Laxmi Charitable Trust's
Sheth L.U.J College of Arts & Sir M.V. College Of Science & Commerce

RAM KUMAR SINGH

PRACTICAL NO.9

AIM:Principal Component Analysis (PCA)

Perform PCA on a dataset to reduce dimensionality.

Evaluate the explained variance and select the appropriate number of principal components.

Visualize the data in the reduced-dimensional space.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.impute import SimpleImputer
```

```
df = pd.read_csv("/content/energy_dataset.csv")

print("\nFirst 5 Rows of Dataset:")
print(df.head())
print("Shape:", df.shape)
```

```
                        time  generation biomass  \
0  2015-01-01 00:00:00+01:00               447.0
1  2015-01-01 01:00:00+01:00               449.0
2  2015-01-01 02:00:00+01:00               448.0
3  2015-01-01 03:00:00+01:00               438.0
4  2015-01-01 04:00:00+01:00               428.0

   generation fossil brown coal/lignite  generation fossil coal-derived gas  \
0                                 329.0                                 0.0
1                                 328.0                                 0.0
2                                 323.0                                 0.0
3                                 254.0                                 0.0
4                                 187.0                                 0.0

   generation fossil gas  generation fossil hard coal  generation fossil oil  \
0                 4844.0                       4821.0                  162.0
1                 5196.0                       4755.0                  158.0
```

```
   generation fossil oil shale  generation fossil peat  generation geothermal  \
0                         0.0                     0.0                    0.0
1                         0.0                     0.0                    0.0
2                         0.0                     0.0                    0.0
3                         0.0                     0.0                    0.0
4                         0.0                     0.0                    0.0

   ...  generation waste  generation wind offshore  generation wind onshore  \
0  ...             196.0                       0.0                   6378.0
1  ...             195.0                       0.0                   5890.0
2  ...             196.0                       0.0                   5461.0
3  ...             191.0                       0.0                   5238.0
4  ...             189.0                       0.0                   4935.0

   forecast solar day ahead  forecast wind offshore eday ahead  \
0                      17.0                                NaN
1                      16.0                                NaN
2                       8.0                                NaN
3                       2.0                                NaN
4                       9.0                                NaN

   forecast wind onshore day ahead  total load forecast  total load actual  \
0                           6436.0              26118.0            25385.0
1                           5856.0              24934.0            24382.0
2                           5454.0              23515.0            22734.0
3                           5151.0              22642.0            21286.0
4                           4861.0              21785.0            20264.0

   price day ahead  price actual
0            50.10         65.41
1            48.10         64.92
2            47.33         64.48
3            42.27         59.32
4            38.41         56.04

[5 rows x 29 columns]
Shape: (35064, 29)
```

```
numeric_cols = df.select_dtypes(include=[np.number]).columns.tolist()
print("\nNumeric Columns:", numeric_cols)

X = df[numeric_cols].copy()
```

```
Numeric Columns: ['generation biomass', 'generation fossil brown coal/lignite', 'genera
```

```
if "price actual" in X.columns:
    y = X["price actual"].copy()
    X = X.drop(columns=["price actual"])
    target_name = "price actual"
    print("Using 'price actual' as pseudo-target.")
else:
    y = pd.Series(np.zeros(len(X)), name="target")
    target_name = "target"
    print("Using dummy target.")
```

```
Using 'price actual' as pseudo-target.
```

```python
cols_all_nan = X.columns[X.isna().all()]
print("\nColumns completely NaN (removed):", list(cols_all_nan))

X = X.drop(columns=cols_all_nan)
print("Shape after removing NaN-only columns:", X.shape)
```

```
Columns completely NaN (removed): ['generation hydro pumped storage aggregated', 'forec
Shape after removing NaN-only columns: (35064, 25)
```

```python
imputer = SimpleImputer(strategy="mean")
X_imputed = pd.DataFrame(imputer.fit_transform(X), columns=X.columns)

