Assignment No : 4

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Load dataset (Replace 'banknote_data.csv' with your actual dataset file)
df = pd.read_csv("/content/sample_data/data_banknote_authentication.csv")

# Display the first few rows of the dataset
print("\nFirst 5 rows of the dataset:")
print(df.head())
```

Output:-

```
First 5 rows of the dataset:
    3.6216  8.6661  -2.8073  -0.44699  0
0  4.54590  8.1674  -2.4586  -1.46210  0
1  3.86600 -2.6383   1.9242   0.10645  0
2  3.45660  9.5228  -4.0112  -3.59440  0
3  0.32924 -4.4552   4.5718  -0.98880  0
4  4.36840  9.6718  -3.9606  -3.16250  0
```

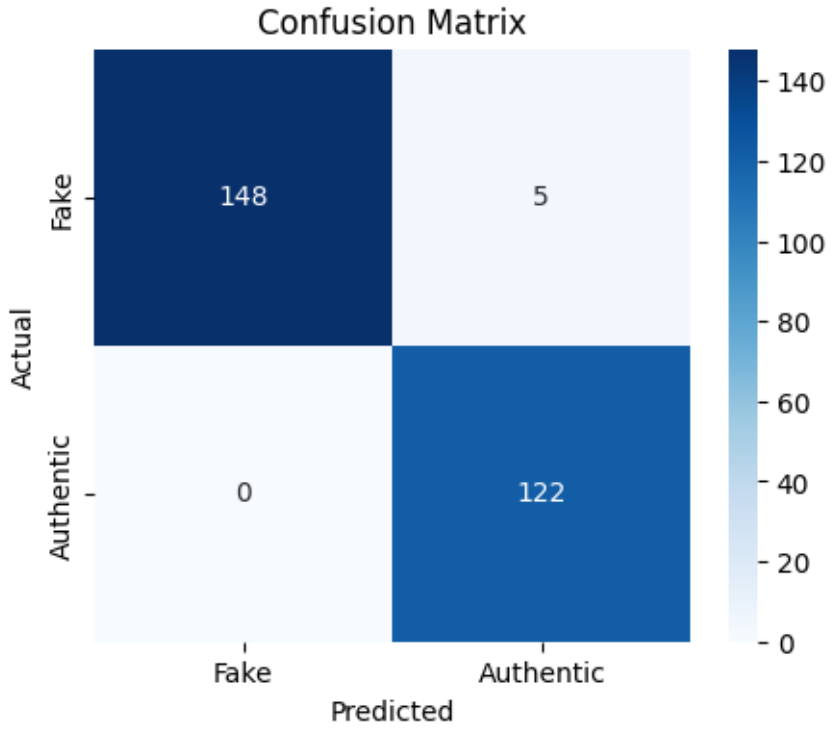
```
# Splitting features and target variable
X = df.iloc[:, :-1]
y = df.iloc[:, -1]
# Splitting dataset into training (80%) and testing (20%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42,
stratify=y)

# Standardize the features (important for logistic regression)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train Logistic Regression model
model = LogisticRegression()
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Generate Confusion Matrix BEFORE accuracy
conf_matrix = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix:")
print(conf_matrix)
```

Output:

```
Confusion Matrix:
[[148   5]
 [  0 122]]
```

```
plt.figure(figsize=(5, 4))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=["Fake", "Authentic"], yticklabels=["Fake", "Authentic"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```

Output:




```
# Classification report
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

Output:

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.97	0.98	153
1	0.96	1.00	0.98	122

```
    accuracy                           0.98       275
   macro avg       0.98      0.98      0.98       275
weighted avg       0.98      0.98      0.98       275
```

```
# Now, calculate model accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"\nModel Accuracy: {accuracy:.2f}")
```

Output:

```
Model Accuracy: 0.98
```