PREPROCESSING DATA

```
import pandas as pd
from sklearn import datasets

#load dataset
digits = datasets.load_digits()

#Identification
X = digits.data
y = digits.target

# Convert data and target features to a DataFrame
df_X = pd.DataFrame(X, columns=digits.feature_names)
df_y = pd.Series(y, name='target')

df_X
```

{"type":"dataframe","variable_name":"df_X"}

```
df_y
```

```
0       0
1       1
2       2
3       3
4       4
       ..
1792    9
1793    0
1794    8
1795    9
1796    8
Name: target, Length: 1797, dtype: int64
```

```
df = pd.concat([df_X, df_y], axis=1)

df
```

{"type":"dataframe","variable_name":"df"}

```
df.target.unique()
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1797 entries, 0 to 1796
Data columns (total 65 columns):
 #   Column     Non-Null Count  Dtype
```

```
---  ------      --------------  -----
 0   pixel_0_0  1797 non-null   float64
 1   pixel_0_1  1797 non-null   float64
 2   pixel_0_2  1797 non-null   float64
 3   pixel_0_3  1797 non-null   float64
 4   pixel_0_4  1797 non-null   float64
 5   pixel_0_5  1797 non-null   float64
 6   pixel_0_6  1797 non-null   float64
 7   pixel_0_7  1797 non-null   float64
 8   pixel_1_0  1797 non-null   float64
 9   pixel_1_1  1797 non-null   float64
 10  pixel_1_2  1797 non-null   float64
 11  pixel_1_3  1797 non-null   float64
 12  pixel_1_4  1797 non-null   float64
 13  pixel_1_5  1797 non-null   float64
 14  pixel_1_6  1797 non-null   float64
 15  pixel_1_7  1797 non-null   float64
 16  pixel_2_0  1797 non-null   float64
 17  pixel_2_1  1797 non-null   float64
 18  pixel_2_2  1797 non-null   float64
 19  pixel_2_3  1797 non-null   float64
 20  pixel_2_4  1797 non-null   float64
 21  pixel_2_5  1797 non-null   float64
 22  pixel_2_6  1797 non-null   float64
 23  pixel_2_7  1797 non-null   float64
 24  pixel_3_0  1797 non-null   float64
 25  pixel_3_1  1797 non-null   float64
 26  pixel_3_2  1797 non-null   float64
 27  pixel_3_3  1797 non-null   float64
 28  pixel_3_4  1797 non-null   float64
 29  pixel_3_5  1797 non-null   float64
 30  pixel_3_6  1797 non-null   float64
 31  pixel_3_7  1797 non-null   float64
 32  pixel_4_0  1797 non-null   float64
 33  pixel_4_1  1797 non-null   float64
 34  pixel_4_2  1797 non-null   float64
 35  pixel_4_3  1797 non-null   float64
 36  pixel_4_4  1797 non-null   float64
 37  pixel_4_5  1797 non-null   float64
 38  pixel_4_6  1797 non-null   float64
 39  pixel_4_7  1797 non-null   float64
 40  pixel_5_0  1797 non-null   float64
 41  pixel_5_1  1797 non-null   float64
 42  pixel_5_2  1797 non-null   float64
 43  pixel_5_3  1797 non-null   float64
 44  pixel_5_4  1797 non-null   float64
 45  pixel_5_5  1797 non-null   float64
 46  pixel_5_6  1797 non-null   float64
 47  pixel_5_7  1797 non-null   float64
```

```
 48  pixel_6_0  1797 non-null   float64
 49  pixel_6_1  1797 non-null   float64
 50  pixel_6_2  1797 non-null   float64
 51  pixel_6_3  1797 non-null   float64
 52  pixel_6_4  1797 non-null   float64
 53  pixel_6_5  1797 non-null   float64
 54  pixel_6_6  1797 non-null   float64
 55  pixel_6_7  1797 non-null   float64
 56  pixel_7_0  1797 non-null   float64
 57  pixel_7_1  1797 non-null   float64
 58  pixel_7_2  1797 non-null   float64
 59  pixel_7_3  1797 non-null   float64
 60  pixel_7_4  1797 non-null   float64
 61  pixel_7_5  1797 non-null   float64
 62  pixel_7_6  1797 non-null   float64
 63  pixel_7_7  1797 non-null   float64
 64  target     1797 non-null   int64
dtypes: float64(64), int64(1)
memory usage: 912.7 KB
```

SPLITTING DATA

```
from sklearn.model_selection import train_test_split

# Membagi data menjadi train dan test
X_train, X_test, y_train, y_test = train_test_split(df_X, df_y,
test_size=0.2, random_state=42)
```

