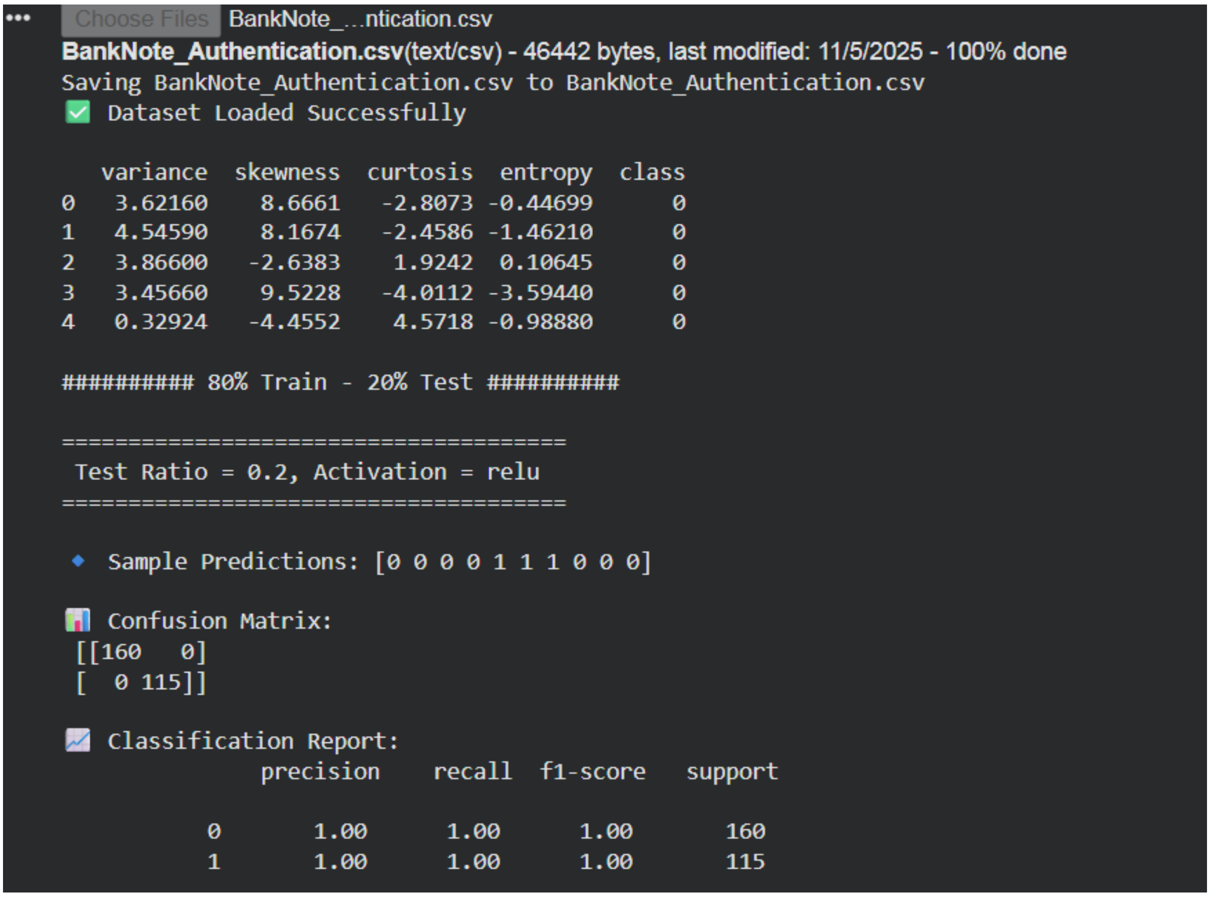
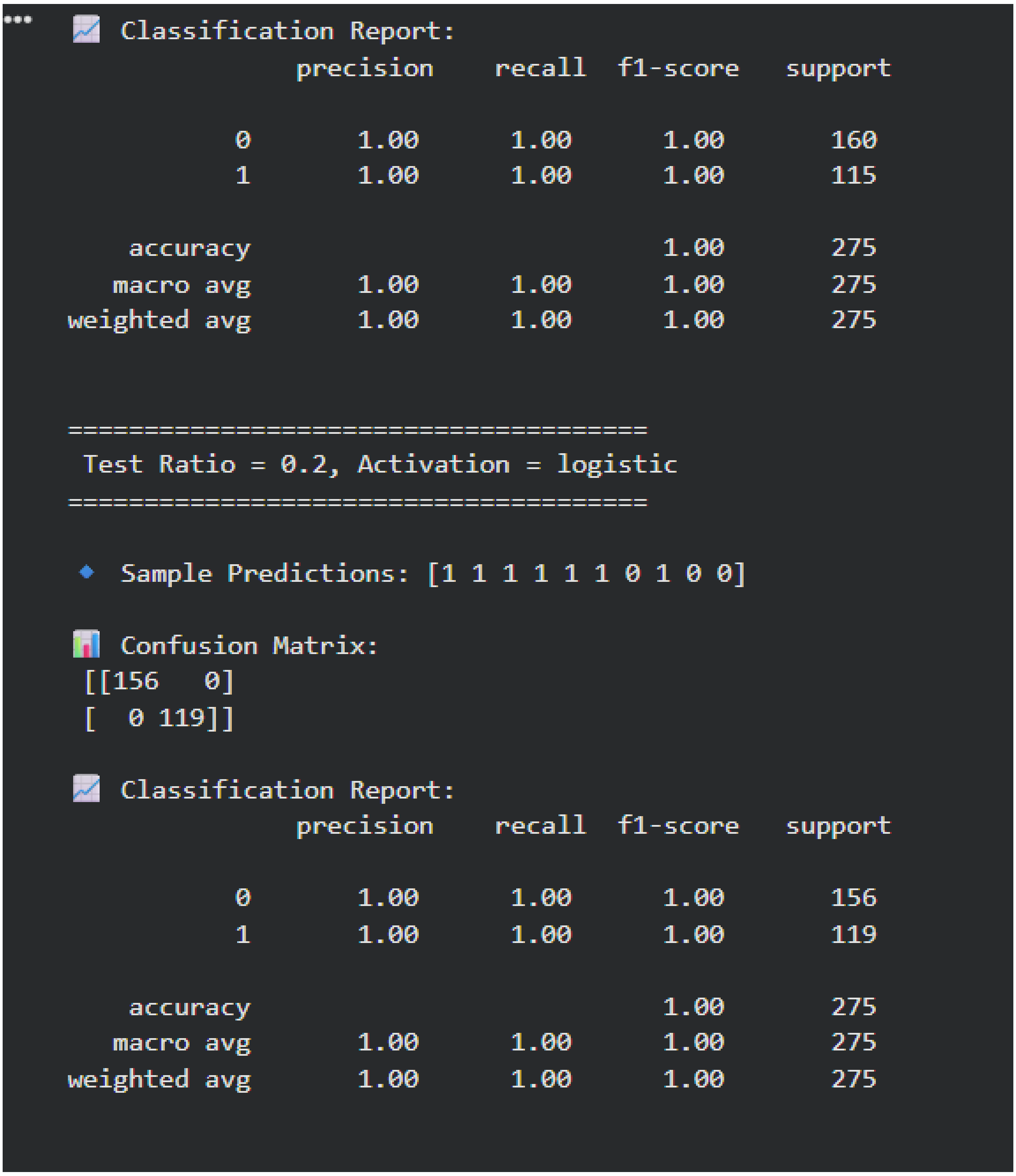
```
# ================== FULL CODE FOR COLAB ==================


# 📌 Upload Dataset
from google.colab import files
uploaded = files.upload()


# 📌 Import Libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report, confusion_matrix


# 📌 Load Dataset
file_name = list(uploaded.keys())[0]
bnotes = pd.read_csv(file_name)


print("✅ Dataset Loaded Successfully\n")
print(bnotes.head())


# 📌 Split data
x = bnotes.drop('class', axis=1)
y = bnotes['class']


