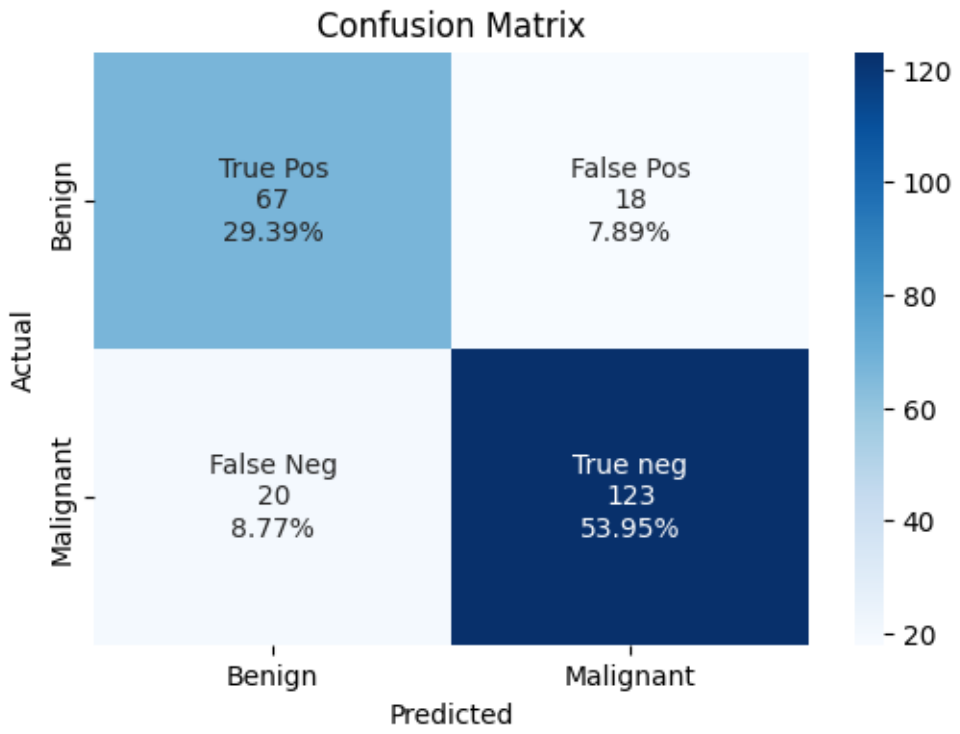
```
Train-test split: 0.4
value: alpha: 1.0 binarize: 0.0 fit_prior: False
**********************************************
Confusion Matrix :
```




```
*********************************************
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.38      1.00      0.55        85
           1       1.00      0.04      0.08       143

    accuracy                           0.40       228
   macro avg       0.69      0.52      0.32       228
weighted avg       0.77      0.40      0.26       228

Train-test split: 0.4
value: alpha: 1.0 binarize: 2.8 fit_prior: False
*********************************************
Confusion Matrix :
```



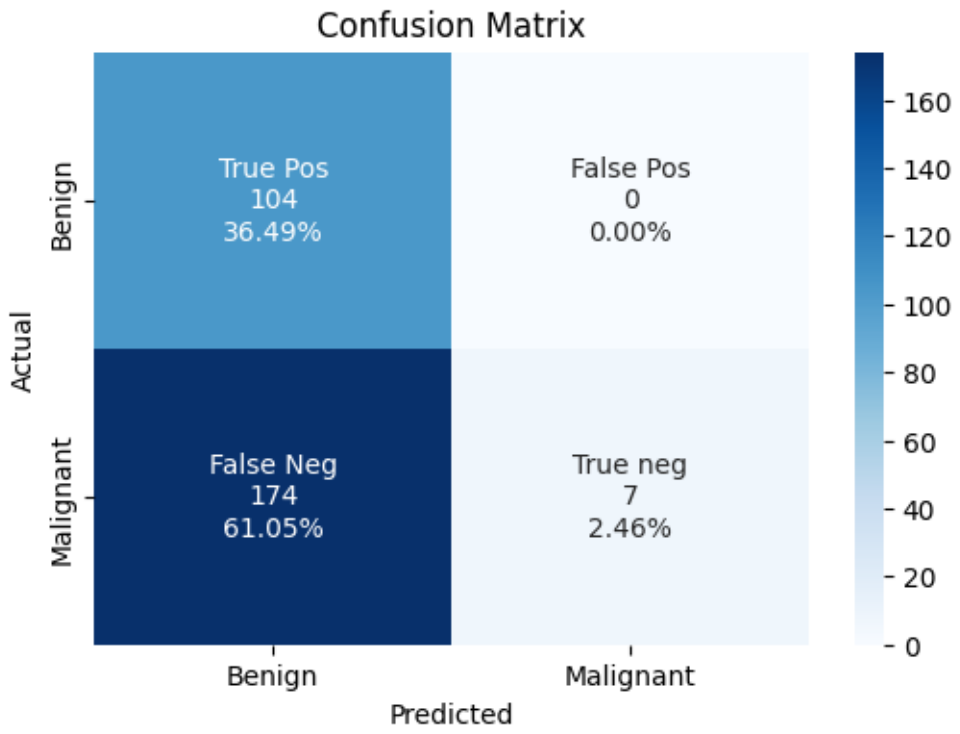

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.77      0.79      0.78        85
           1       0.87      0.86      0.87       143

    accuracy                           0.83       228
   macro avg       0.82      0.82      0.82       228
weighted avg       0.83      0.83      0.83       228

Train-test split: 0.4
value: alpha: 1.0 binarize: 2.8 fit_prior: True
**********************************************
Confusion Matrix :
```



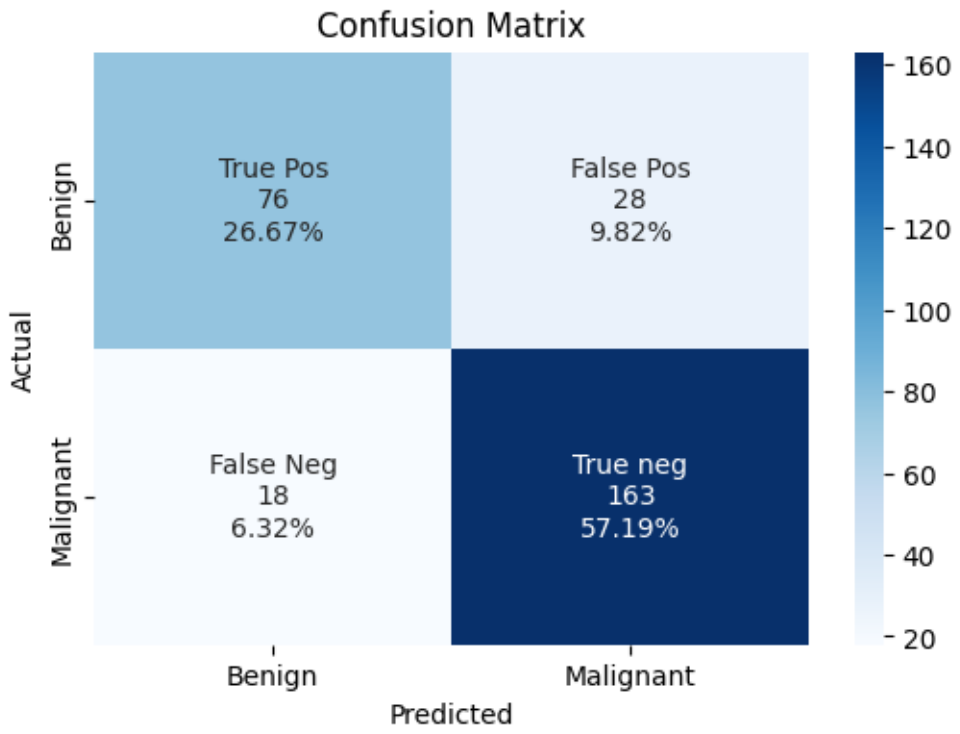

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.77      0.79      0.78        85
           1       0.87      0.86      0.87       143

    accuracy                           0.83       228
   macro avg       0.82      0.82      0.82       228
weighted avg       0.83      0.83      0.83       228
```

[10]:
```
## Train-Test split 0.5
FBouBernoulli(0.5)
FBouBernoulli(0.5, 1.0, 2.9)
FBouBernoulli(0.5, 1.0, 2.9, True)
```

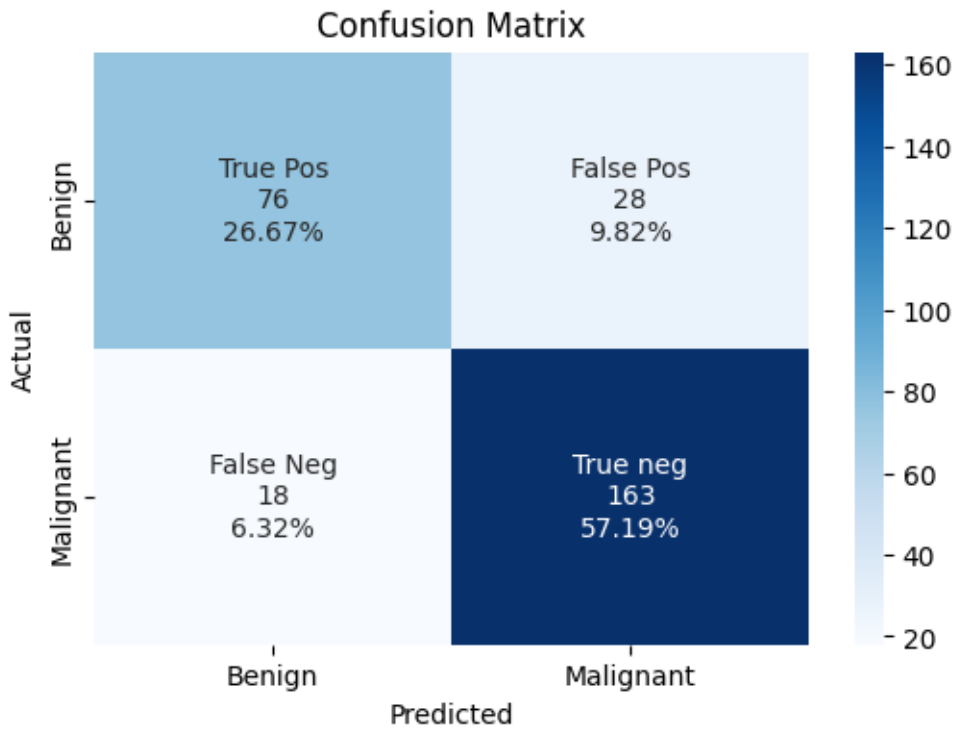
```
Train-test split: 0.5
value: alpha: 1.0 binarize: 0.0 fit_prior: False
**********************************************
Confusion Matrix :
```




```
*********************************************
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.37      1.00      0.54       104
           1       1.00      0.04      0.07       181

    accuracy                           0.39       285
   macro avg       0.69      0.52      0.31       285
weighted avg       0.77      0.39      0.25       285

Train-test split: 0.5
value: alpha: 1.0 binarize: 2.9 fit_prior: False
*********************************************
Confusion Matrix :
```

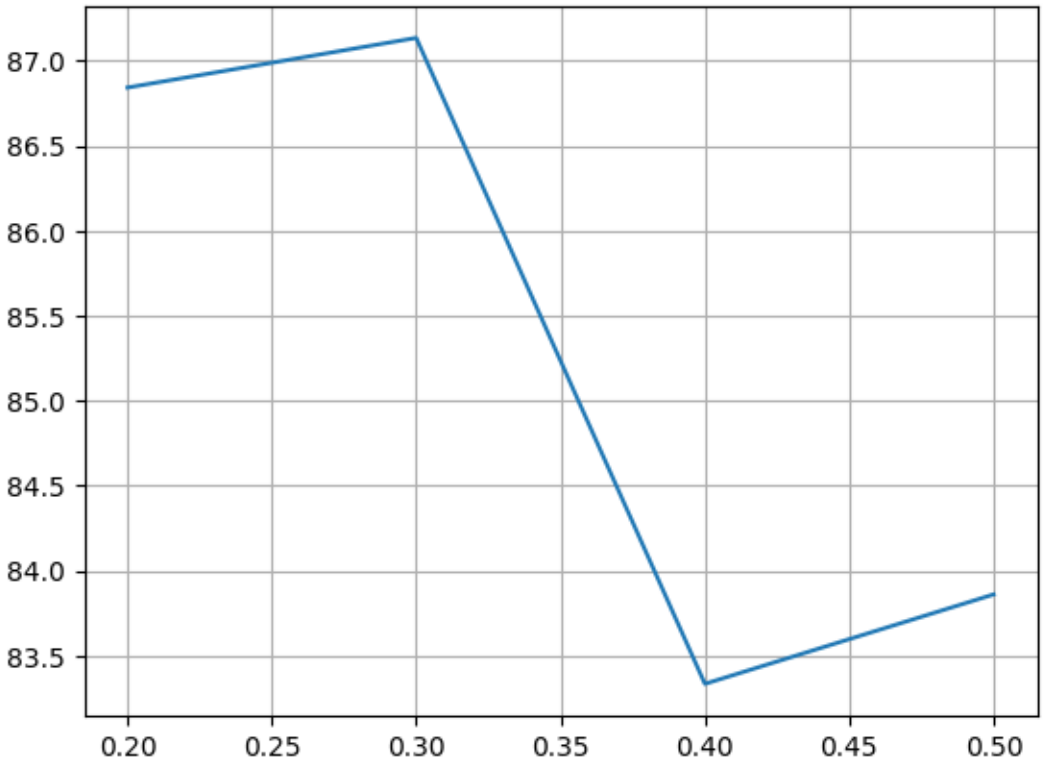



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.81      0.73      0.77       104
           1       0.85      0.90      0.88       181

    accuracy                           0.84       285
   macro avg       0.83      0.82      0.82       285
weighted avg       0.84      0.84      0.84       285

