In [ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import confusion_matrix, classification_report, Confusion
```

In [ ]:
```
import glob
files=glob.glob('/content/drive/My Drive/MACHINE LEARNING/Colab Notebooks/*.cs
for file in files:
  print(file)
```

In [ ]:
```
file_path='/content/drive/My Drive/MACHINE LEARNING/BankNote_Authentication.cs
df=pd.read_csv(file_path)
```

In [ ]:
```
X = df.drop('class', axis=1).values
y = df['class'].values
```

In [ ]:
```
X_train, X_test, y_train, y_test = train_test_split(
      X, y, test_size=0.2, random_state=42
      )
```

In [ ]:
```
mlp = MLPClassifier(
      hidden_layer_sizes=(10, 10),
      activation='relu',
      solver='adam',
      learning_rate_init=0.001,
      max_iter=500,
      early_stopping=True,
      validation_fraction=0.1,
      random_state=42
      )
```

In [ ]:
```
mlp.fit(X_train, y_train)
```

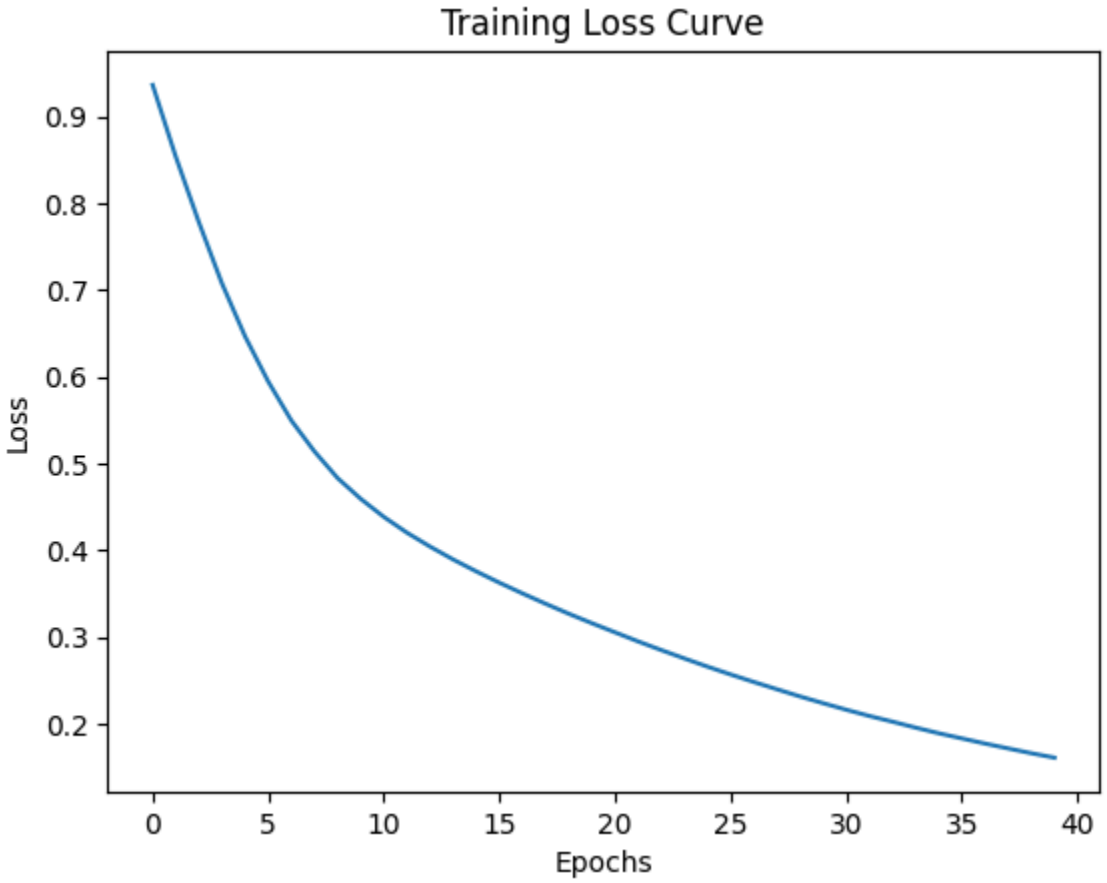
Out[ ]:
```
MLPClassifier(early_stopping=True, hidden_layer_sizes=(10, 10), max_i
ter=500,
              random_state=42)
```

In [ ]:
```
y_pred = mlp.predict(X_test)
```

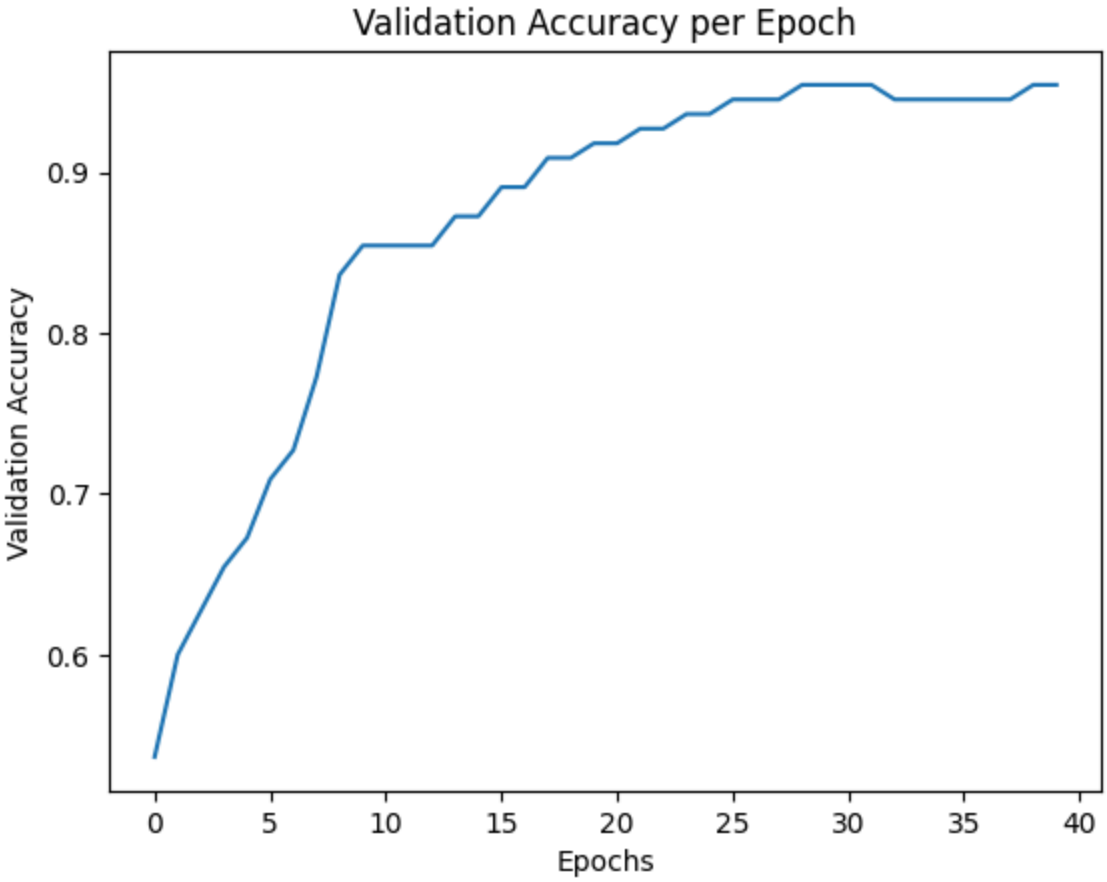
In [ ]:
```
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n",cm)
```

```
Confusion Matrix:
 [[147   1]
 [ 26 101]]
```

In [ ]:
```python
plt.plot(mlp.loss_curve_)
plt.title("Training Loss Curve")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.show()
```



In [ ]:
```python
if hasattr(mlp, 'validation_scores_'):
  plt.plot(mlp.validation_scores_)
  plt.title("Validation Accuracy per Epoch")
  plt.xlabel("Epochs")
  plt.ylabel("Validation Accuracy")
  plt.show()
```


Validation Accuracy per Epoch
Validation Accuracy
0.6
0.7
0.8
0.9
0
5
10
15
20
25
30
35
40
Epochs