# Also handle NaNs in target, if any
if y.isna().any():
    y = y.fillna(y.mean())

print("Remaining NaNs:", X_imputed.isna().sum().sum())
print("First 5 rows after imputation:")
print(X_imputed.head())
```

```
   generation fossil coal-derived gas  generation fossil gas  \
0                                 0.0                 4844.0   
1                                 0.0                 5196.0   
2                                 0.0                 4857.0   
3                                 0.0                 4314.0   
4                                 0.0                 4130.0   

   generation fossil hard coal  generation fossil oil  \
0                       4821.0                  162.0   
1                       4755.0                  158.0   
2                       4581.0                  157.0   
3                       4131.0                  160.0   
4                       3840.0                  156.0   

   generation fossil oil shale  generation fossil peat  generation geothermal  \
0                          0.0                     0.0                    0.0   
1                          0.0                     0.0                    0.0   
2                          0.0                     0.0                    0.0   
3                          0.0                     0.0                    0.0   
4                          0.0                     0.0                    0.0   

   generation hydro pumped storage consumption  ...  \
0                                        863.0  ...   
1                                        920.0  ...   
```

```
3                        75.0              50.0            191.0
4                        74.0              42.0            189.0

   generation wind offshore  generation wind onshore  \
0                       0.0                   6378.0
1                       0.0                   5890.0
2                       0.0                   5461.0
3                       0.0                   5238.0
4                       0.0                   4935.0

   forecast solar day ahead  forecast wind onshore day ahead  \
0                      17.0                           6436.0
1                      16.0                           5856.0
2                       8.0                           5454.0
3                       2.0                           5151.0
4                       9.0                           4861.0

   total load forecast  total load actual  price day ahead
0              26118.0            25385.0            50.10
1              24934.0            24382.0            48.10
2              23515.0            22734.0            47.33
3              22642.0            21286.0            42.27
4              21785.0            20264.0            38.41

[5 rows x 25 columns]
```

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_imputed)

print("\nStandardScaler Parameters:")
print(scaler.get_params())
```

