1.

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns

# The handwritten digits dataset contains 1797 images where each image is 8x8
# Thus, we have 64 features (8x8)
# X: features (64)
# y: label (0-9)
# Load the digits dataset
digits = load_digits()
X, y = digits.data, digits.target
print(f'Shape X: {X.shape}')
print(f'Shape y: {y.shape}')


# Visualize some samples
fig, axes = plt.subplots(1, 5, figsize=(10, 3))
for ax, idx in zip(axes, range(5)):
    ax.imshow(digits.images[idx], cmap='gray')
    ax.set_title(f'Label: {digits.target[idx]}')
    ax.axis('off')
plt.show()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

print(X.shape)
print(X_test.shape)

knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

pred = knn.predict(X_test)

print('Model accuracy score: ', accuracy_score(y_test, pred))

conf_matrix = confusion_matrix(y_test, pred)
print(f'\nConfusion Matrix: \n{conf_matrix}')

# confustion matrix
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=knn.classes_, yticklabels=knn.classes_)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

```python
# Visualize some samples
fig, axes = plt.subplots(1, 5, figsize=(10, 3))
for ax, idx in zip(axes, range(5)):
    ax.imshow(X_test[idx].reshape(8,8), cmap='gray')
    ax.set_title(f'Label: {y_test[idx]} Predicted: {pred[idx]}')
    ax.axis('off')
plt.show()
```

```
_train, y_test = train_test_split(X, y, test_size=0.20)
```

C:\WINDOWS\system32\cmd.

```
Shape X: (1797, 64)
Shape y: (1797,)
(1797, 64)
(360, 64)
Model accuracy score:  0.9805555555555555

Confusion Matrix:
[[38  0  0  0  0  0  0  0  0  0]
 [ 0 43  0  0  0  0  0  0  0  0]
 [ 0  0 26  0  0  0  0  0  0  0]
 [ 0  0  0 32  0  0  0  1  0  0]
 [ 0  0  0  0 42  0  0  0  0  0]
 [ 0  0  0  0  0 36  1  0  0  1]
 [ 0  0  0  0  0  0 38  0  0  0]
 [ 0  0  0  0  0  0  0 35  0  0]
 [ 0  1  0  0  0  0  0  0 32  0]
 [ 0  1  0  1  1  0  0  0  0 31]]
```

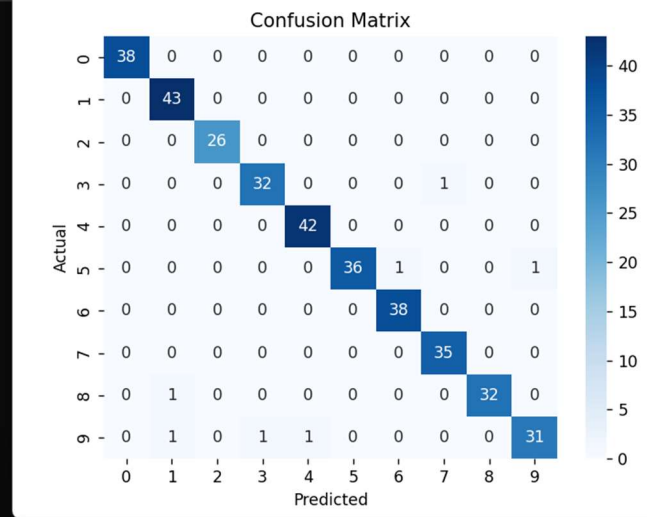
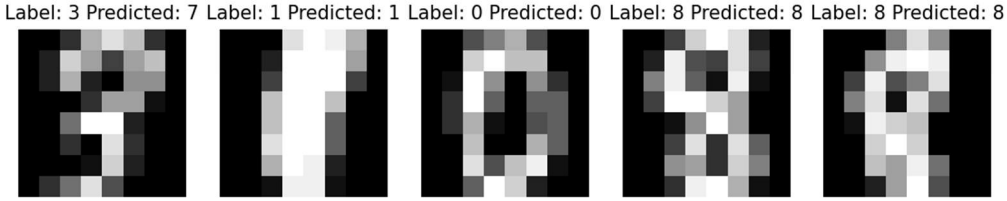


