Pca:

import pickle

from sklearn.decomposition import PCA

from sklearn.datasets import make\_classification

import matplotlib.pyplot as plt

# --- Replace below with your dataset ---

X, \_ = make\_classification(n\_samples=1000, n\_features=10, random\_state=42) # Dummy dataset for PCA

# ----------------------------------------

pca = PCA(n\_components=2) # Reduce to 2 dimensions for visualization (you can change n\_components)

X\_pca = pca.fit\_transform(X)

# Plotting the results (optional)

plt.scatter(X\_pca[:, 0], X\_pca[:, 1])

plt.title('PCA - 2D Projection')

plt.xlabel('Principal Component 1')

plt.ylabel('Principal Component 2')

plt.show()

# Save the PCA model

with open('pca\_model.pkl', 'wb') as f:

pickle.dump(pca, f)

print("PCA model saved as pca\_model.pkl")

Lda:

import pickle

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis as LDA

from sklearn.datasets import make\_classification

import matplotlib.pyplot as plt

# --- Replace below with your dataset ---

X, y = make\_classification(n\_samples=1000, n\_features=10, n\_classes=2, random\_state=42) # Dummy dataset for classification

# ----------------------------------------

lda = LDA(n\_components=1) # Reduce to 1 component for visualization (adjust based on your needs)

X\_lda = lda.fit\_transform(X, y)

# Plotting the results (optional)

plt.scatter(X\_lda, y, c=y, cmap='viridis')

plt.title('LDA - 1D Projection')

plt.xlabel('Linear Discriminant Component')

plt.ylabel('Class')

plt.show()

# Save the LDA model

with open('lda\_model.pkl', 'wb') as f:

pickle.dump(lda, f)

print("LDA model saved as lda\_model.pkl")