Train-test split: 0.5
value: alpha: 1.0 binarize: 2.9 fit_prior: True
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.81      0.73      0.77       104
           1       0.85      0.90      0.88       181

    accuracy                           0.84       285
   macro avg       0.83      0.82      0.82       285
weighted avg       0.84      0.84      0.84       285
```

[11]:
```
x_points = [float(key) for key in dict_bnb]
y_points = [i*100 for i in dict_bnb.values()]
plt.plot(x_points, y_points)
plt.grid(True)
plt.show()
```

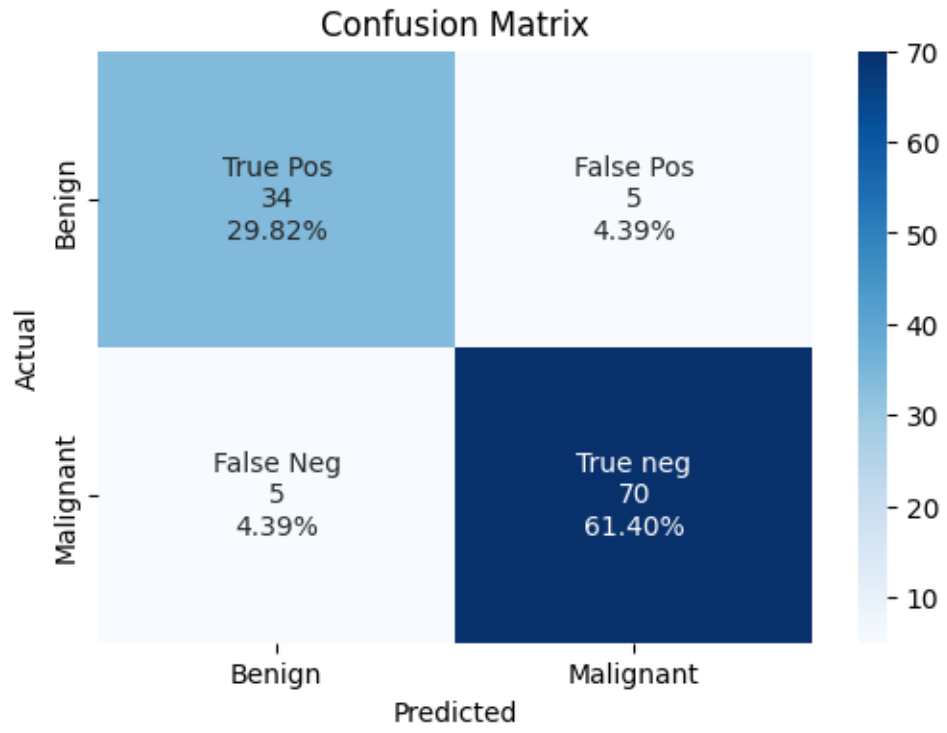


0.1 Classification using Multinomial Naive Bayes

[12]:
```
def FMultinomial(split, alpha_value = 1.0):
  from sklearn.naive_bayes import MultinomialNB
  from sklearn.metrics import accuracy_score
  from sklearn.model_selection import train_test_split
  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = split)
  classifier = MultinomialNB(alpha = alpha_value)
  classifier.fit(X_train, y_train)
  y_pred = classifier.predict(X_test)
  print("Train-test split: " + str(split))
  print("value: alpha: "+str(alpha_value))
  print("*********************************************")
  accuracy = accuracy_score(y_test, y_pred)
  if str(split) in dict_mnb:
    dict_mnb[str(split)] = max(accuracy, dict_mnb[str(split)])
    if str(split) == '0.3' and accuracy > dict_mnb[str(split)]:
      RocAucmnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
  else:
    dict_mnb[str(split)] = accuracy
    if str(split) == '0.3':
      RocAucmnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
```

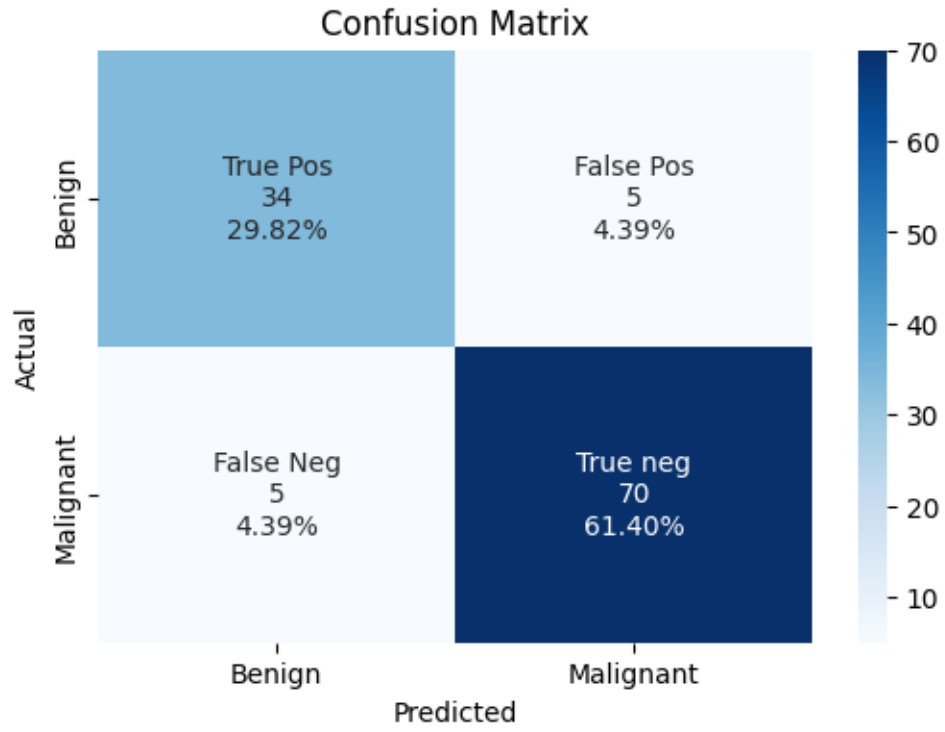
```
  reports(y_test, y_pred)
  reports(y_test, y_pred)

## Train-Test split 0.2
FMultinomial(0.2)
FMultinomial(0.2, 1.8)
#92, 91, 89, 91
```

```
Train-test split: 0.2
value: alpha: 1.0
**********************************************
Confusion Matrix :
```

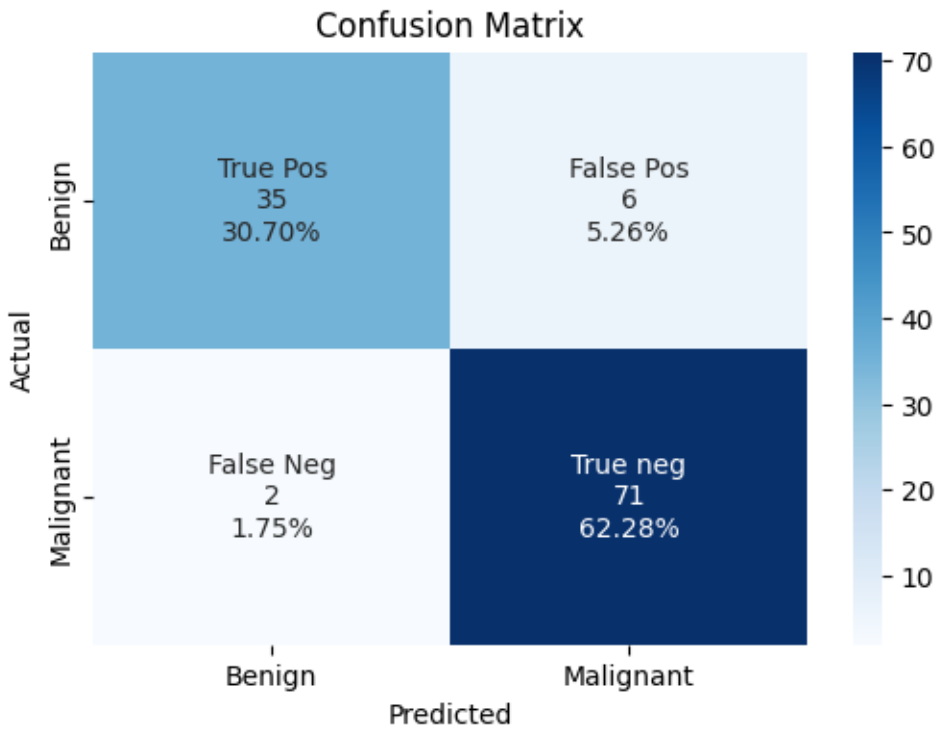



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.87      0.87      0.87        39
           1       0.93      0.93      0.93        75

    accuracy                           0.91       114
   macro avg       0.90      0.90      0.90       114
weighted avg       0.91      0.91      0.91       114
```

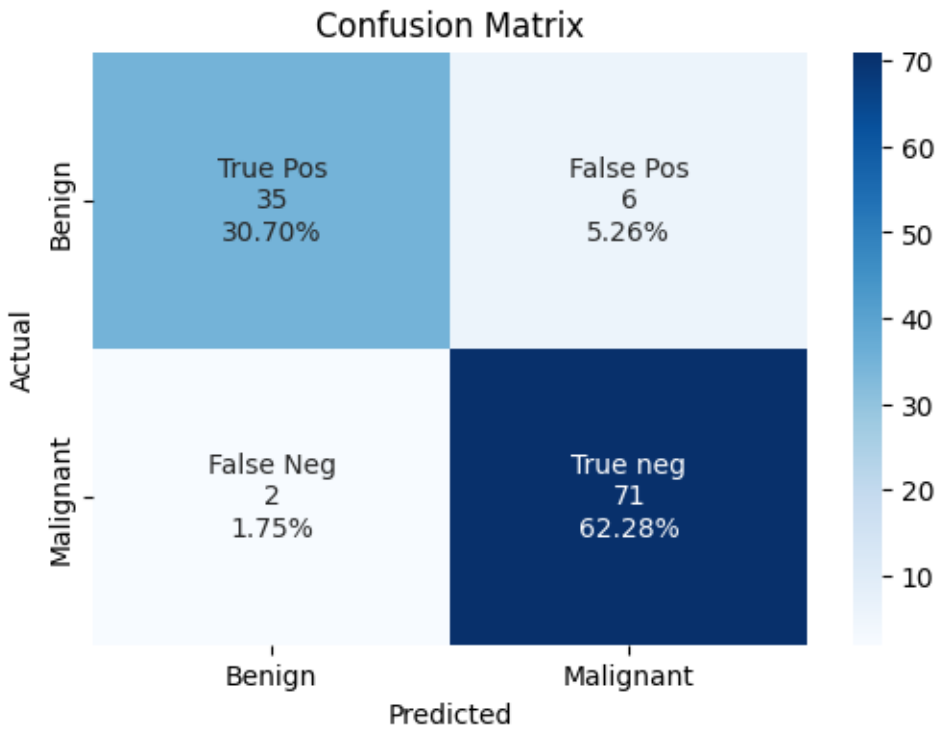
Confusion Matrix :




```
*********************************************
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.87      0.87      0.87        39
           1       0.93      0.93      0.93        75

    accuracy                           0.91       114
   macro avg       0.90      0.90      0.90       114
weighted avg       0.91      0.91      0.91       114

Train-test split: 0.2
value: alpha: 1.8
*********************************************
Confusion Matrix :
```



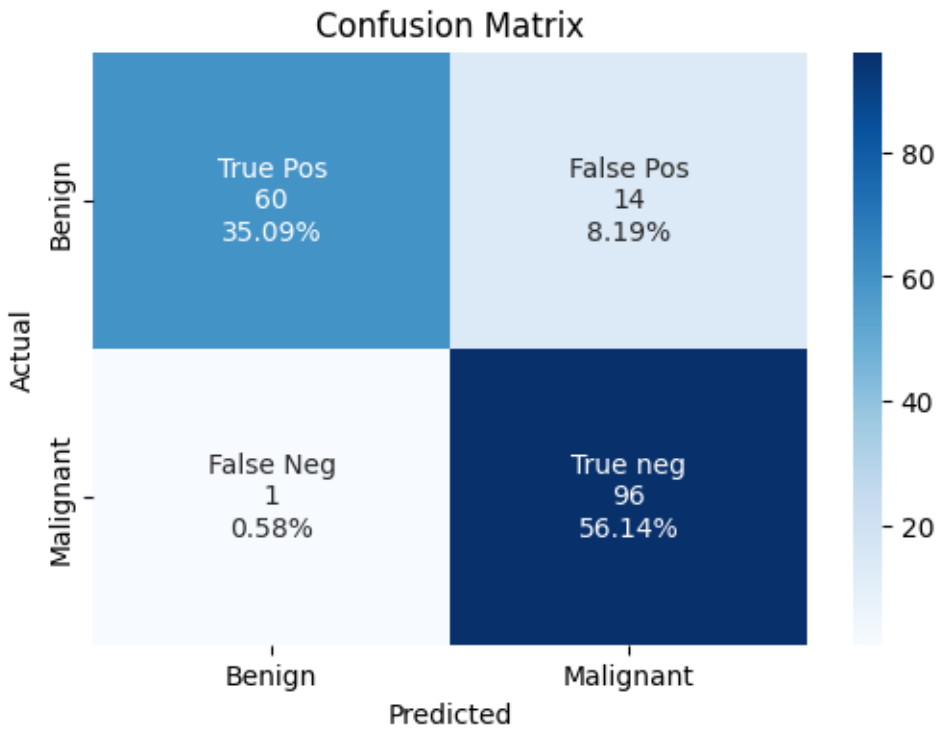

```
*********************************************
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.95      0.85      0.90        41
           1       0.92      0.97      0.95        73

    accuracy                           0.93       114
   macro avg       0.93      0.91      0.92       114
weighted avg       0.93      0.93      0.93       114

Confusion Matrix :
```



