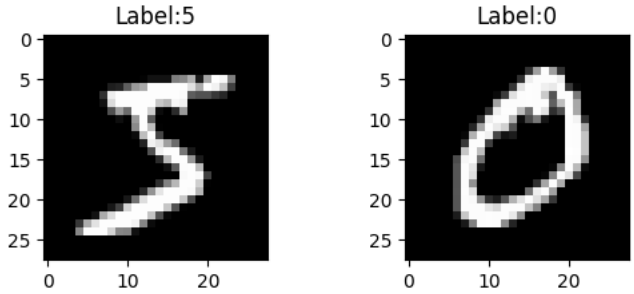
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.metrics import confusion_matrix,accuracy_score,ConfusionMatrixDisplay
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from tensorflow.keras.datasets import mnist
import sklearn.metrics


(train_images,train_label),(test_images,test_label)=mnist.load_data()
print('Training data shape',train_images.shape)
print('Training label shape',train_label.shape)
print('Testing data shape',test_images.shape)
print('Testing label shape',test_label.shape)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 1s 0us/step
Training data shape (60000, 28, 28)
Training label shape (60000,)
Testing data shape (10000, 28, 28)
Testing label shape (10000,)
```

```
plt.subplot(221)
plt.title('Label:5')
plt.imshow(train_images[0],cmap=plt.get_cmap('gray'))
plt.subplot(222)
plt.title('Label:0')
plt.imshow(train_images[1],cmap=plt.get_cmap('gray'))
```

```
<matplotlib.image.AxesImage at 0x795cb3ccefe0>
```




```
train_images = train_images/255.0
test_images = test_images/255.0

train_images_flat = train_images.reshape(train_images.shape[0],-1)
test_images_flat = test_images.reshape(test_images.shape[0],-1)


# X_train,X_val,y_train,y_val = train_test_split(train_images_flat,train_label,test_size=0.2,random_state=42)

print('train shape',train_images_flat.shape)

print('Testing shape',test_images_flat.shape)
```

```
train shape (60000, 784)
Testing shape (10000, 784)
```

```
# training the SVC model
# svc = SVC()
# svc.fit(X_train,y_train)

# y_val_predict = svc.predict(X_val)
# accuracy_val = accuracy_score(y_val,y_val_predict)
# print('Accuracy Score',accuracy_val)

# applying Kmeans
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=10,random_state=42)
kmeans.fit(train_images_flat)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:1416: FutureWarning: The default value of `n_init` will ch
  super()._check_params_vs_input(X, default_n_init=10)
```

KMeans

KMeans(n_clusters=10, random_state=42)

```
# y_test_predict = svc.predict(test_images_flat)
# accuracy_test = accuracy_score(test_label,y_test_predict)
# print(' Test Accuracy Score',accuracy_test)

y_test_predict = kmeans.predict(test_images_flat)
accuracy_test = accuracy_score(test_label,y_test_predict)
print(' Test Accuracy Score',accuracy_test)

print("y_test_predict :",y_test_predict)

# computer cluster centers
cluster_centers = kmeans.cluster_centers_
print("Cluster : ",cluster_centers)

# calculate RMSE
rmse = sklearn.metrics.mean_squared_error(test_label,y_test_predict,squared=False)
print('RMSE',rmse)
```

```
 Test Accuracy Score 0.1369
y_test_predict : [6 8 5 ... 6 6 0]
Cluster :  [[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
RMSE 3.9770089263163593
```