STM32-F401RE FNN Evaluation

```
import numpy as np
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix, classification_report
from sklearn.metrics import roc_auc_score
```

DATA

Dataset: 226 entry **Test_Dataset**: 25 entry

```
y_test =
[1.00,1.00,0.00,1.00,1.00,1.00,0.00,1.00,1.00,1.00,1.00,1.00,0.00,0.00
,0.00,0.00,1.00,0.00,1.00,0.00,1.00,0.00,0.00,0.00,1.00]
y_test_pred =
np.array([0.65,0.54,0.00,1.00,0.53,1.00,0.00,0.74,0.29,1.00,1.00,0.98,
0.00,0.00,0.00,0.00,1.00,0.00,1.00,0.01,1.00,0.07,0.27,0.01,1.00])
y_train =
[0.00,1.00,0.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,1.00,0.00,0.00,1.00
,0.00,0.00,1.00,0.00,0.00,1.00,0.00,1.00,1.00,1.00,1.00,1.00,1.00,0.00
,1.00,0.00,1.00,0.00,1.00,1.00,1.00,0.00,1.00,1.00,0.00,0.00,0.00,1.00
,0.00,1.00,1.00,0.00,1.00,0.00,0.00,1.00,1.00,0.00,1.00,0.00,0.00,1.00
,0.00,0.00,1.00,0.00,0.00,0.00,1.00,0.00,0.00,1.00,1.00,0.00,0.00,1.00
,0.00,1.00,1.00,1.00,1.00,1.00,1.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
,1.00,1.00,1.00,0.00,1.00,1.00,1.00,1.00,1.00,0.00,0.00,1.00,1.00,1.00
,0.00,1.00,1.00,1.00,1.00,0.00,1.00,1.00,0.00,0.00,0.00,1.00,1.00,1.00
,1.00,1.00,1.00,0.00,0.00,0.00,1.00,1.00,0.00,0.00,0.00,1.00,0.00,1.00
,0.00,0.00,1.00,1.00,1.00,0.00,1.00,1.00,1.00,1.00,0.00,1.00,0.00,1.00
,0.00,1.00,0.00,0.00,0.00,0.00,1.00,0.00,1.00,1.00,1.00,1.00,0.00,0.00
,1.00,0.00,1.00,0.00,0.00,1.00,1.00,1.00,0.00,0.00,0.00,1.00,0.00,1.00
,1.00,1.00,0.00,1.00,0.00,1.00,0.00,0.00,1.00,1.00,1.00,0.00,1.00,1.00
,0.00,0.00,1.00,1.00,1.00,0.00,1.00,0.00,1.00,1.00,1.00,1.00,0.00,1.00
,0.00,0.00,1.00,1.00,0.00,1.00,1.00,0.00,1.00,0.00,1.00,0.00,0.00,1.00
,1.00,1.00,1.00,1.00,1.00,0.00,0.00,1.00,0.00,0.00,0.00,0.00,1.00,0.00
,0.00,1.00]
y_train_pred =
np.array([0.29,1.00,0.51,1.00,1.00,1.00,1.00,1.00,1.00,0.21,1.00,0.01,
0.00,1.00,0.00,0.02,1.00,0.00,0.00,1.00,0.00,1.00,1.00,0.75,1.00,1.00,
1.00,0.00,1.00,0.04,1.00,0.00,1.00,1.00,0.34,0.00,1.00,0.99,0.00,0.24,
0.00,1.00,0.00,0.46,0.98,0.00,1.00,0.00,0.23,1.00,1.00,0.07,1.00,0.00,
0.00,1.00,0.00,0.00,0.98,0.25,0.00,0.00,1.00,0.11,0.28,1.00,1.00,0.02,
0.01,1.00,0.00,1.00,1.00,0.98,1.00,1.00,0.63,0.22,0.00,0.00,0.00,0.07,
0.00,0.13,1.00,1.00,1.00,0.00,1.00,1.00,1.00,1.00,1.00,0.00,0.06,0.98,
0.98,1.00,0.03,1.00,0.37,1.00,1.00,0.00,1.00,1.00,0.00,0.00,0.04,1.00,
0.98,0.98,1.00,1.00,0.97,0.00,0.00,0.00,1.00,1.00,0.00,0.00,0.00,1.00,
0.00,1.00,0.01,0.00,1.00,1.00,1.00,0.27,0.98,1.00,1.00,0.96,0.29,1.00,
0.00,1.00,0.00,1.00,0.20,0.01,0.00,0.12,1.00,0.00,0.98,0.75,1.00,0.98,
0.00,0.01,1.00,0.00,1.00,0.41,0.00,1.00,0.98,1.00,0.00,0.00,0.00,1.00,
0.00,1.00,1.00,1.00,0.00,0.89,0.00,1.00,0.14,0.00,0.98,1.00,1.00,0.00,
1.00,1.00,0.00,0.00,1.00,1.00,1.00,0.00,1.00,0.00,1.00,0.75,0.98,1.00,
0.01,0.64,0.01,0.00,1.00,1.00,0.23,0.96,0.96,0.00,1.00,0.00,1.00,0.01,
0.24,1.00,0.98,1.00,1.00,1.00,1.00,0.45,0.00,0.30,0.00,0.00,0.00,0.10,
1.00,0.00,0.00,1.00])
```

