```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_olivetti_faces
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score,confusion_matrix
data=fetch_olivetti_faces()
X,y=data.data, data.target
images=data.images
plt.figure(figsize=(10,4))
for i in range(10):
    plt.subplot(2,5,i+1)
    plt.imshow(images[i],cmap='gray')
    plt.title(f"Label:{y[i]}")
    plt.axis('off')
plt.suptitle("Sample Faces from Olivetti Dataset")
plt.tight_layout()
plt.show()
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=42)
model=GaussianNB()
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
accuracy=accuracy_score(y_test,y_pred)
print(f"Naive Bayes Classifier Accuracy:{round(accuracy*100,2)}%")
misclassified=np.where(y_pred!=y_test)[0]
if len(misclassified) > 0:
    plt.figure(figsize=(10,3))
    for i in range(min(5,len(misclassified))):
        idx=misclassified[i]
        plt.subplot(1,5,i+1)
        plt.imshow(X_test[idx].reshape(64,64),cmap='gray')
        plt.title(f"True:{y_test[idx]}\nPred: {y_pred[idx]}", fontsize=8)
        plt.axis('off')
    plt.suptitle("Misclassified Samples")
    plt.tight_layout()
    plt.show()
else:
    print("No misclassified samples to display.")
```