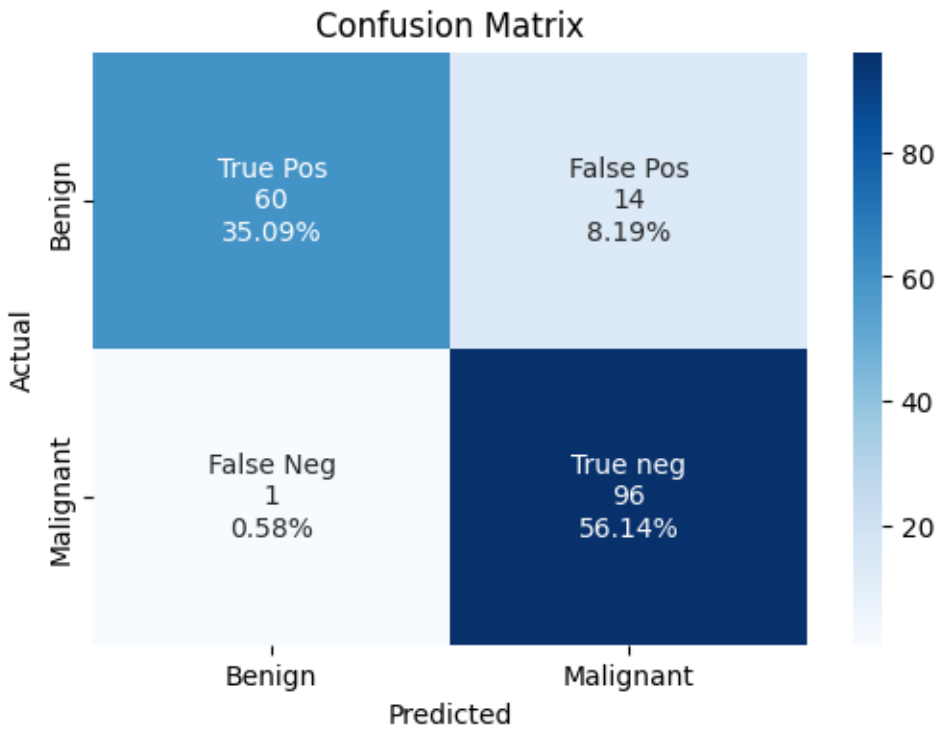

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.95      0.85      0.90        41
           1       0.92      0.97      0.95        73

    accuracy                           0.93       114
   macro avg       0.93      0.91      0.92       114
weighted avg       0.93      0.93      0.93       114
```

[13]:
```
## Train-Test split 0.3
FMultinomial(0.3)
FMultinomial(0.3, 2.5)
```

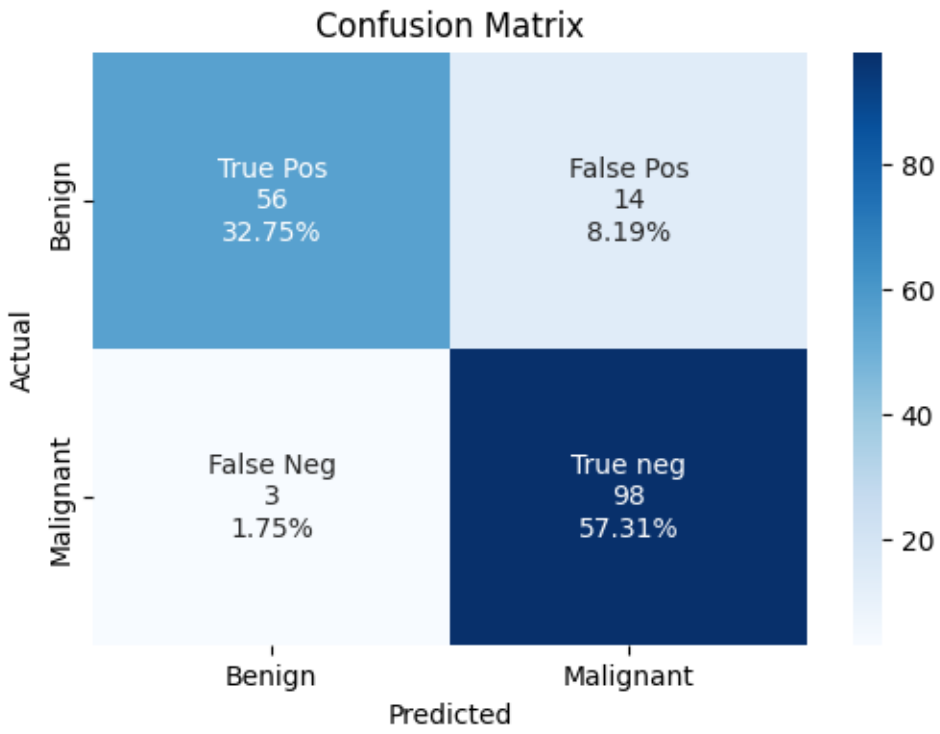
```
Train-test split: 0.3
value: alpha: 1.0
**********************************************
Confusion Matrix :
```



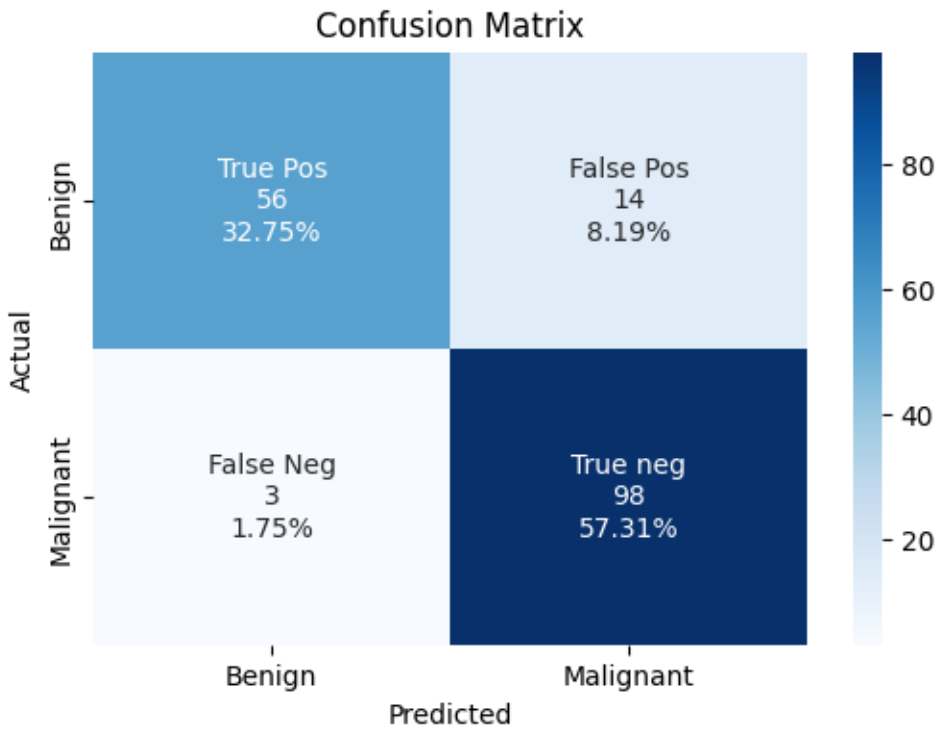

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.98      0.81      0.89        74
           1       0.87      0.99      0.93        97

    accuracy                           0.91       171
   macro avg       0.93      0.90      0.91       171
weighted avg       0.92      0.91      0.91       171

Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.98      0.81      0.89        74
           1       0.87      0.99      0.93        97

    accuracy                           0.91       171
   macro avg       0.93      0.90      0.91       171
weighted avg       0.92      0.91      0.91       171

Train-test split: 0.3
value: alpha: 2.5
**********************************************
Confusion Matrix :
```



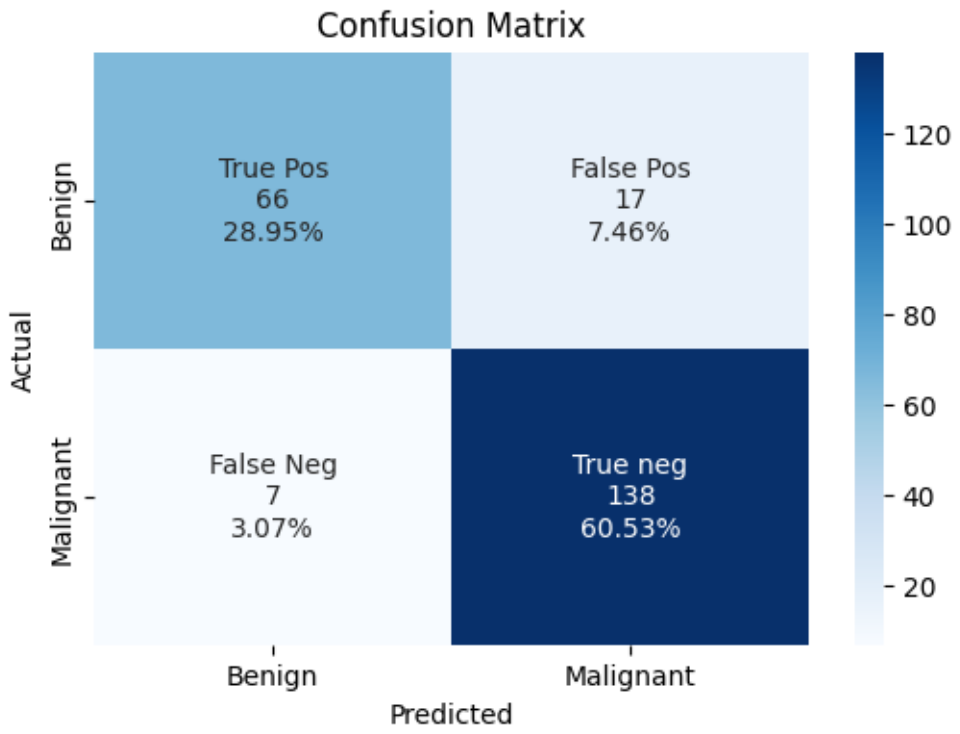

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.95      0.80      0.87        70
           1       0.88      0.97      0.92       101

    accuracy                           0.90       171
   macro avg       0.91      0.89      0.89       171
weighted avg       0.91      0.90      0.90       171

Confusion Matrix :
```



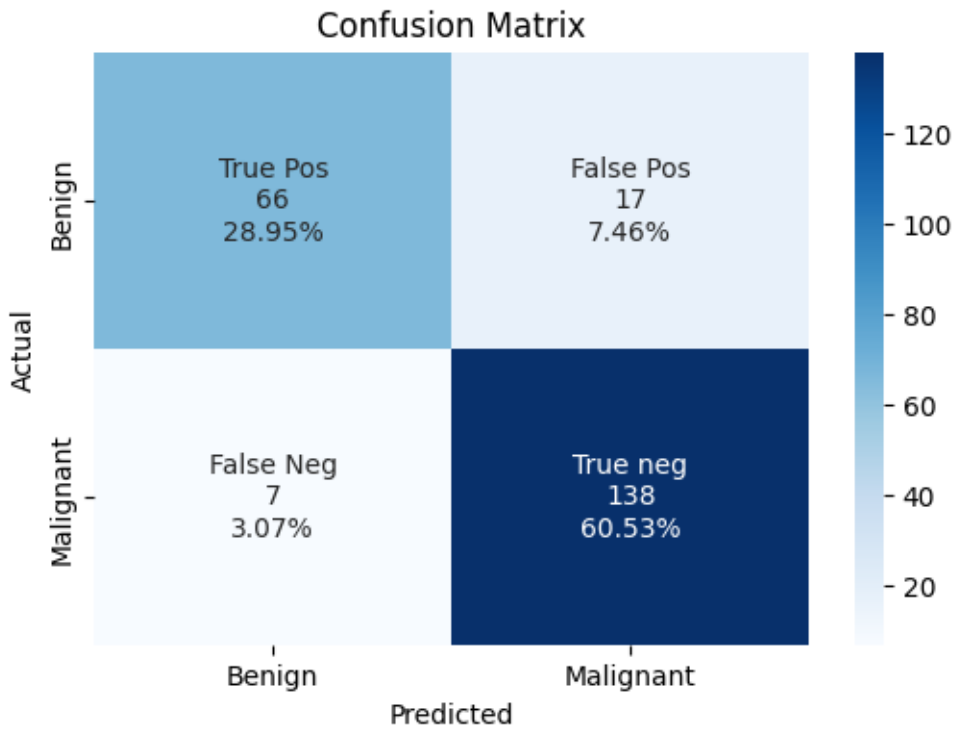

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.95      0.80      0.87        70
           1       0.88      0.97      0.92       101

    accuracy                           0.90       171
   macro avg       0.91      0.89      0.89       171
weighted avg       0.91      0.90      0.90       171
```

[14]:
```
## Train-Test split 0.4
FMultinomial(0.4)
FMultinomial(0.4, 2.1)
```

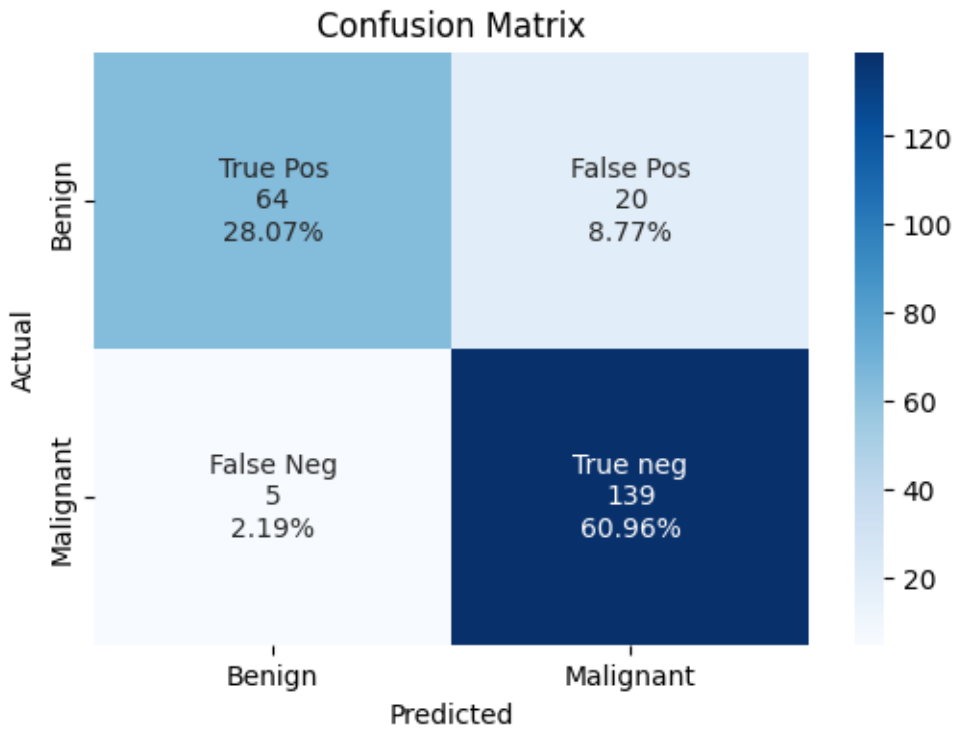
```
Train-test split: 0.4
value: alpha: 1.0
**********************************************
Confusion Matrix :
```



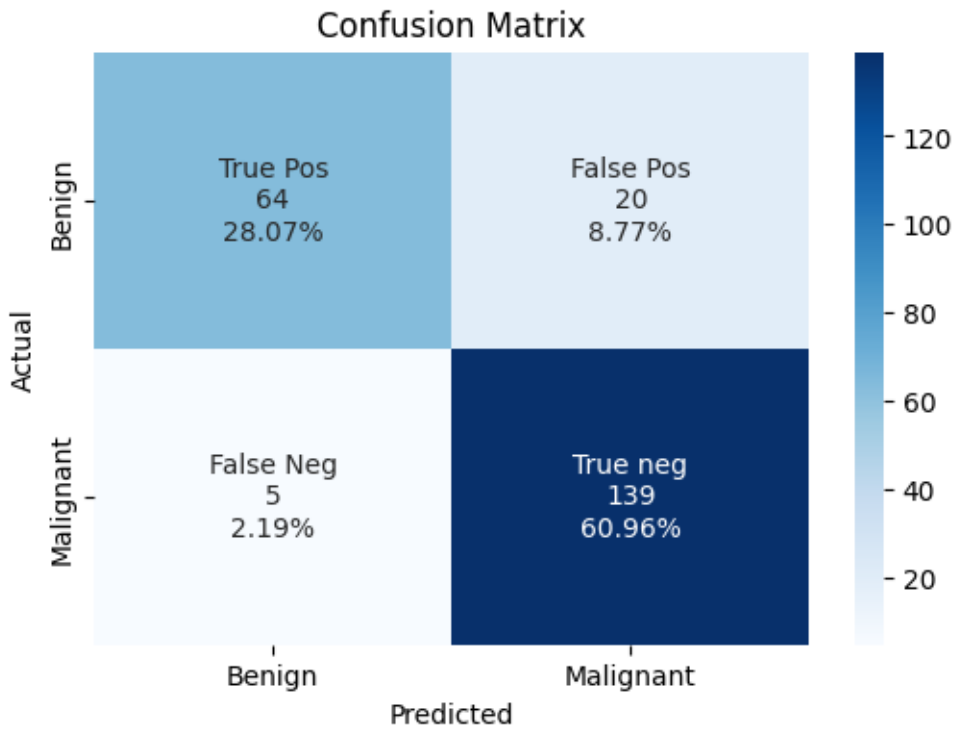

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.90      0.80      0.85        83
           1       0.89      0.95      0.92       145

    accuracy                           0.89       228
   macro avg       0.90      0.87      0.88       228
weighted avg       0.90      0.89      0.89       228

