Experiment - 2 PCA

Code

```
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import fetch_openml
import matplotlib.pyplot as plt
import seaborn as sns

# For 3D plotting
from mpl_toolkits.mplot3d import Axes3D

# Fetch the MNIST dataset from openml
mnist = fetch_openml('mnist_784')
df_mnist = pd.DataFrame(data=mnist.data)
df_mnist['target'] = mnist.target

# Standardize the data
features_mnist = df_mnist.columns[:-1]
x_mnist = df_mnist.loc[:, features_mnist].values
y_mnist = df_mnist.loc[:, ['target']].values
x_mnist = StandardScaler().fit_transform(x_mnist)

# Apply PCA
pca_mnist = PCA(n_components=3)
principal_components_mnist = pca_mnist.fit_transform(x_mnist)
principal_df_mnist = pd.DataFrame(data=principal_components_mnist, columns=['PC1', 'PC2', 'PC3'])
final_df_mnist = pd.concat([principal_df_mnist, df_mnist[['target']]], axis=1)

# Check Components
print("PCA Components (MNIST):\n", pca_mnist.components_)
print("Explained Variance Ratio (MNIST):\n", pca_mnist.explained_variance_ratio_)

# Visualize and Save the Principal Components
fig_mnist = plt.figure(figsize=(8, 6))
ax_mnist = fig_mnist.add_subplot(111, projection='3d')
ax_mnist.set_title('3D PCA Plot (MNIST)')
targets_mnist = np.unique(df_mnist['target'])
colors_mnist = plt.cm.rainbow(np.linspace(0, 1, len(targets_mnist)))

for target, color in zip(targets_mnist, colors_mnist):
    indices_mnist = final_df_mnist['target'] == target
    ax_mnist.scatter(final_df_mnist.loc[indices_mnist, 'PC1'],
                     final_df_mnist.loc[indices_mnist, 'PC2'],
                     final_df_mnist.loc[indices_mnist, 'PC3'],
                     c=[color],
                     s=10,
```

```
                        label=target)
ax_mnist.set_xlabel('Principal Component 1')
ax_mnist.set_ylabel('Principal Component 2')
ax_mnist.set_zlabel('Principal Component 3')
ax_mnist.legend(targets_mnist)

# Save the plot as an image file
plt.savefig('pca_mnist_plot.png')
plt.show()

# Calculate Variance Ratio
print("Explained Variance Ratio (MNIST):", pca_mnist.explained_variance_ratio_)
```

Output