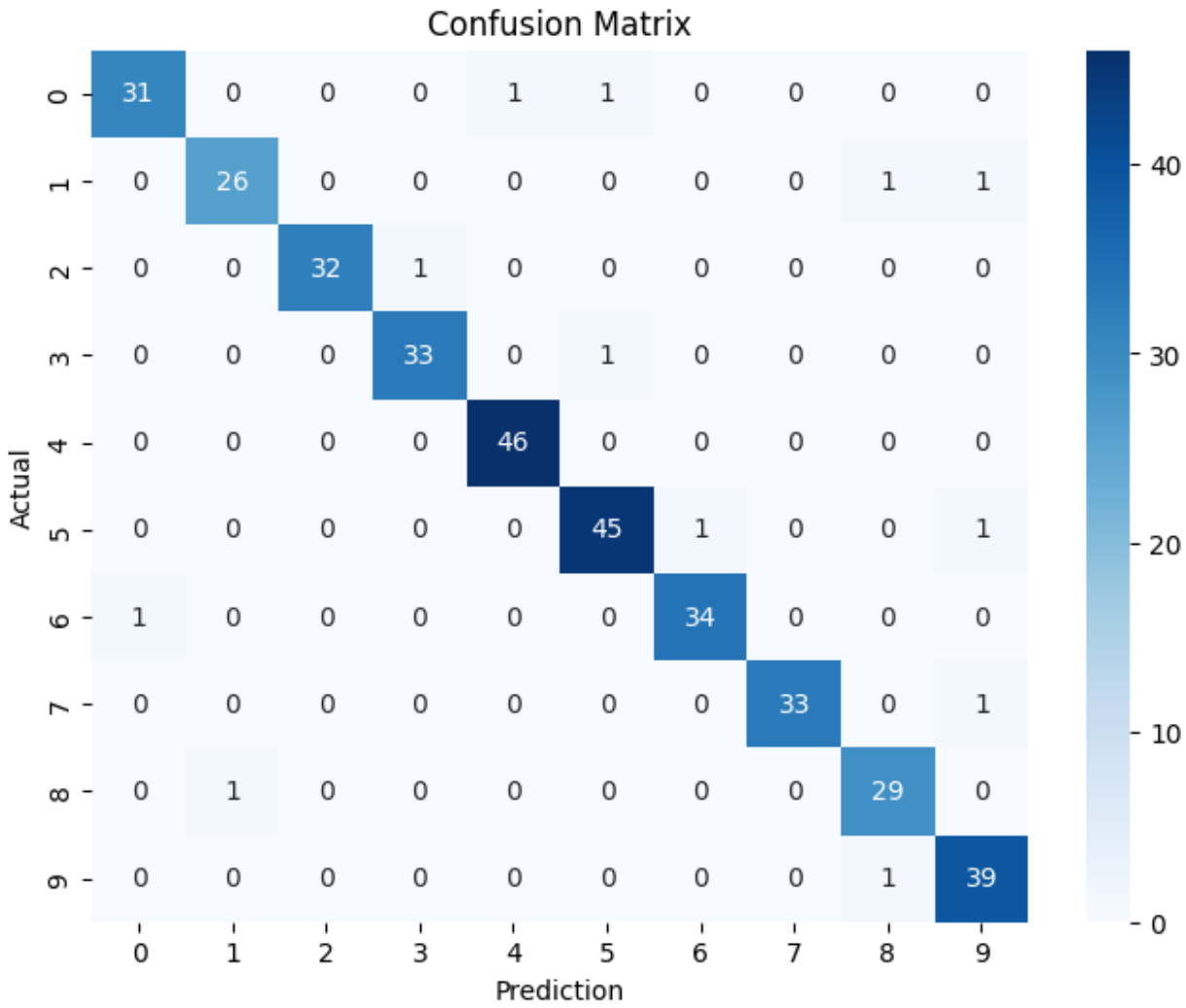
USING MULTILAYER PERCEPTRON ALGORITHM

```
from sklearn.neural_network import MLPClassifier

model = MLPClassifier(hidden_layer_sizes=(100, 50), max_iter=300,
random_state=42)
model.fit(X_train, y_train)
```

```
MLPClassifier(hidden_layer_sizes=(100, 50), max_iter=300,
random_state=42)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report

# Memprediksi dan mengevaluasi
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print("Classification Report:")
print(f"Accuration: {accuracy * 100:.2f}%")
```

```
Classification Report:
Accuration: 96.67%

import seaborn as sns
import matplotlib.pyplot as plt

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=digits.target_names,
            yticklabels=digits.target_names)
plt.xlabel("Prediction")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```



```python
from sklearn.metrics import accuracy_score, classification_report

target_digits_names = [str(name) for name in digits.target_names]

print(classification_report(y_test,
                            y_pred,
                            target_names=target_digits_names))
```

```
              precision    recall  f1-score   support

           0       0.97      0.94      0.95        33
           1       0.96      0.93      0.95        28
           2       1.00      0.97      0.98        33
           3       0.97      0.97      0.97        34
           4       0.98      1.00      0.99        46
           5       0.96      0.96      0.96        47
           6       0.97      0.97      0.97        35
           7       1.00      0.97      0.99        34
           8       0.94      0.97      0.95        30
           9       0.93      0.97      0.95        40

    accuracy                           0.97       360
   macro avg       0.97      0.96      0.97       360
weighted avg       0.97      0.97      0.97       360
```