Confusion Matrix :
```




```
*********************************************
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.90      0.80      0.85        83
           1       0.89      0.95      0.92       145

    accuracy                           0.89       228
   macro avg       0.90      0.87      0.88       228
weighted avg       0.90      0.89      0.89       228

Train-test split: 0.4
value: alpha: 2.1
*********************************************
Confusion Matrix :
```



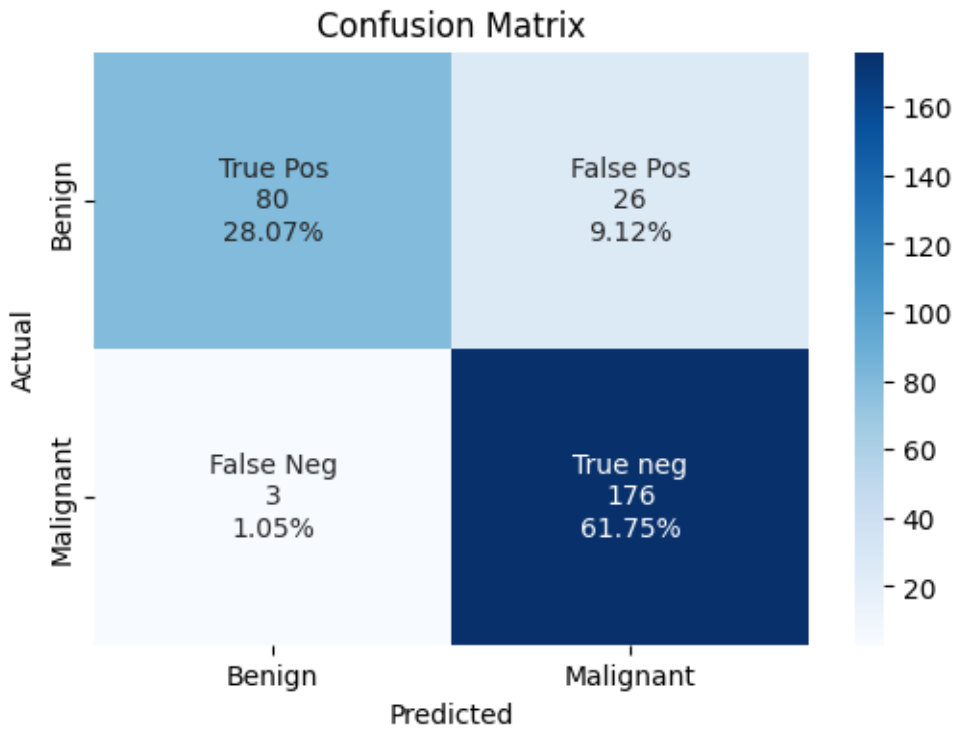

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.93      0.76      0.84        84
           1       0.87      0.97      0.92       144

    accuracy                           0.89       228
   macro avg       0.90      0.86      0.88       228
weighted avg       0.89      0.89      0.89       228

Confusion Matrix :
```



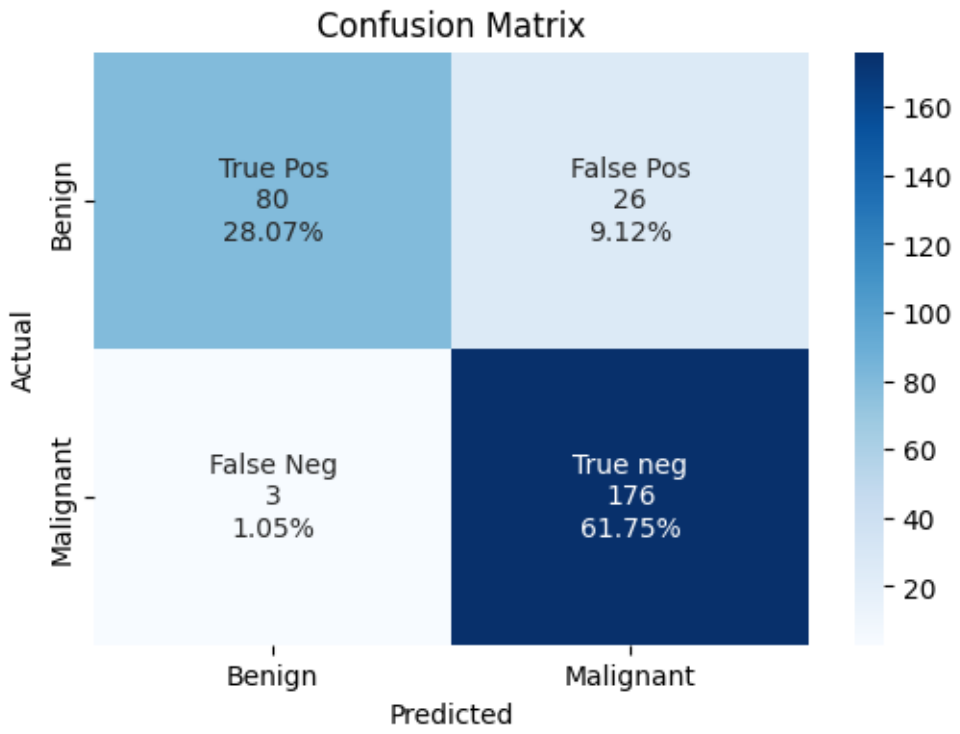

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.93      0.76      0.84        84
           1       0.87      0.97      0.92       144

    accuracy                           0.89       228
   macro avg       0.90      0.86      0.88       228
weighted avg       0.89      0.89      0.89       228
```

[15]:
```
## Train-Test split 0.5
FMultinomial(0.5)
FMultinomial(0.5, 1.8)
```

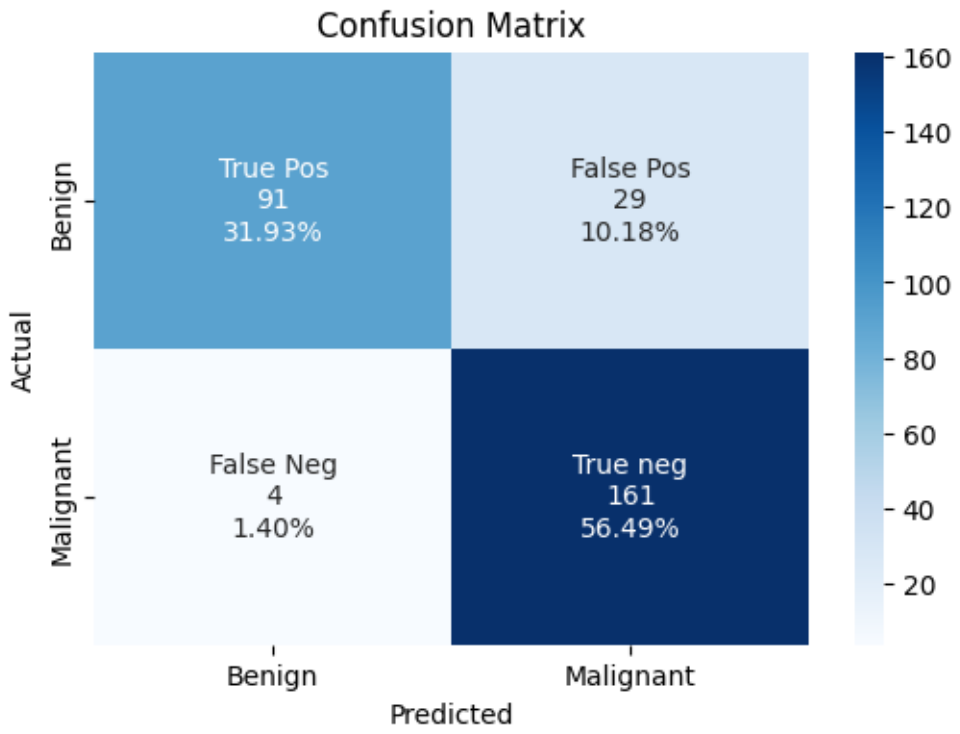
```
Train-test split: 0.5
value: alpha: 1.0
**********************************************
Confusion Matrix :
```



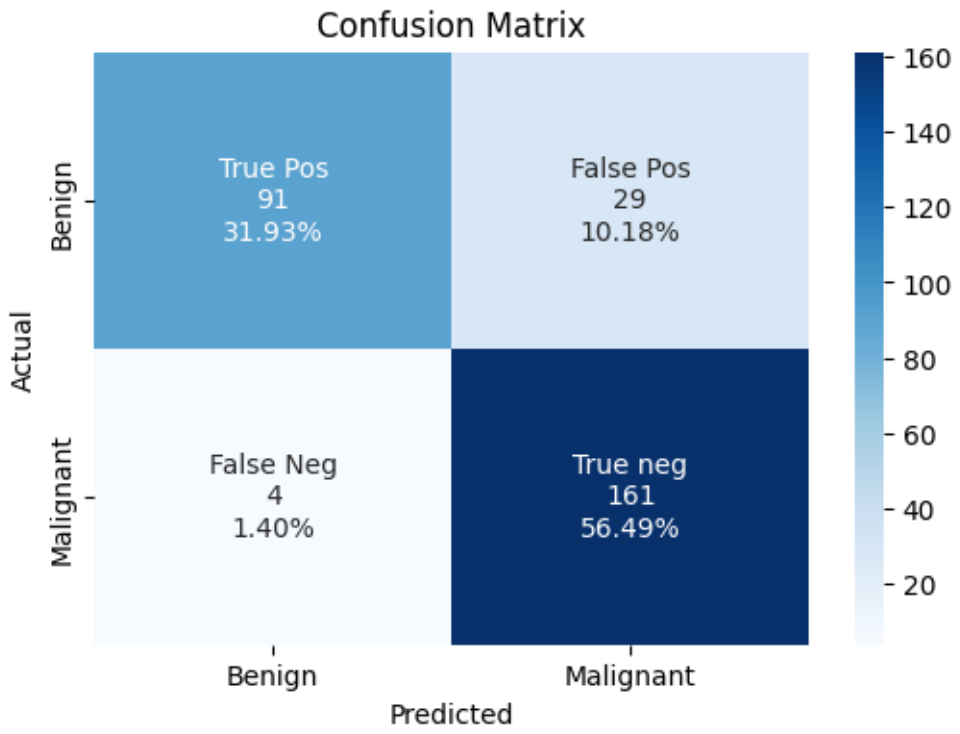

```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.96      0.75      0.85       106
           1       0.87      0.98      0.92       179

    accuracy                           0.90       285
   macro avg       0.92      0.87      0.89       285
weighted avg       0.91      0.90      0.90       285

Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.96      0.75      0.85       106
           1       0.87      0.98      0.92       179

    accuracy                           0.90       285
   macro avg       0.92      0.87      0.89       285
weighted avg       0.91      0.90      0.90       285

Train-test split: 0.5
value: alpha: 1.8
**********************************************
Confusion Matrix :
```

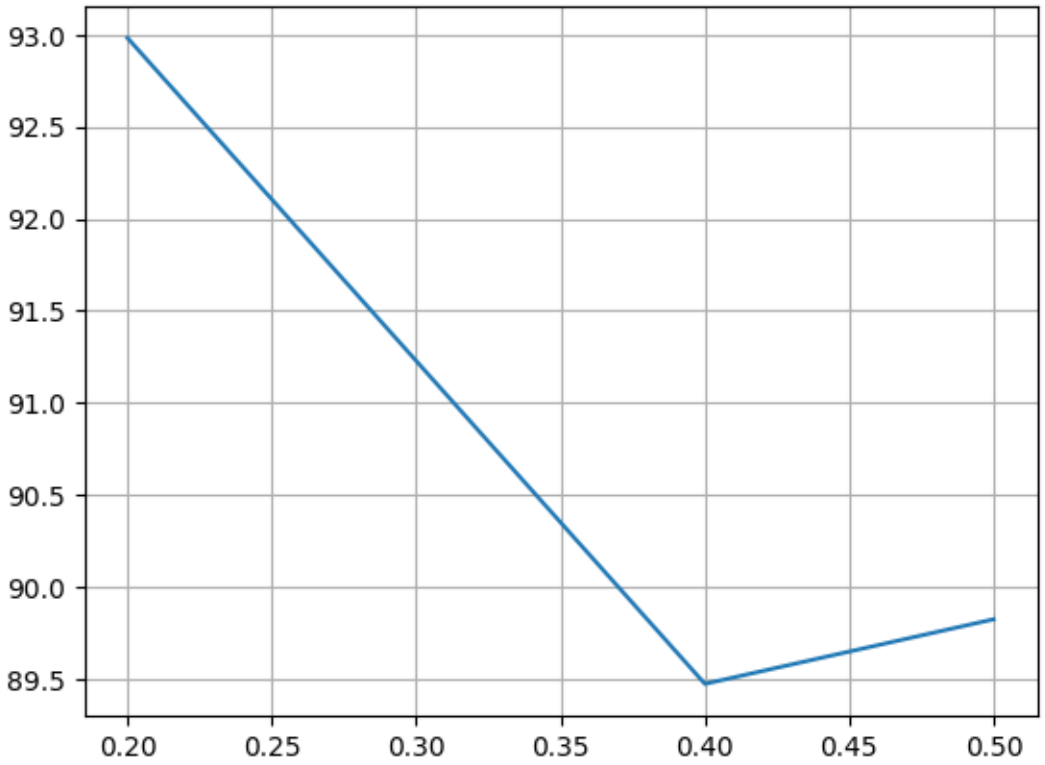



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.96      0.76      0.85       120
           1       0.85      0.98      0.91       165

    accuracy                           0.88       285
   macro avg       0.90      0.87      0.88       285
weighted avg       0.89      0.88      0.88       285

Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.96      0.76      0.85       120
           1       0.85      0.98      0.91       165

    accuracy                           0.88       285
   macro avg       0.90      0.87      0.88       285
weighted avg       0.89      0.88      0.88       285
```

[16]:
```
x_points = [float(key) for key in dict_mnb]
y_points = [i*100 for i in dict_mnb.values()]
plt.plot(x_points, y_points)
plt.grid(True)
plt.show()
```

