#CBB Code

import pandas as pd

import numpy as np

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

from sklearn.cluster import KMeans

from sklearn.metrics import silhouette\_score

from sklearn.manifold import TSNE

from umap import UMAP

import matplotlib.pyplot as plt

df = pd.read\_csv('/Users/kkelley/Desktop/CBB Results/CBBML.csv')

print(df[df.isnull().any(axis=1)])

df.dropna(inplace=True)

print(df[df.isnull().any(axis=1)])

# Round numeric columns to 2 decimal places for clarity

df = df.round(2)

# Extract columns relevant for unsupervised learning (you can adjust as necessary)

features = ['Away Team', 'Home Team', 'Away Score',

'Home Score', 'Away OR', 'Away DR', 'Away AT', 'Home OR', 'Home DR', 'Home AT', 'delta OR', 'delta DR', 'delta AT']

# Select only the numeric columns

X = df[features]

# Normalize the data

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X)

# Perform PCA to reduce dimensionality

pca = PCA(n\_components=2)

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

# Apply K-Means clustering

kmeans = KMeans(n\_clusters=2) # Set the number of clusters you want to create

df['Cluster'] = kmeans.fit\_predict(X\_scaled)

# Plot the PCA results

plt.figure(figsize=(8, 6))

plt.scatter(X\_pca[:, 0], X\_pca[:, 1], c=df['Cluster'], cmap='viridis')

plt.title("PCA of Basketball Dataset")

plt.xlabel("Principal Component 1")

plt.ylabel("Principal Component 2")

plt.colorbar(label='Cluster')

plt.show()

# t-SNE visualization

tsne = TSNE(n\_components=2, perplexity=5, learning\_rate=100, random\_state=42)

X\_tsne = tsne.fit\_transform(X\_scaled)

plt.figure(figsize=(8, 6))

plt.scatter(X\_tsne[:, 0], X\_tsne[:, 1], c=df['Cluster'], cmap='viridis')

plt.title("t-SNE of Basketball Dataset")

plt.colorbar(label='Cluster')

plt.show()

# UMAP visualization

umap = UMAP(n\_components=2, random\_state=42)

X\_umap = umap.fit\_transform(X\_scaled)

plt.figure(figsize=(8, 6))

plt.scatter(X\_umap[:, 0], X\_umap[:, 1], c=df['Cluster'], cmap='viridis')

plt.title("UMAP of Basketball Dataset")

plt.colorbar(label='Cluster')

plt.show()

# Examine the cluster centers (centroids)

print("Cluster centers (in original feature space):")

print(kmeans.cluster\_centers\_)

# Check the distribution of clusters

print(df['Cluster'].value\_counts())