Set an "acceptance" limit

```
y_train_pred_bin = (y_train_pred >= 0.05).astype(int)
y_test_pred_bin = (y_test_pred >= 0.05).astype(int)
```

Helper function in order to evaluate our model

```
def evaluate_model(y_true, y_pred, dataset_name=""):
    print(f"=== Evaluation for {dataset_name} ===")
    print(f"Accuracy: {accuracy_score(y_true, y_pred):.4f}")
    print(f"Precision: {precision_score(y_true, y_pred):.4f}")
    print(f"Recall: {recall_score(y_true, y_pred):.4f}")
    print(f"F1 Score: {f1_score(y_true, y_pred):.4f}")
    print("Matrice di confusione:")
    print(confusion_matrix(y_true, y_pred))
    print("Report dettagliato:")
    print(classification_report(y_true, y_pred))
    print("\n")
```

Evaluation on training data

```
evaluate_model(y_train, y_train_pred_bin, "Training Set")
```

```
=== Evaluation for Training Set ===
Accuracy: 0.9027
Precision: 0.8514
Recall: 1.0000
F1 Score: 0.9197
Matrice di confusione:
[[ 78  22]
 [  0 126]]
Report dettagliato:
              precision    recall  f1-score   support

         0.0       1.00      0.78      0.88       100
         1.0       0.85      1.00      0.92       126

    accuracy                           0.90       226
   macro avg       0.93      0.89      0.90       226
weighted avg       0.92      0.90      0.90       226
```

```
auc = roc_auc_score(y_train, y_train_pred)
print("ROC AUC Score on training data:", auc)

ROC AUC Score on training data: 0.9981746031746032
```

Evaluation on test data

```
evaluate_model(y_test, y_test_pred_bin, "Test Set")

=== Evaluation for Test Set ===
Accuracy: 0.9200
Precision: 0.8750
Recall: 1.0000
F1 Score: 0.9333
Matrice di confusione:
[[ 9  2]
 [ 0 14]]
Report dettagliato:
              precision    recall  f1-score   support

         0.0       1.00      0.82      0.90        11
         1.0       0.88      1.00      0.93        14

    accuracy                           0.92        25
   macro avg       0.94      0.91      0.92        25
weighted avg       0.93      0.92      0.92        25
```

```
auc = roc_auc_score(y_test, y_test_pred)
print("ROC AUC Score on test data:", auc)

ROC AUC Score on test data: 1.0
```