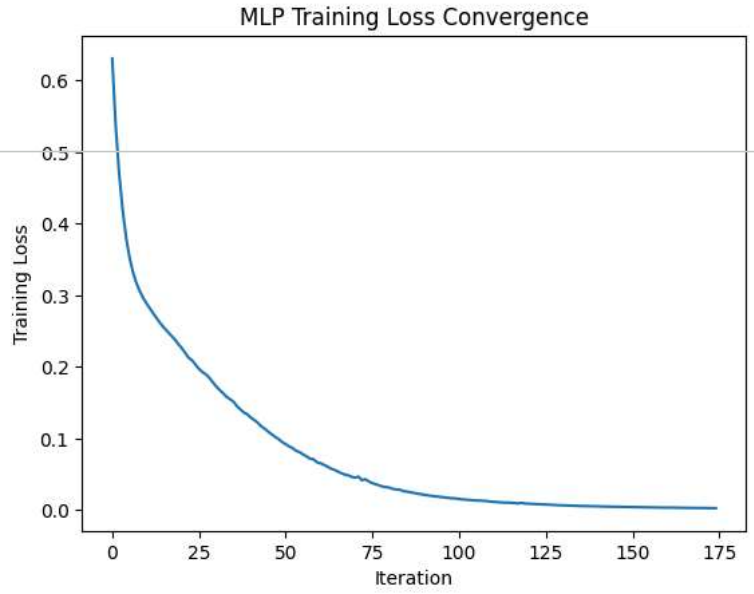
```
import numpy as np
import pandas as pd
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
import matplotlib.pyplot as plt

data=pd.read_csv('https://raw.githubusercontent.com/ElavarasanMurugan/EX-6-NN/main/heart.csv')
X=data.iloc[:,:-1].values
y=data.iloc[:,-1].values

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
scaler=StandardScaler()
X_train=scaler.fit_transform(X_train)
X_test=scaler.transform(X_test)

mlp=MLPClassifier(hidden_layer_sizes=(100,100),max_iter=1000,random_state=42)
training_loss=mlp.fit(X_train,y_train).loss_curve_
y_pred=mlp.predict(X_test)
accuracy=accuracy_score(y_test,y_pred)
print("Accuracy:",accuracy)

plt.plot(training_loss)
plt.title('MLP Training Loss Convergence')
plt.xlabel('Iteration')
plt.ylabel('Training Loss')
plt.show()

X=data.drop('target',axis=1)
y=data['target']
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
scaler=StandardScaler()
X_train_scaled=scaler.fit_transform(X_train)
X_test_scaled=scaler.transform(X_test)
mlp_classifier=MLPClassifier(hidden_layer_sizes=(64,),max_iter=1000,random_state=42)
mlp_classifier.fit(X_train_scaled,y_train)
y_pred=mlp_classifier.predict(X_test_scaled)

accuracy=accuracy_score(y_test,y_pred)
conf_matrix=confusion_matrix(y_test,y_pred)
classification_rep=classification_report(y_test,y_pred)

print("Accuracy:",accuracy)
print("Confusion Matrix:\n",conf_matrix)
print("Classification Report:\n",classification_rep)
```

```
Accuracy: 0.9853658536585366
```




```
Accuracy: 0.9853658536585366
Confusion Matrix:
 [[102   0]
 [  3 100]]
Classification Report:
```