# 📌 Function to train & print results
def train_model(test_ratio, activation):
    print("\n========================================")
    print(f" Test Ratio = {test_ratio}, Activation = {activation}")
    print("========================================")
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=test_ratio)

    mlp = MLPClassifier(max_iter=500, activation=activation)
    mlp.fit(x_train, y_train)

    pred = mlp.predict(x_test)

    print("\n 🔹 Sample Predictions:", pred[:10])
    print("\n 📊 Confusion Matrix:\n", confusion_matrix(y_test, pred))
    print("\n 📈 Classification Report:\n", classification_report(y_test, pred))

# 📌 Activation functions to test
activations = ['relu', 'logistic', 'tanh', 'identity']

# 🔥 Run for test size 0.2 (80-20 split)
print("\n########## 80% Train - 20% Test ##########")
for act in activations:
    train_model(0.2, act)

# 🔥 Run for test size 0.3 (70-30 split)
print("\n########## 70% Train - 30% Test ##########")
for act in activations:
    train_model(0.3, act)
```

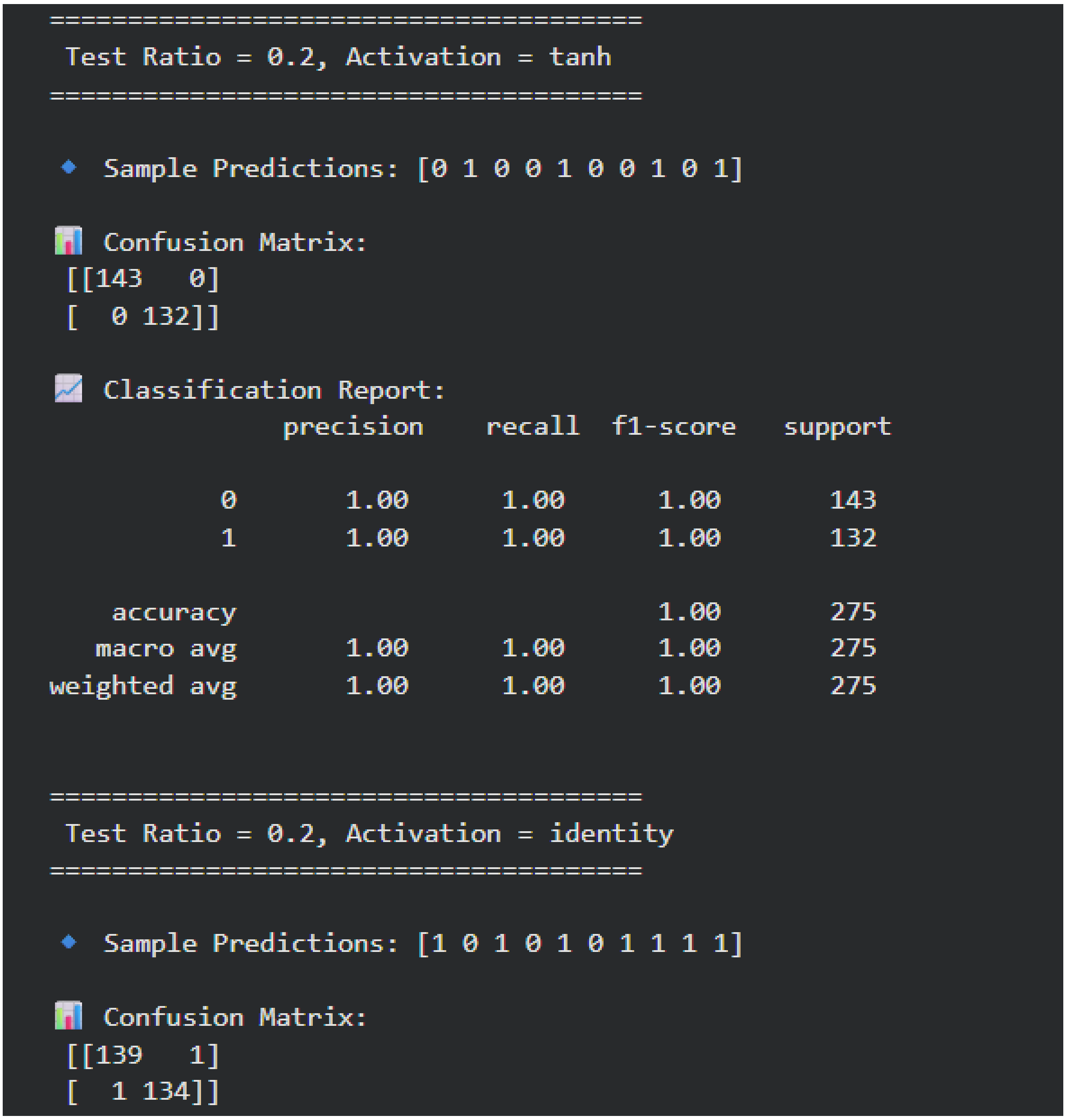
```
print("\n✅ All Experiments Completed Successfully!")