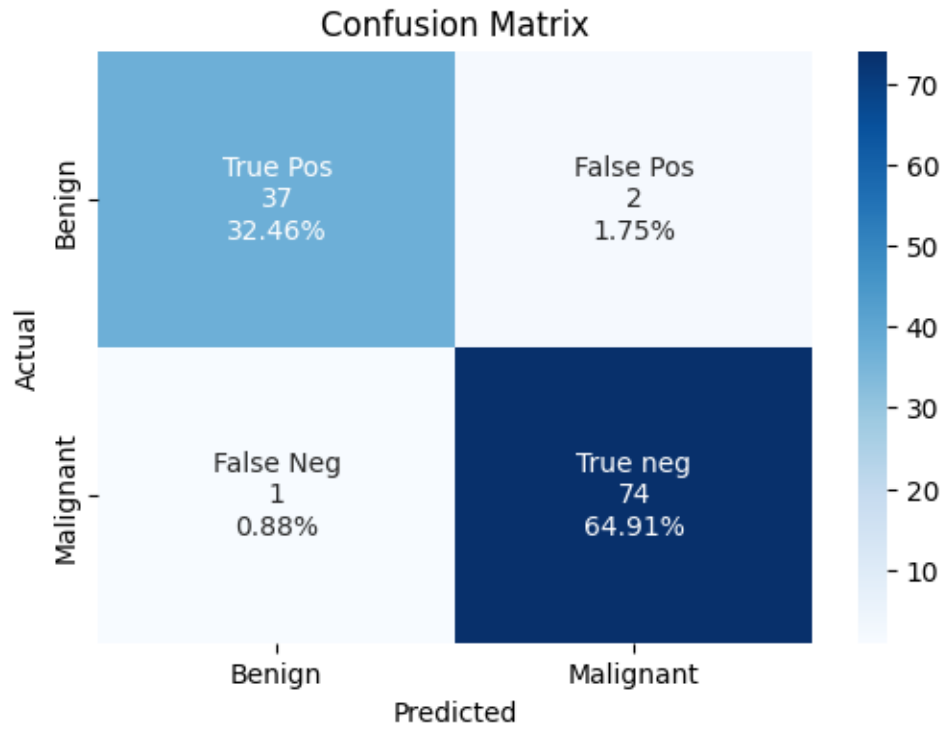


0.1.1 Classification using Guassian Naive Bayes

[17]:
```
def FGaussian(split):
  from sklearn.naive_bayes import GaussianNB
  from sklearn.model_selection import train_test_split
  from sklearn.metrics import accuracy_score
  from sklearn.preprocessing import StandardScaler
  scaler = StandardScaler()
  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = split,␣
 ↪random_state=44)
  scaler.fit_transform(X_train)
  scaler.transform(X_test)
  classifier = GaussianNB()
  classifier.fit(X_train, y_train)
  y_pred = classifier.predict(X_test)
  print("Train-test split: " + str(split))
  print("**********************************************")
  reports(y_test, y_pred)
  accuracy = accuracy_score(y_test, y_pred)
  if(str(split) in dict_gnb):
    dict_gnb[str(split)] = max(accuracy, dict_gnb[str(split)])
    if(str(split) == '0.3' and accuracy > dict_bnb[str(split)]):
```

```
      RocAucgnb['max'] = {'y_test': y_test, 'y_pred': y_pred}
  else:
    dict_gnb[str(split)] = accuracy
    RocAucgnb['max'] = {'y_test': y_test, 'y_pred': y_pred}

## Train-Test split 0.2
FGaussian(0.2)
# 94, 97, 94, 96,
```

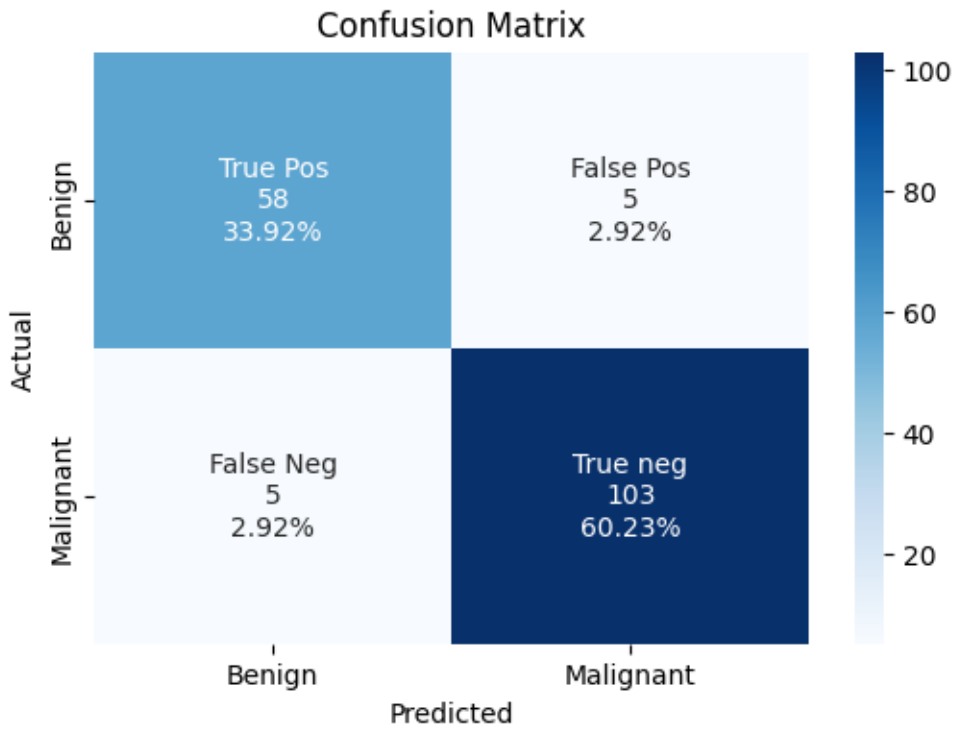
```
Train-test split: 0.2
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.97      0.95      0.96        39
           1       0.97      0.99      0.98        75

    accuracy                           0.97       114
   macro avg       0.97      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114
```

[18]:
```
## Train-Test split 0.3
FGaussian(0.3)
```

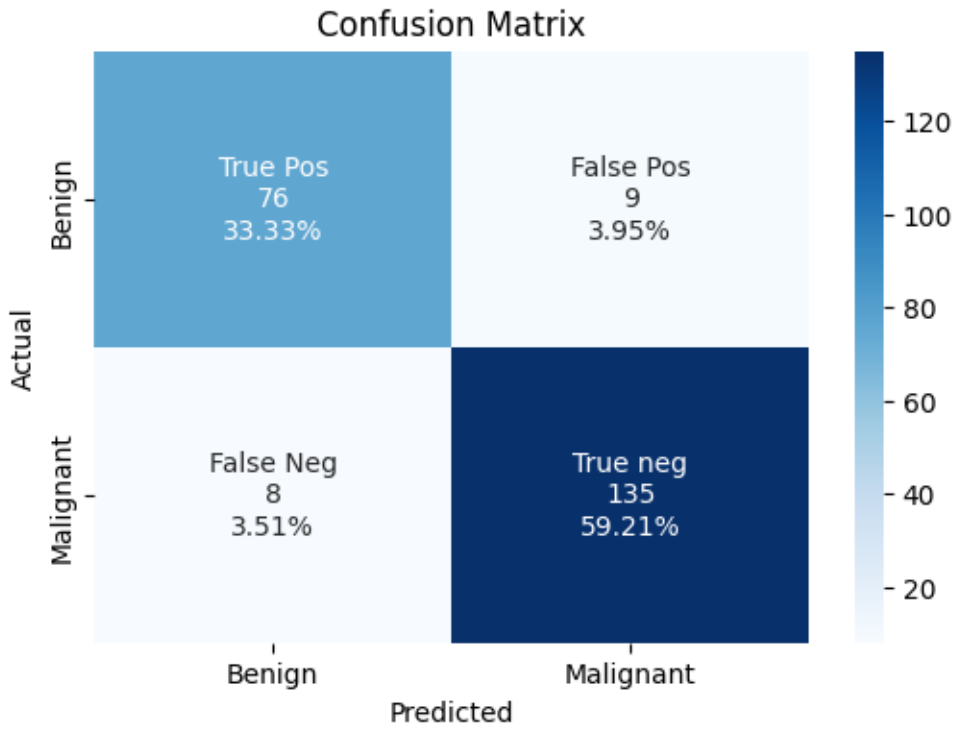
```
Train-test split: 0.3
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.92      0.92      0.92        63
           1       0.95      0.95      0.95       108

    accuracy                           0.94       171
   macro avg       0.94      0.94      0.94       171
weighted avg       0.94      0.94      0.94       171
```

[19]:
```
## Train-Test split 0.4
FGaussian(0.4)
```

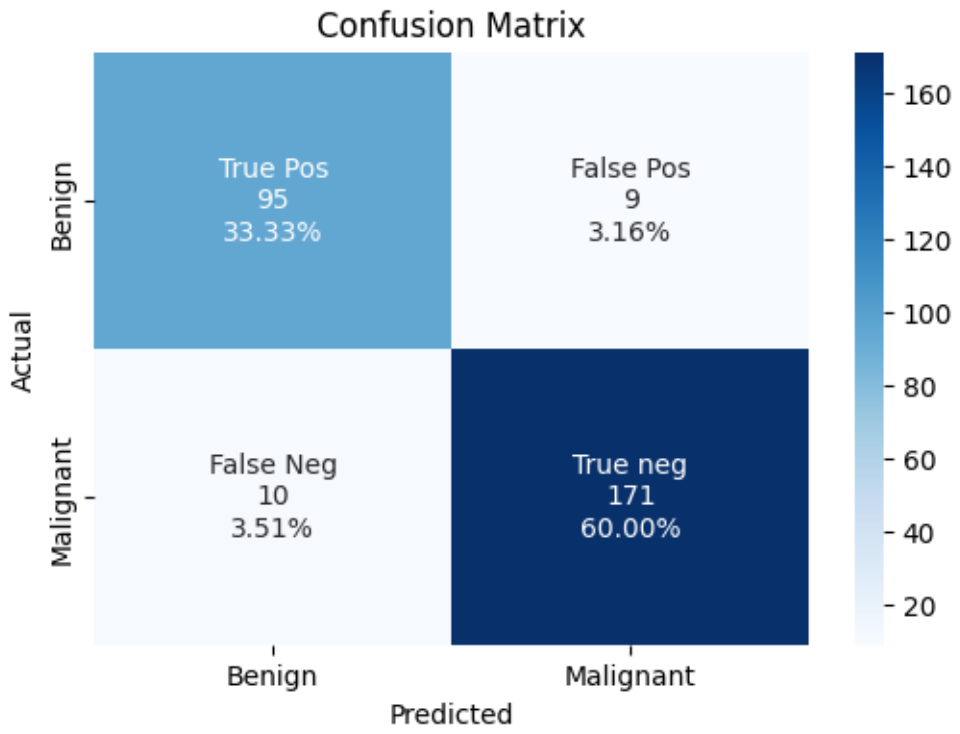
```
Train-test split: 0.4
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.90      0.89      0.90        85
           1       0.94      0.94      0.94       143

    accuracy                           0.93       228
   macro avg       0.92      0.92      0.92       228
weighted avg       0.93      0.93      0.93       228
```

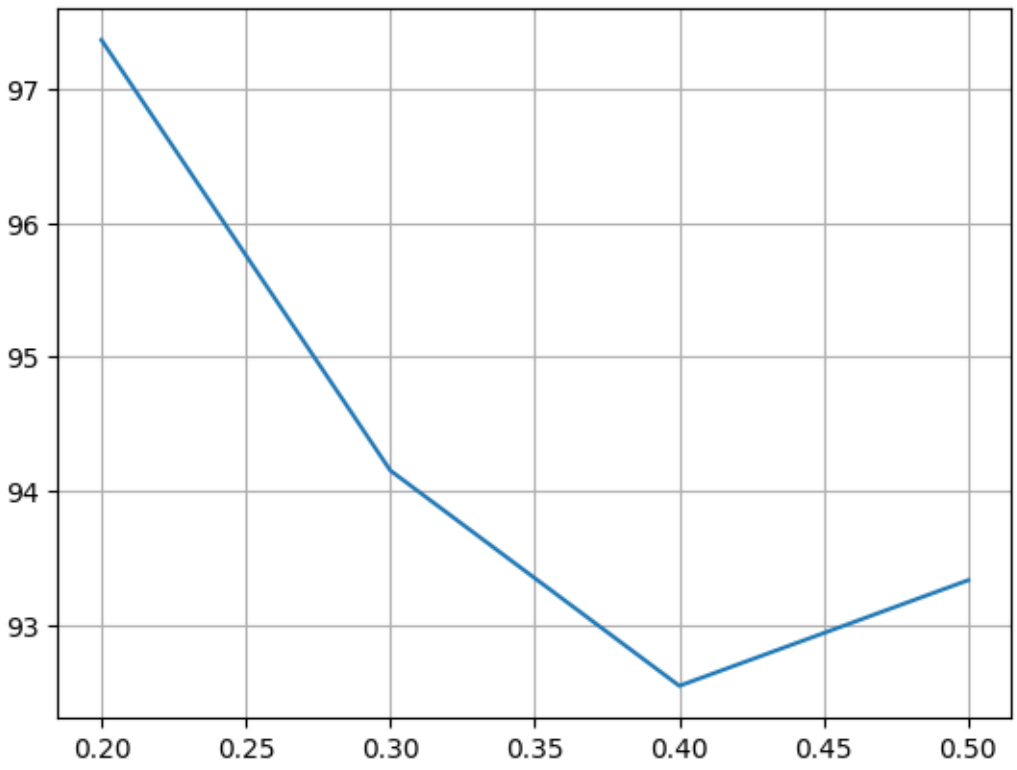
[20]:
```
## Train-Test split 0.5
FGaussian(0.5)
```

```
Train-test split: 0.5
**********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.90      0.91      0.91       104
           1       0.95      0.94      0.95       181

    accuracy                           0.93       285
   macro avg       0.93      0.93      0.93       285
weighted avg       0.93      0.93      0.93       285
```

[21]:
```
x_points = [float(key) for key in dict_gnb]
y_points = [i*100 for i in dict_gnb.values()]
plt.plot(x_points, y_points)
plt.grid(True)
plt.show()
```