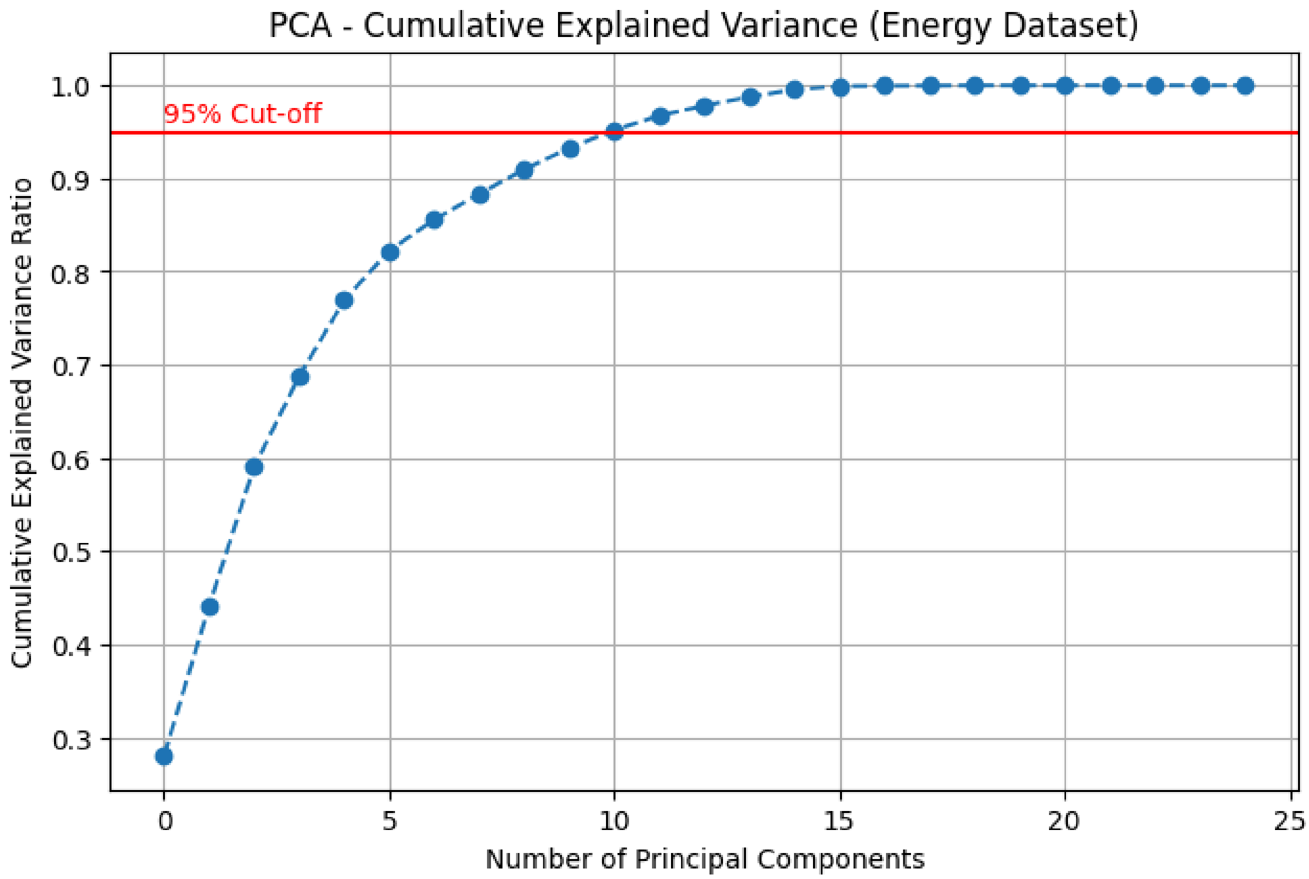
```
StandardScaler Parameters:
{'copy': True, 'with_mean': True, 'with_std': True}
```

```
pca_full = PCA()
pca_full.fit(X_scaled)

explained_variance_ratio = pca_full.explained_variance_ratio_
cumulative_variance = np.cumsum(explained_variance_ratio)

plt.figure(figsize=(8, 5))
plt.plot(cumulative_variance, marker='o', linestyle='--')
plt.axhline(y=0.95, linestyle='-', color='r')
plt.text(0, 0.96, '95% Cut-off', color='red')
plt.xlabel('Number of Principal Components')
plt.ylabel('Cumulative Explained Variance Ratio')
plt.title('PCA - Cumulative Explained Variance (Energy Dataset)')
plt.grid(True)
plt.show()

n_components = np.argmax(cumulative_variance >= 0.95) + 1
print(f"\nNumber of components for 95% variance: {n_components}")
```