```



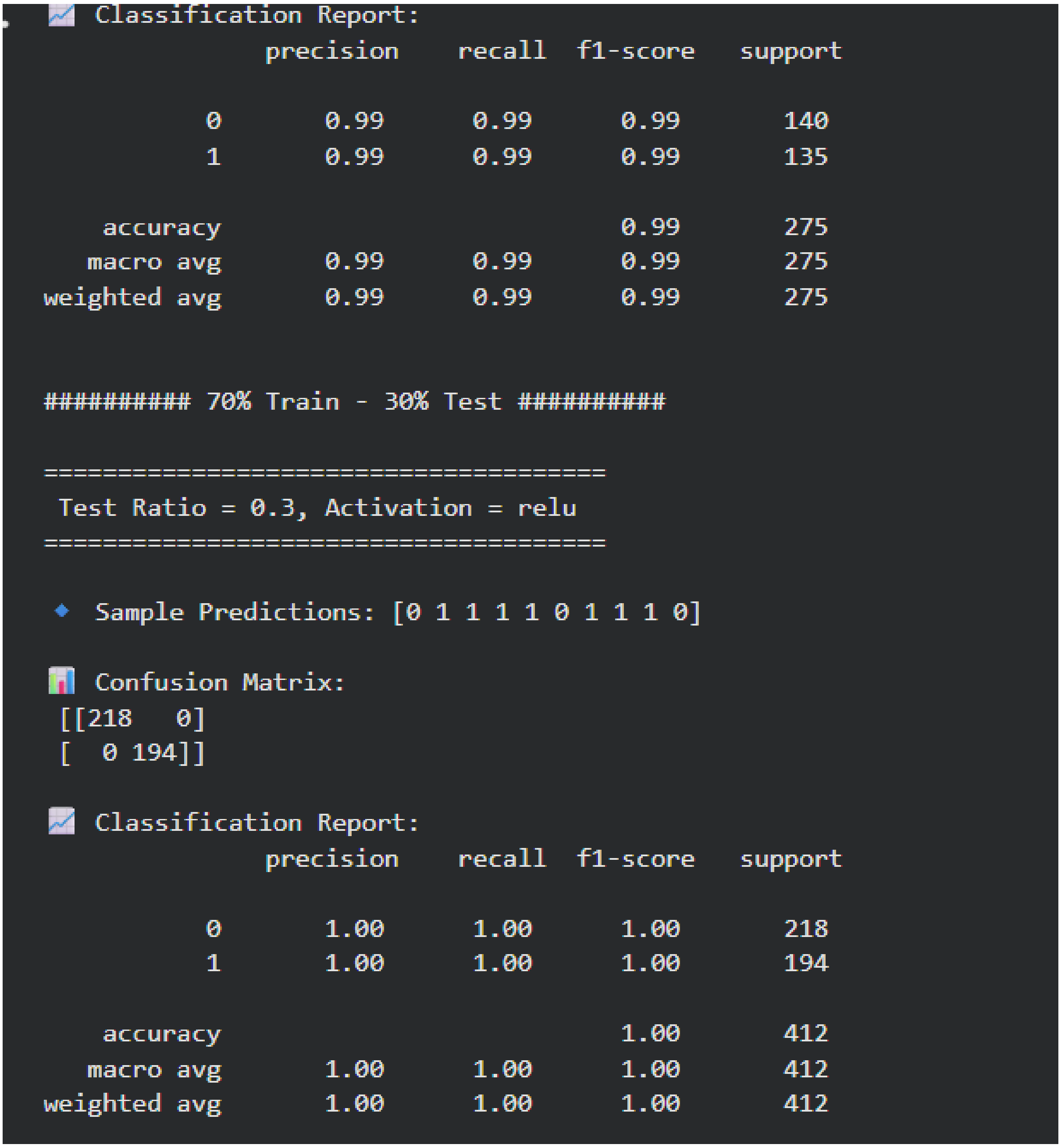
```
BankNote_Authentication.csv(text/csv) - 46442 bytes, last modified: 11/5/2025 - 100% done
Saving BankNote_Authentication.csv to BankNote_Authentication.csv
✅ Dataset Loaded Successfully

   variance  skewness  curtosis  entropy  class
0   3.62160    8.6661   -2.8073 -0.44699      0
1   4.54590    8.1674   -2.4586 -1.46210      0
2   3.86600   -2.6383    1.9242  0.10645      0
3   3.45660    9.5228   -4.0112 -3.59440      0
4   0.32924   -4.4552    4.5718 -0.98880      0

########## 80% Train - 20% Test ##########

======================================
 Test Ratio = 0.2, Activation = relu
======================================

🔹 Sample Predictions: [0 0 0 0 1 1 1 0 0 0]

📊 Confusion Matrix:
 [[160   0]
 [  0 115]]

📈 Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       160
           1       1.00      1.00      1.00       115
```



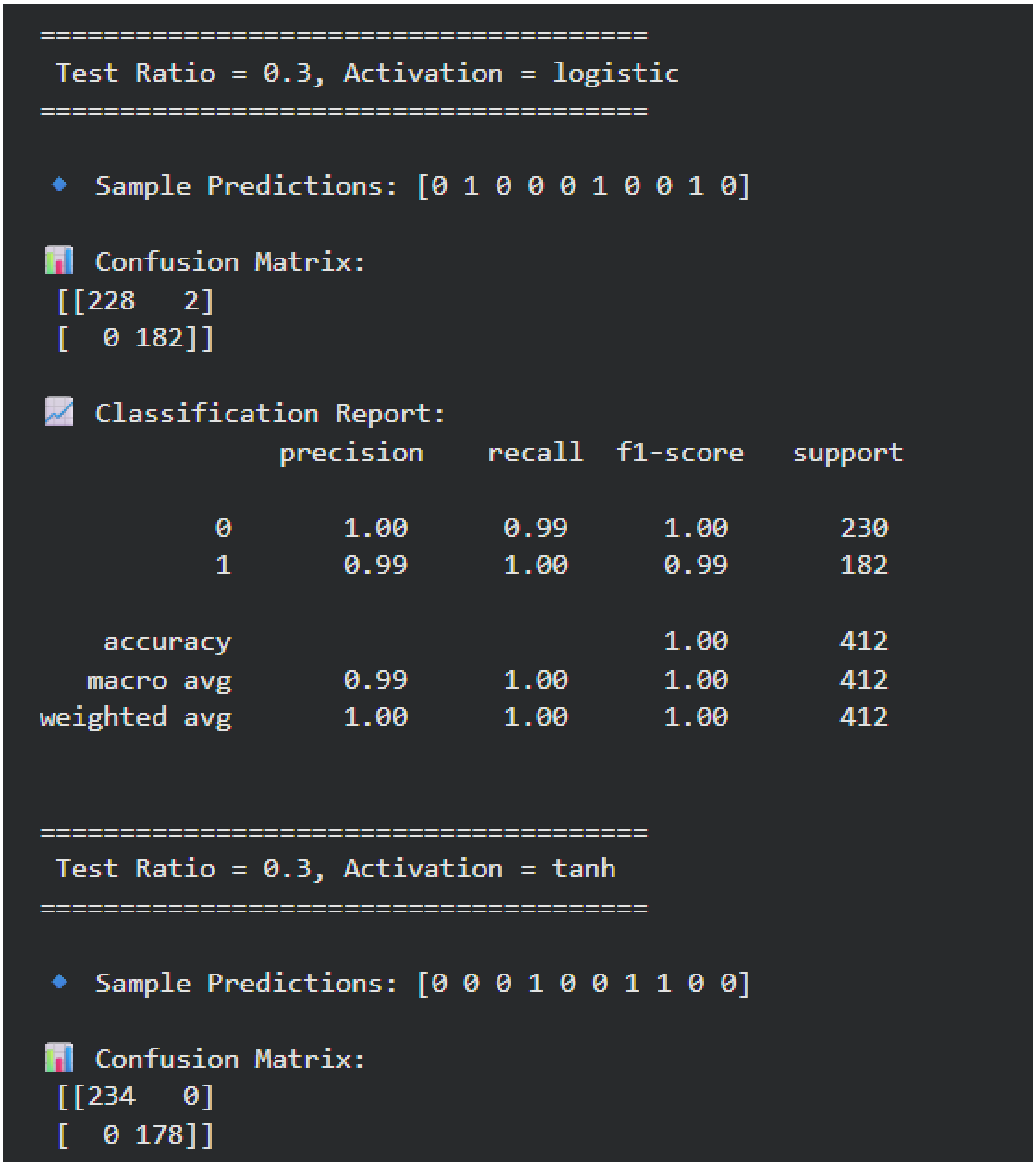
```
•••  📈 Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       160
           1       1.00      1.00      1.00       115

    accuracy                           1.00       275
   macro avg       1.00      1.00      1.00       275
weighted avg       1.00      1.00      1.00       275



======================================
 Test Ratio = 0.2, Activation = logistic
======================================

🔹 Sample Predictions: [1 1 1 1 1 1 0 1 0 0]

📊 Confusion Matrix:
 [[156   0]
 [  0 119]]