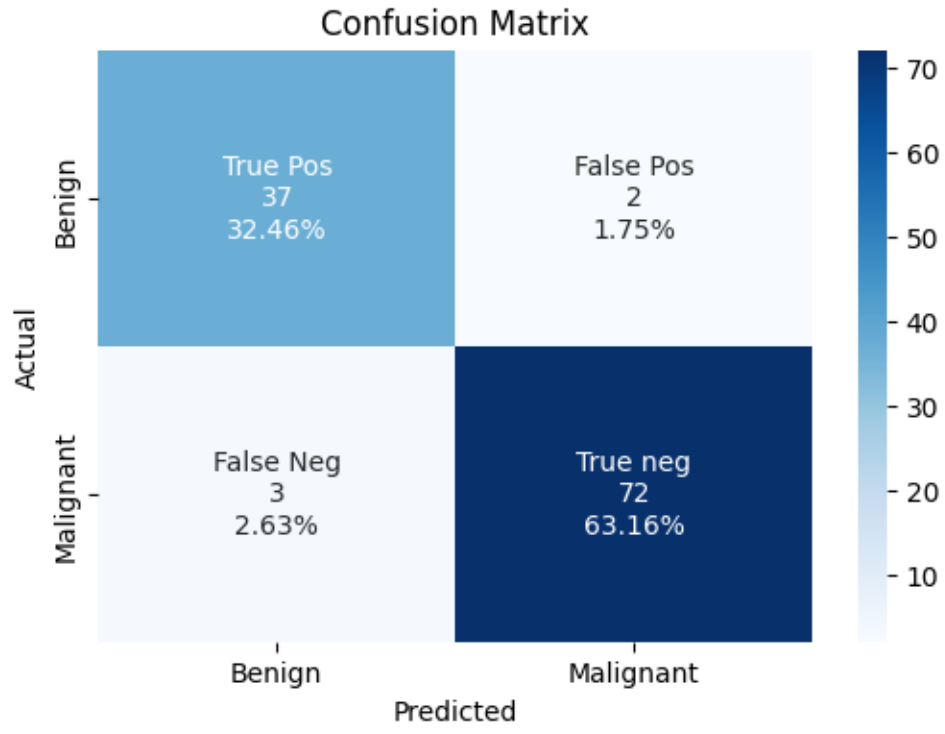
0.1.2 Classification using Decision Tree

[22]:
```
def decision_tree(split, criterion_value):
  from sklearn.model_selection import train_test_split
  from sklearn.tree import DecisionTreeClassifier
  from sklearn import tree
  from sklearn.metrics import accuracy_score
  from sklearn.preprocessing import StandardScaler
  scaler = StandardScaler()
  X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = split,␣
 ↪random_state=44)
  scaler.fit_transform(X_train)
  scaler.transform(X_test)

  classifier = DecisionTreeClassifier(criterion = criterion_value)
  classifier.fit(X_train, y_train)
  y_pred = classifier.predict(X_test)
  print("Train-test split: " + str(split))
  print("Value: Entropy: " + criterion_value)
  print("**********************************************")
  reports(y_test, y_pred)
  accuracy = accuracy_score(y_test, y_pred)
```

```
if(str(split) in dict_dtr):
  dict_dtr[str(split)] = max(accuracy, dict_dtr[str(split)])
  if(str(split) == '0.3' and accuracy > dict_bnb[str(split)]):
    RocAucdtr['max'] = {'y_test': y_test, 'y_pred': y_pred}
else:
  dict_dtr[str(split)] = accuracy
  if(str(split) == '0.3'):
    RocAucdtr['max'] = {'y_test': y_test, 'y_pred': y_pred}

fig = plt.figure(figsize=(12,8))
_ = tree.plot_tree(classifier,
                   feature_names=column_names,
                   class_names=['outcome1', 'outcome2'],
                   filled=True)
```

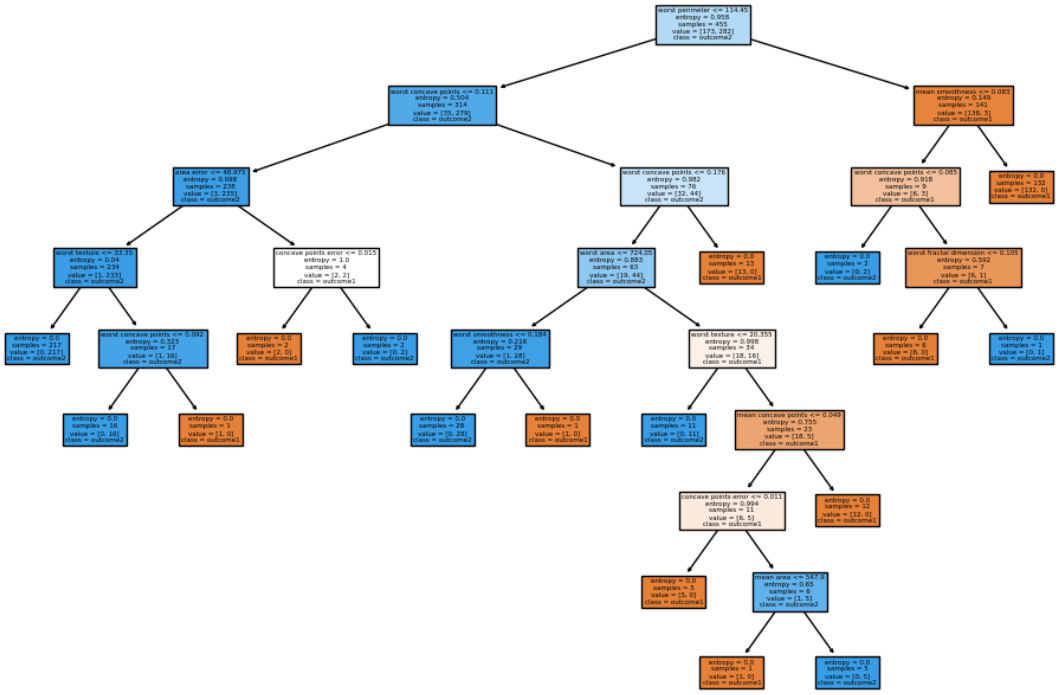
[23]:
```
decision_tree(0.2, 'entropy')
```

```
Train-test split: 0.2
Value: Entropy: entropy
*********************************************
Confusion Matrix :
```



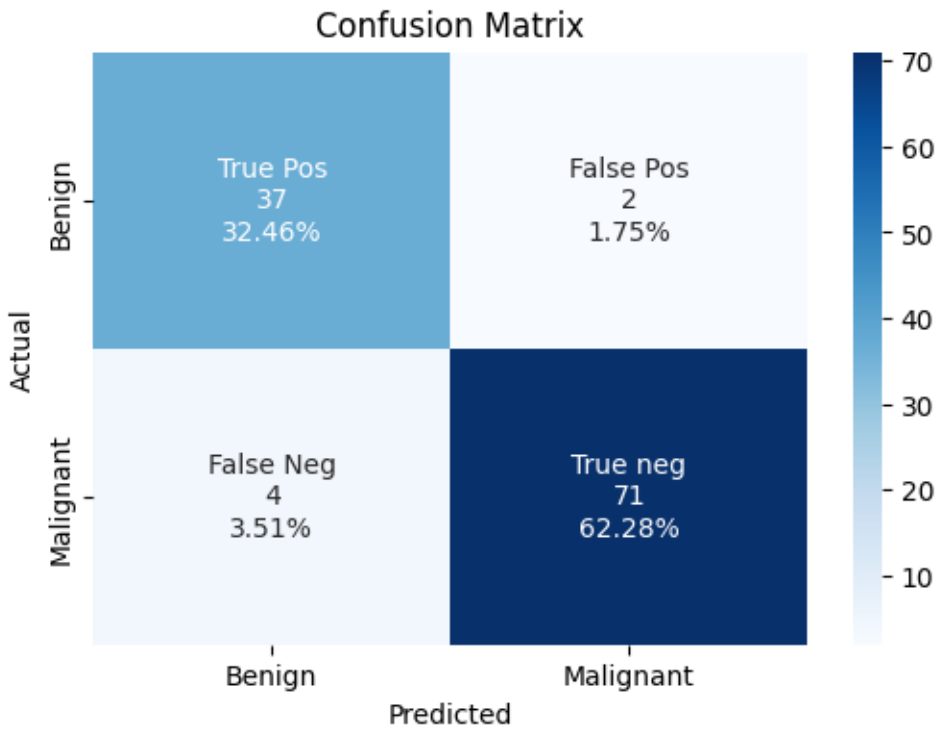

```
*********************************************
```

```
*********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.93      0.95      0.94        39
           1       0.97      0.96      0.97        75

    accuracy                           0.96       114
   macro avg       0.95      0.95      0.95       114
weighted avg       0.96      0.96      0.96       114
```

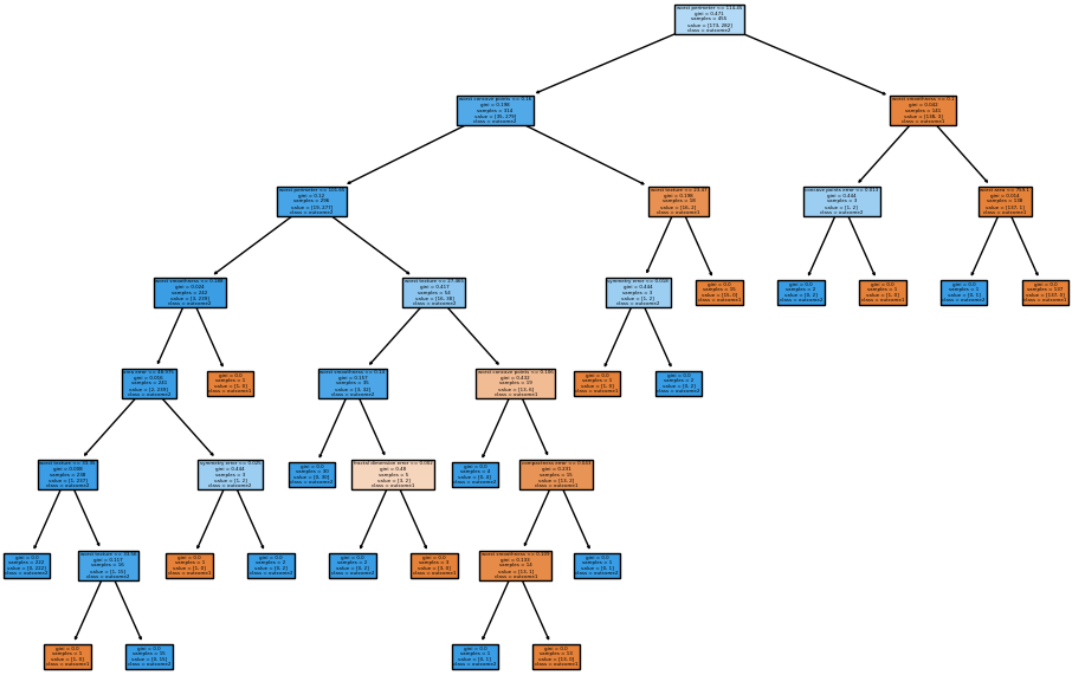


[24]:
```
decision_tree(0.2, 'gini')
```

```
Train-test split: 0.2
Value: Entropy: gini
*********************************************
Confusion Matrix :
```

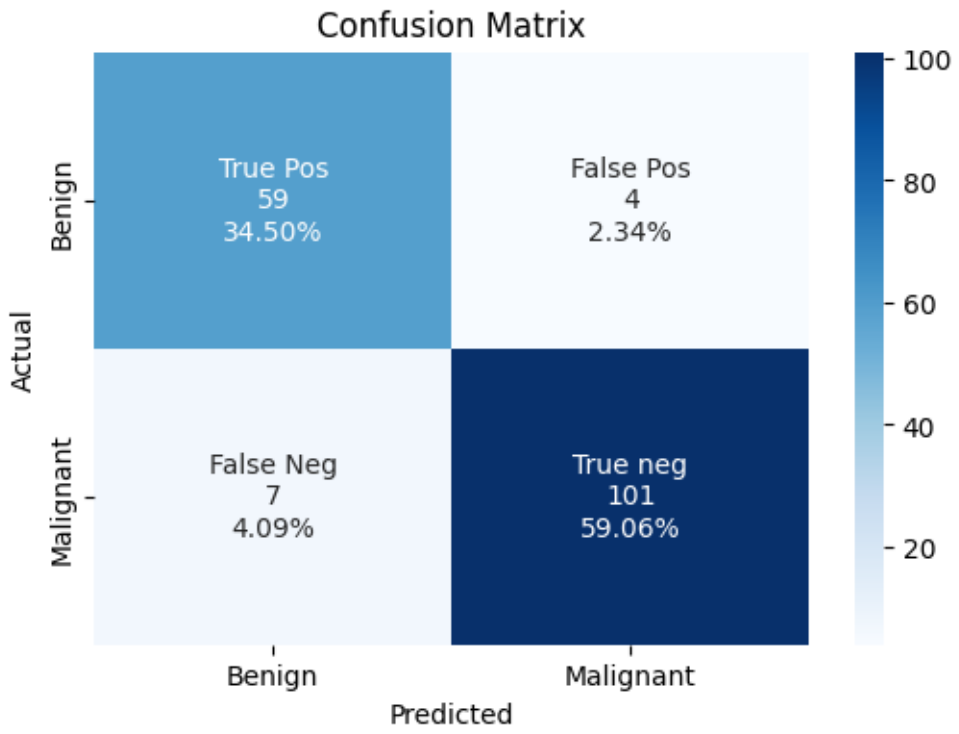



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.90      0.95      0.92        39
           1       0.97      0.95      0.96        75

    accuracy                           0.95       114
   macro avg       0.94      0.95      0.94       114
weighted avg       0.95      0.95      0.95       114
```

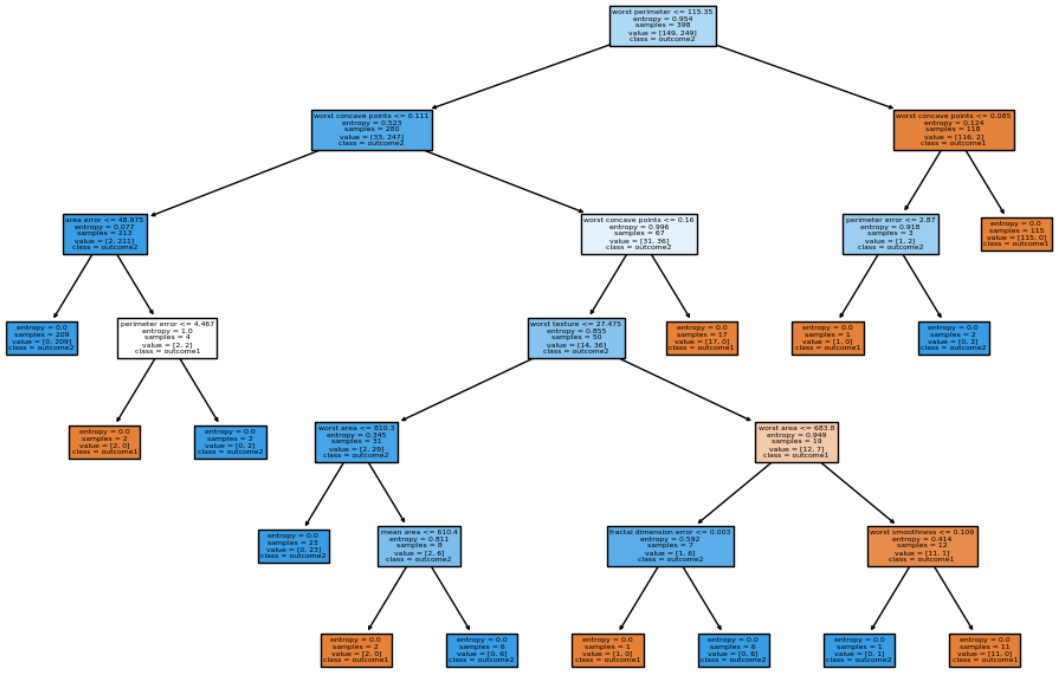


[25]:
```
decision_tree(0.3, 'entropy')
```

