

PCA Algo

```
python Copy code

# =====
# Foundations of Data Science Project
# Dataset: Cleaned_Matches_Dataset.csv
# Topic: PCA (Principal Component Analysis)
# =====

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# =====
# 1. Load Dataset
# =====
df = pd.read_csv("cleaned_Matches_Dataset.csv")
```

```
# Drop unwanted index column if present
if "Unnamed: 0" in df.columns:
    df = df.drop(columns=["Unnamed: 0"])

print("Shape of dataset:", df.shape)
```

 Copy code

```
# =====
# 2. Prepare Numeric Data
# =====
num_cols = ["result_margin", "target_runs", "target_overs"]
df_num = df[num_cols].fillna(0)

# Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df_num)
```

```
# =====
# 3. PCA (Manual Implementation using NumPy)
# =====
```

 Copy code

```
print("\n== PCA (Manual Implementation) ==")

# Step 1: Compute covariance matrix
cov_matrix = np.cov(X_scaled.T)

# Step 2: Eigen decomposition
eigen_values, eigen_vectors = np.linalg.eig(cov_matrix)

# Step 3: Sort eigenvalues & eigenvectors
idx = np.argsort(eigen_values)[::-1]
eigen_values = eigen_values[idx]
eigen_vectors = eigen_vectors[:, idx]

# Step 4: Project data onto first 2 principal components
X_pca_manual = X_scaled.dot(eigen_vectors[:, :2])

print("Eigenvalues:", eigen_values)
print("First 2 Principal Components (manual):\n", X_pca_manual[:5])
```

[Copy code](#)

```
# Plot manual PCA
plt.scatter(X_pca_manual[:,0], X_pca_manual[:,1], c='blue')
plt.title("PCA Projection (Manual)")
plt.xlabel("PC1")
plt.ylabel("PC2")
plt.show()

# =====
# 4. PCA using sklearn
# =====
print("\n== PCA (sklearn) ==")

pca = PCA(n_components=2)
X_pca_sklearn = pca.fit_transform(X_scaled)

print("Explained Variance Ratio:", pca.explained_variance_ratio_)
print("First 2 Principal Components (sklearn):\n", X_pca_sklearn[:5])

# Plot sklearn PCA
plt.scatter(X_pca_sklearn[:,0], X_pca_sklearn[:,1], c='green')
plt.title("PCA Projection (sklearn)")
plt.xlabel("PC1")
plt.ylabel("PC2")
plt.show()
```