📈 Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       156
           1       1.00      1.00      1.00       119

    accuracy                           1.00       275
   macro avg       1.00      1.00      1.00       275
weighted avg       1.00      1.00      1.00       275
```



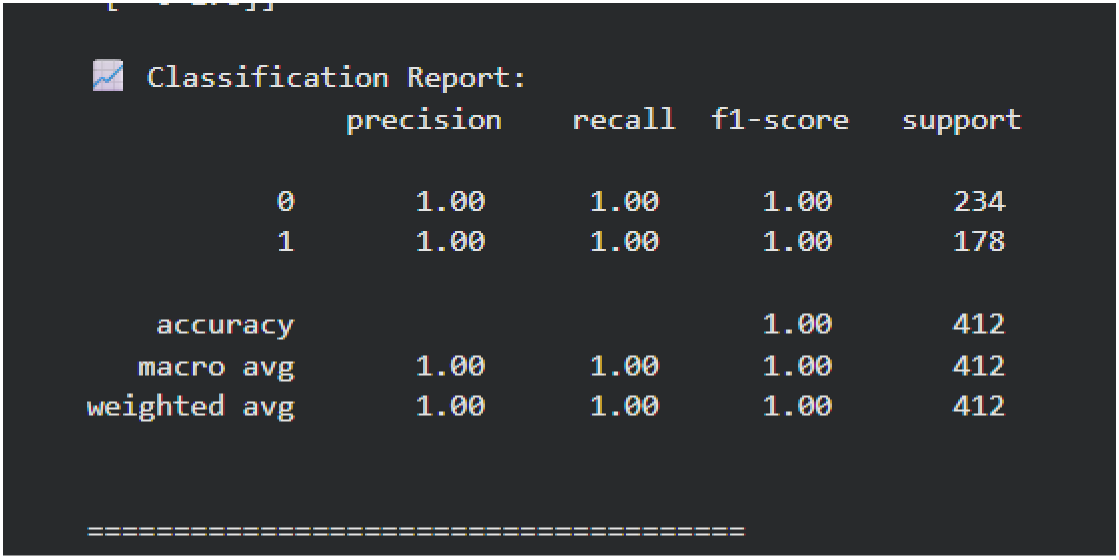
```
========================================
 Test Ratio = 0.2, Activation = tanh
========================================

🔹 Sample Predictions: [0 1 0 0 1 0 0 1 0 1]

📊 Confusion Matrix:
 [[143   0]
 [  0 132]]

📈 Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       143
           1       1.00      1.00      1.00       132

    accuracy                           1.00       275
   macro avg       1.00      1.00      1.00       275
weighted avg       1.00      1.00      1.00       275



========================================
 Test Ratio = 0.2, Activation = identity
========================================

🔹 Sample Predictions: [1 0 1 0 1 0 1 1 1 1]

📊 Confusion Matrix:
 [[139   1]
 [  1 134]]
```



```
📈 Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       140
           1       0.99      0.99      0.99       135

    accuracy                           0.99       275
   macro avg       0.99      0.99      0.99       275
weighted avg       0.99      0.99      0.99       275


########## 70% Train - 30% Test ##########

======================================
 Test Ratio = 0.3, Activation = relu
======================================

🔹 Sample Predictions: [0 1 1 1 1 0 1 1 1 0]

📊 Confusion Matrix:
 [[218   0]
 [  0 194]]

📈 Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       218
           1       1.00      1.00      1.00       194

    accuracy                           1.00       412
   macro avg       1.00      1.00      1.00       412
weighted avg       1.00      1.00      1.00       412
```



```
=======================================
 Test Ratio = 0.3, Activation = logistic
=======================================

 🔹 Sample Predictions: [0 1 0 0 0 1 0 0 1 0]

📊 Confusion Matrix:
 [[228   2]
 [  0 182]]

📈 Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       230
           1       0.99      1.00      0.99       182

    accuracy                           1.00       412
   macro avg       0.99      1.00      1.00       412
weighted avg       1.00      1.00      1.00       412


=======================================
 Test Ratio = 0.3, Activation = tanh
=======================================

 🔹 Sample Predictions: [0 0 0 1 0 0 1 1 0 0]

📊 Confusion Matrix:
 [[234   0]
 [  0 178]]
```



```
📈 Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       234
           1       1.00      1.00      1.00       178

    accuracy                           1.00       412
   macro avg       1.00      1.00      1.00       412
weighted avg       1.00      1.00      1.00       412


=======================================
```