```
Train-test split: 0.3
Value: Entropy: entropy
*********************************************
Confusion Matrix :
```

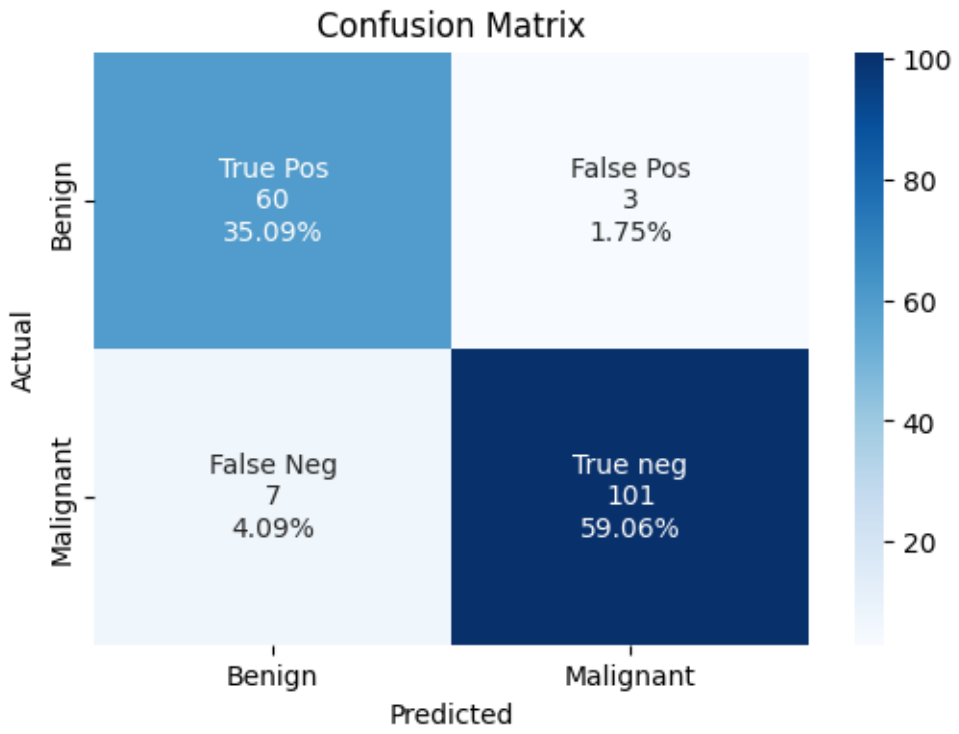



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.89      0.94      0.91        63
           1       0.96      0.94      0.95       108

    accuracy                           0.94       171
   macro avg       0.93      0.94      0.93       171
weighted avg       0.94      0.94      0.94       171
```

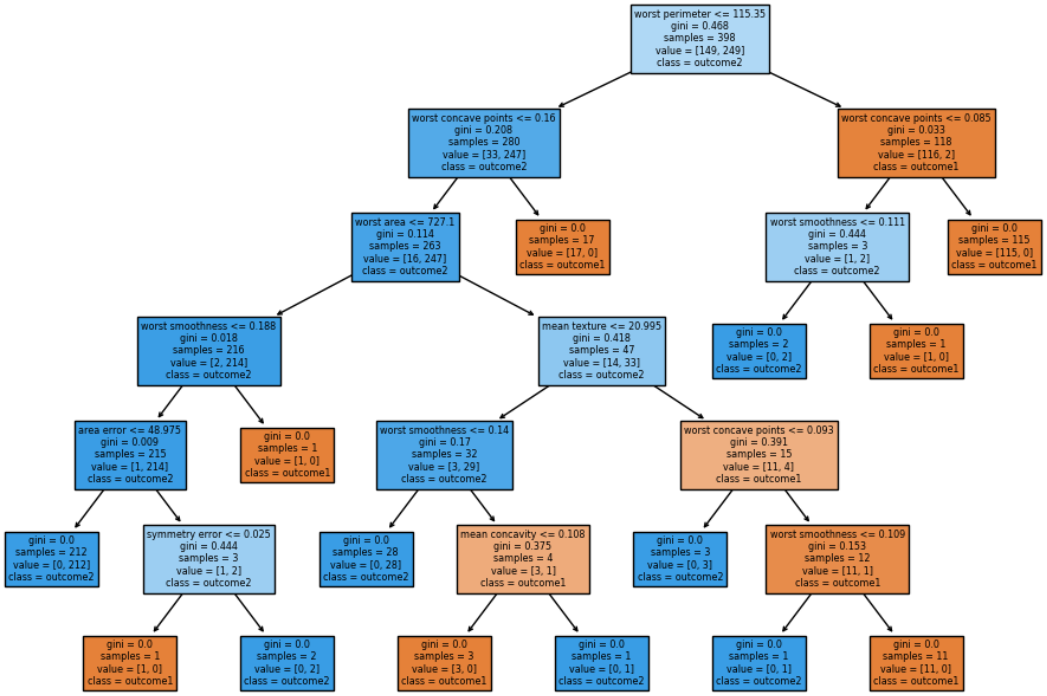


[26]:
```
decision_tree(0.3, 'gini')
```

```
Train-test split: 0.3
Value: Entropy: gini
*********************************************
Confusion Matrix :
```

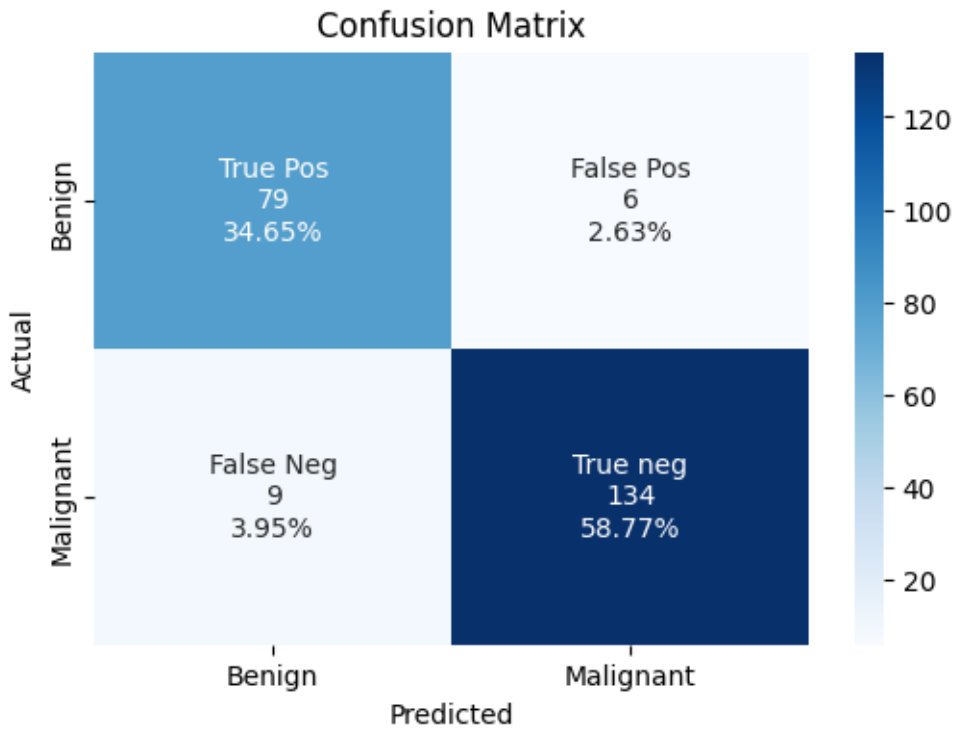



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.90      0.95      0.92        63
           1       0.97      0.94      0.95       108

    accuracy                           0.94       171
   macro avg       0.93      0.94      0.94       171
weighted avg       0.94      0.94      0.94       171
```

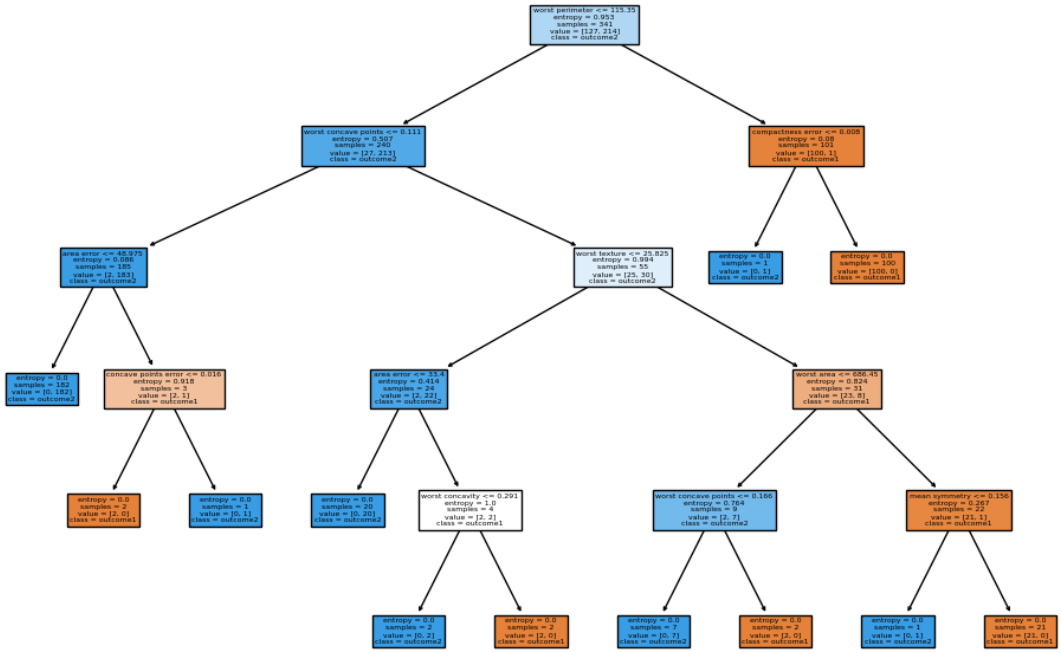


[27]:
```
decision_tree(0.4, 'entropy')
```

```
Train-test split: 0.4
Value: Entropy: entropy
*********************************************
Confusion Matrix :
```

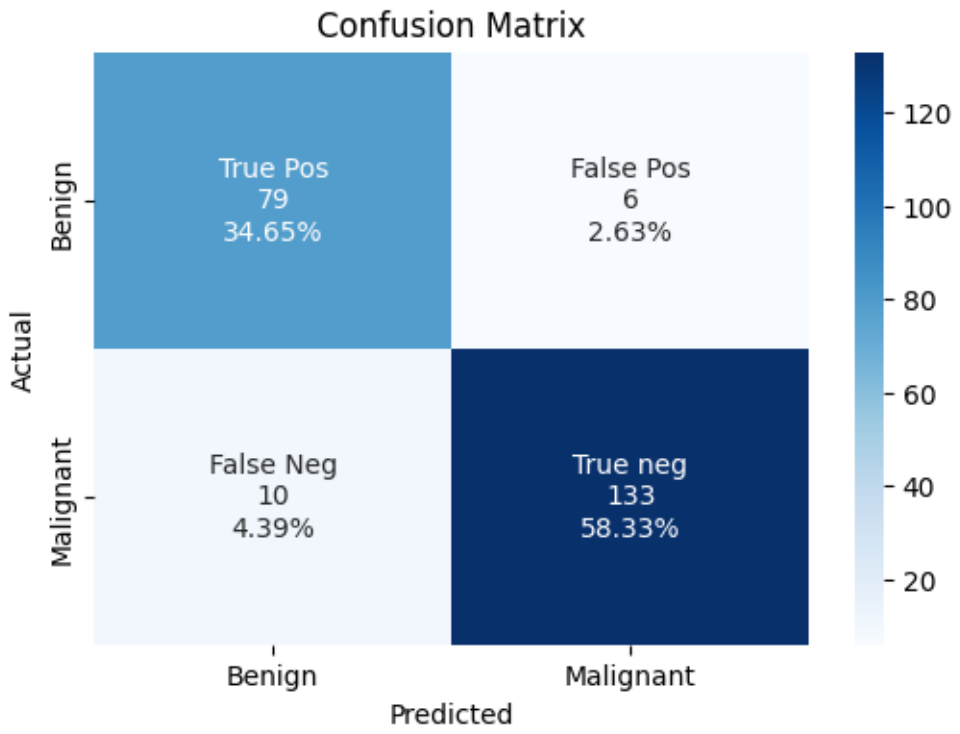



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.90      0.93      0.91        85
           1       0.96      0.94      0.95       143

    accuracy                           0.93       228
   macro avg       0.93      0.93      0.93       228
weighted avg       0.93      0.93      0.93       228
```

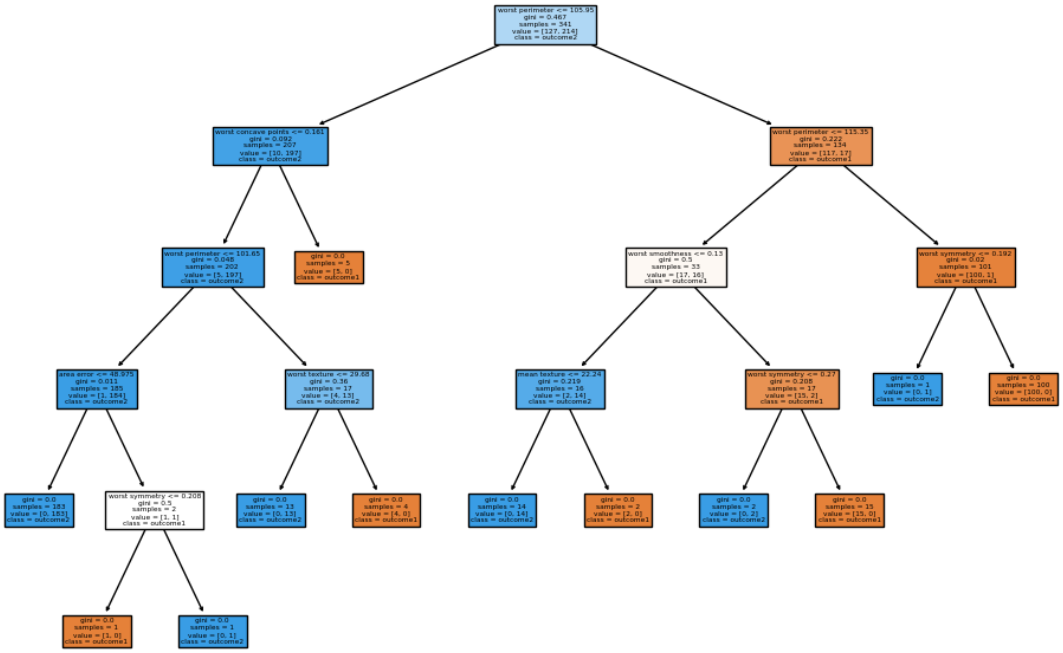


[28]:
```
decision_tree(0.4, 'gini')
```

```
Train-test split: 0.4
Value: Entropy: gini
*********************************************
Confusion Matrix :
```