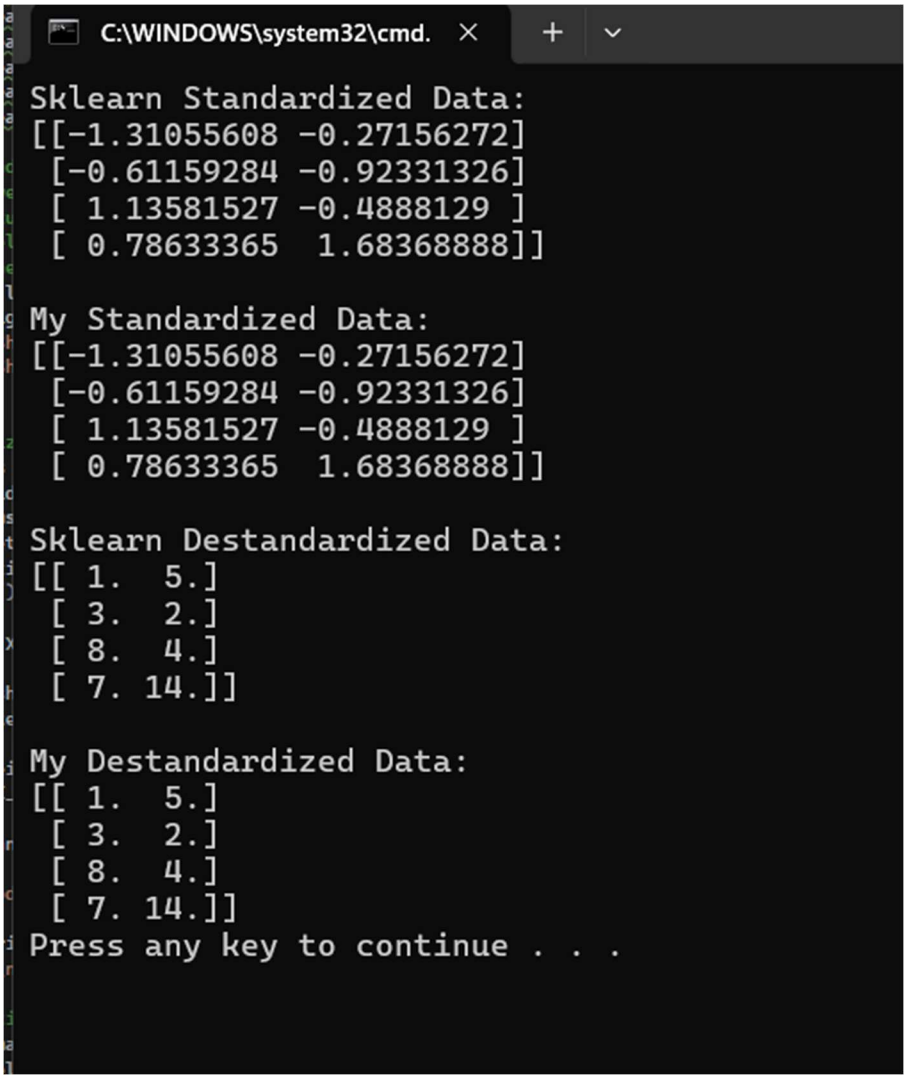
Figure 1



2.

```python
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns

# The handwritten digits dataset contains 1797 images where each image is 8x8
# Thus, we have 64 features (8x8)
# X: features (64)
# y: label (0-9)
# Load the digits dataset
digits = load_digits()
X, y = digits.data, digits.target
print(f'Shape X: {X.shape}')
print(f'Shape y: {y.shape}')


# Visualize some samples
fig, axes = plt.subplots(1, 5, figsize=(10, 3))
for ax, idx in zip(axes, range(5)):
    ax.imshow(digits.images[idx], cmap='gray')
    ax.set_title(f'Label: {digits.target[idx]}')
    ax.axis('off')
plt.show()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

print(X.shape)
print(X_test.shape)

knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

pred = knn.predict(X_test)

print('Model accuracy score: ', accuracy_score(y_test, pred))

conf_matrix = confusion_matrix(y_test, pred)
print(f'\nConfusion Matrix: \n{conf_matrix}')

# confustion matrix
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues',
xticklabels=knn.classes_, yticklabels=knn.classes_)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

```python
# Visualize some samples
fig, axes = plt.subplots(1, 5, figsize=(10, 3))
for ax, idx in zip(axes, range(5)):
    ax.imshow(X_test[idx].reshape(8,8), cmap='gray')
    ax.set_title(f'Label: {y_test[idx]} Predicted: {pred[idx]}')
    ax.axis('off')
plt.show()
```


C:\WINDOWS\system32\cmd.
Sklearn Standardized Data:
[[-1.31055608 -0.27156272]
[-0.61159284 -0.92331326]
[1.13581527 -0.4888129]
[0.78633365 1.68368888]]
My Standardized Data:
[[-1.31055608 -0.27156272]
[-0.61159284 -0.92331326]
[1.13581527 -0.4888129]
[0.78633365 1.68368888]]
Sklearn Destandardized Data:
[[1. 5.]
[3. 2.]
[8. 4.]
[7. 14.]]
My Destandardized Data:
[[1. 5.]
[3. 2.]
[8. 4.]
[7. 14.]]
Press any key to continue . . .