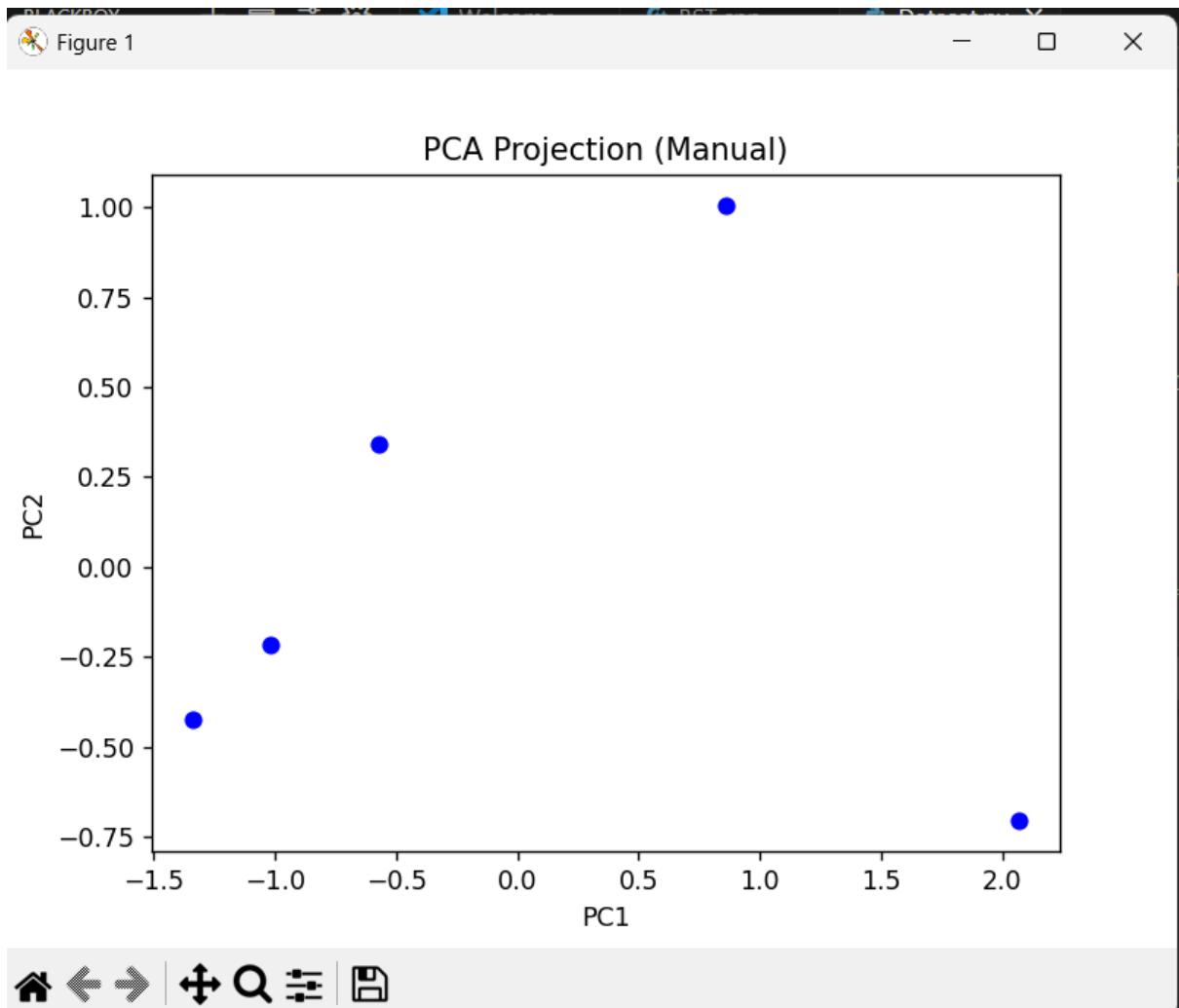




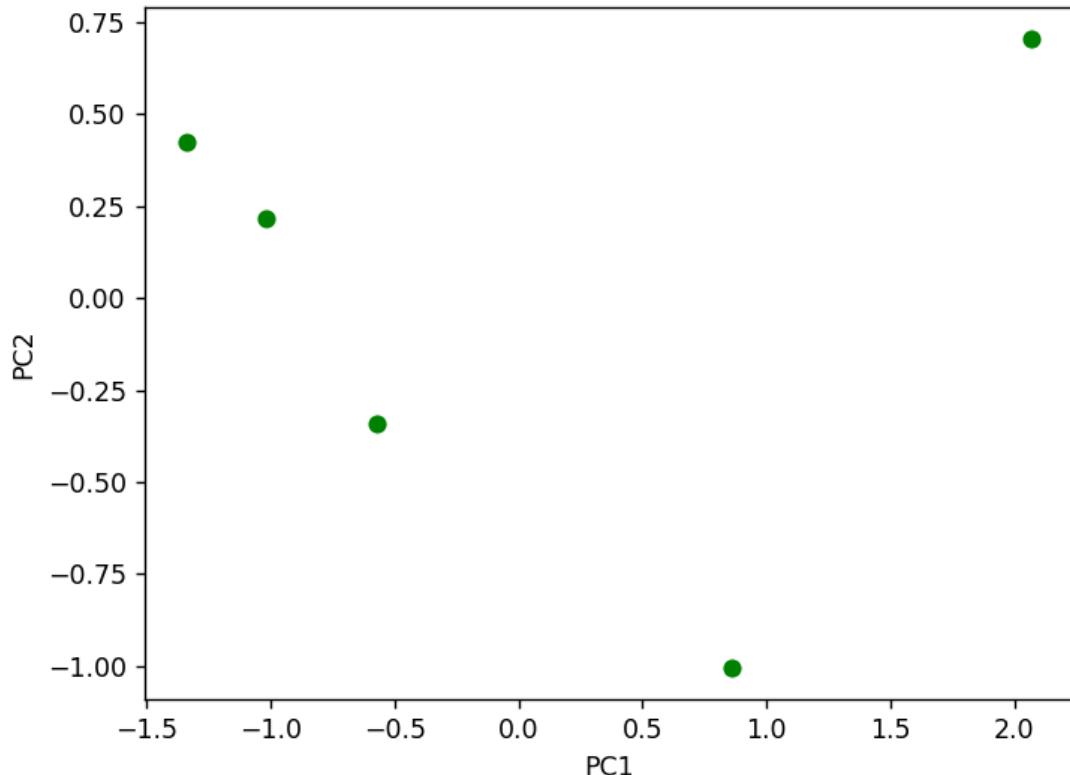
Figure 1

-

□

×

PCA Projection (sklearn)



|



|



|



|



|



(x, y) = (1.693, -0.561)