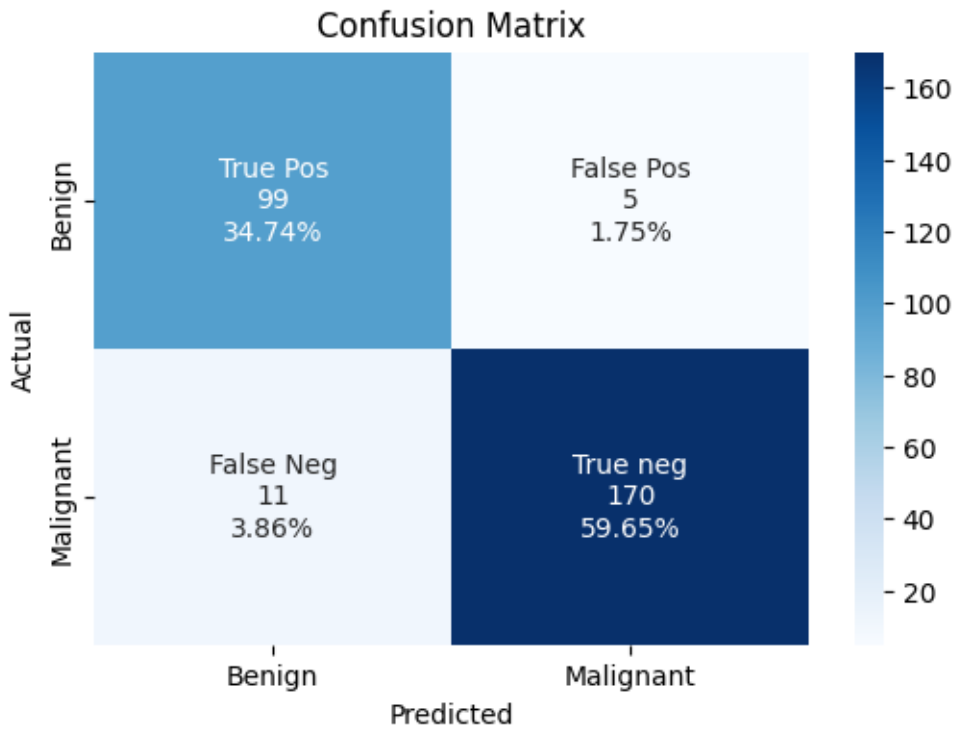



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.89      0.93      0.91        85
           1       0.96      0.93      0.94       143

    accuracy                           0.93       228
   macro avg       0.92      0.93      0.93       228
weighted avg       0.93      0.93      0.93       228
```

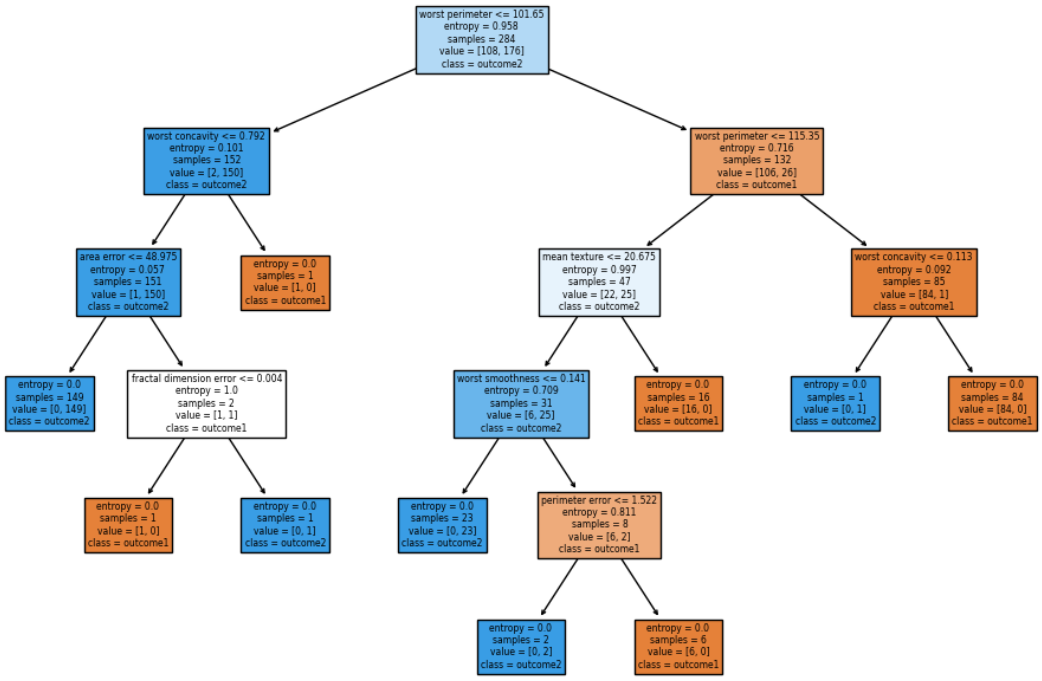


[29]:
```
decision_tree(0.5, 'entropy')
```

```
Train-test split: 0.5
Value: Entropy: entropy
*********************************************
Confusion Matrix :
```

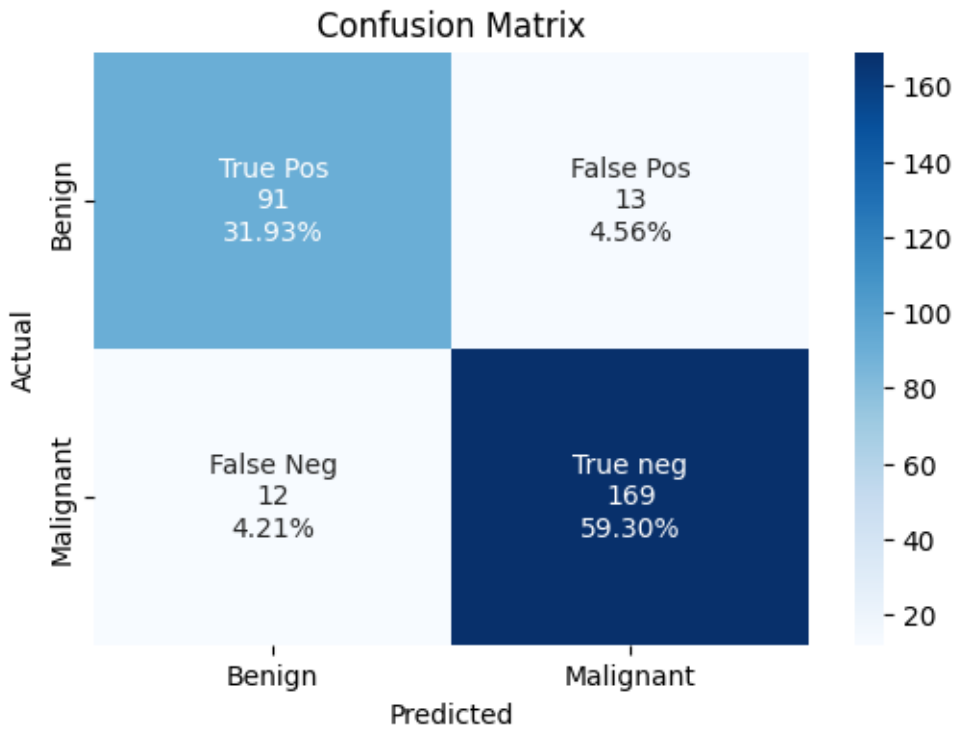



```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.90      0.95      0.93       104
           1       0.97      0.94      0.96       181

    accuracy                           0.94       285
   macro avg       0.94      0.95      0.94       285
weighted avg       0.95      0.94      0.94       285
```

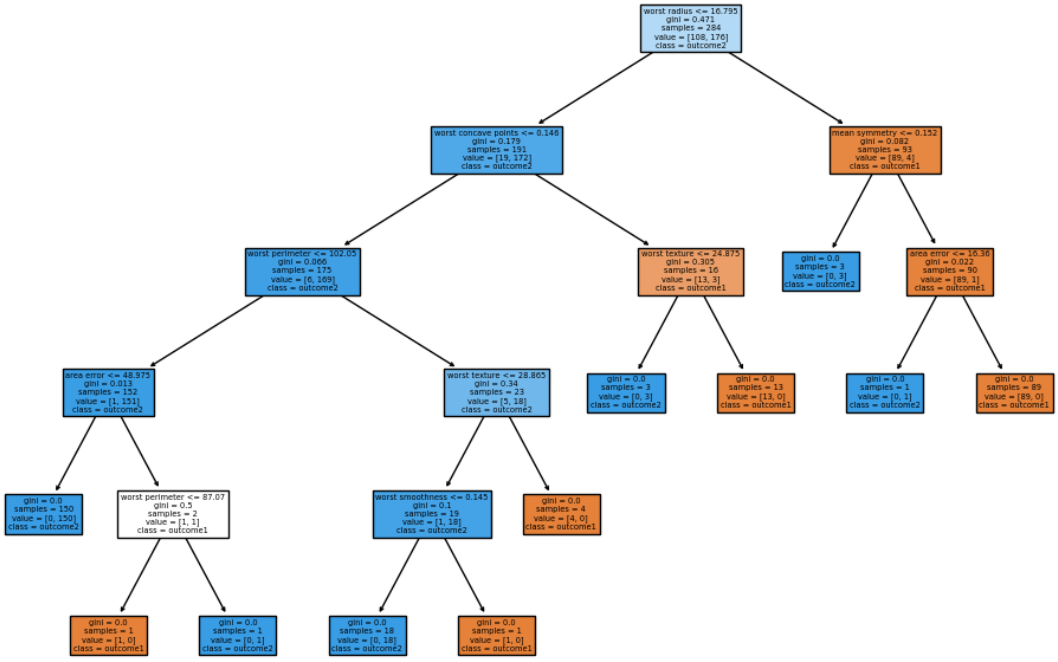


[30]:
```
decision_tree(0.5, 'gini')
```

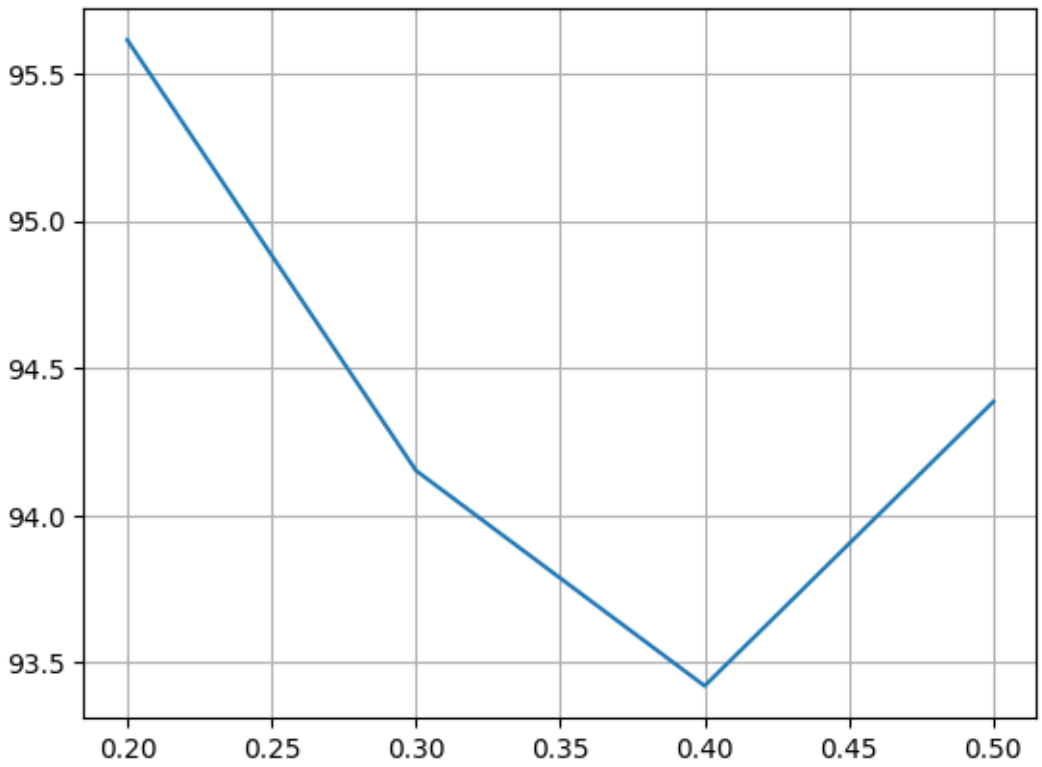
```
Train-test split: 0.5
Value: Entropy: gini
*********************************************
Confusion Matrix :
```




```
**********************************************
**********************************************
Classification Evaluation :
              precision    recall  f1-score   support

           0       0.88      0.88      0.88       104
           1       0.93      0.93      0.93       181

    accuracy                           0.91       285
   macro avg       0.91      0.90      0.91       285
weighted avg       0.91      0.91      0.91       285
```



[31]:
```
x_points = [float(key) for key in dict_dtr]
y_points = [i*100 for i in dict_dtr.values()]
plt.plot(x_points, y_points)
plt.grid(True)
plt.show()
```



[32]:
```
from sklearn import metrics
def auc_roc():
    fpr1, tpr1, _1 = metrics.roc_curve(RocAucbnb['max']['y_test'],␣
 ↪RocAucbnb['max']['y_pred'], pos_label=1)
    fpr4, tpr4, _3 = metrics.roc_curve(RocAucmnb['max']['y_test'],␣
 ↪RocAucmnb['max']['y_pred'], pos_label=1)
    fpr2, tpr2, _2 = metrics.roc_curve(RocAucgnb['max']['y_test'],␣
 ↪RocAucgnb['max']['y_pred'], pos_label=1)
    fpr3, tpr3, _3 = metrics.roc_curve(RocAucdtr['max']['y_test'],␣
 ↪RocAucdtr['max']['y_pred'], pos_label=1)
    plt.plot(fpr1, tpr1, linestyle='--',color='orange', label='BernoulliNB')
    plt.plot(fpr4, tpr4, linestyle='--',color='green', label='MultinomialNB')
    plt.plot(fpr2, tpr2, linestyle='--', color='blue', label= 'GaussianNB')
    plt.plot(fpr3, tpr3, linestyle='--', color='black', label= 'Decision Tree')
    plt.title('ROC curve')
    # x label
    plt.xlabel('False Positive Rate')
    # y label
    plt.ylabel('True Positive rate')

    plt.legend(loc='best')
    plt.savefig('ROC',dpi=300)
```

```
    plt.show()
auc_roc()
```