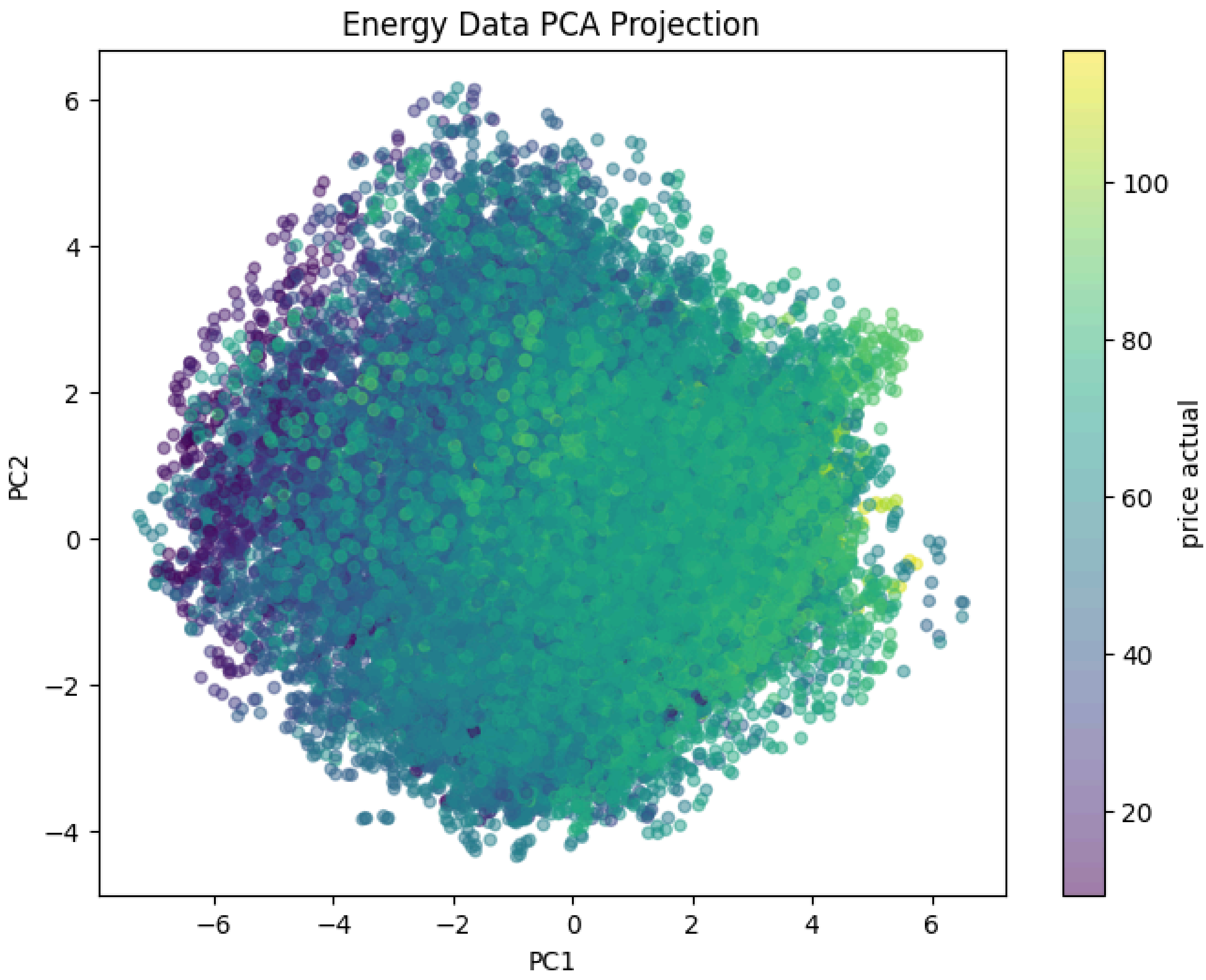

```
Number of components for 95% variance: 11
```

```
pca = PCA(n_components=n_components)
X_reduced = pca.fit_transform(X_scaled)

print("\nFinal PCA Parameters:")
print(pca.get_params())
```

```
Final PCA Parameters:
{'copy': True, 'iterated_power': 'auto', 'n_components': np.int64(11), 'n_oversamples':
```

```
plt.figure(figsize=(8, 6))
plt.scatter(X_reduced[:, 0], X_reduced[:, 1], c=y, cmap='viridis', s=20, alpha=0.5)
plt.colorbar(label=target_name)
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.title('Energy Data PCA Projection')
plt.show()
```




```
pca_cols = [f"PC{i+1}" for i in range(n_components)]
pca_df = pd.DataFrame(X_reduced, columns=pca_cols)
pca_df[target_name] = y.values

output_file = "/content/energy_pca_reduced.csv"
pca_df.to_csv(output_file, index=False)

print("\nFirst 5 rows of PCA reduced data:")
print(pca_df.head())
print("\nSaved to:", output_file)
```

```
First 5 rows of PCA reduced data:
        PC1       PC2       PC3       PC4       PC5       PC6       PC7  \
0 -1.939891 -0.803230  0.699363 -0.021090  0.369430  0.884504 -1.011903
1 -2.057097 -1.116139  0.845545  0.238332  0.443172  0.891167 -1.141480
2 -2.373668 -1.533295  0.908606  0.482030  0.434244  0.877217 -1.069578
3 -2.960248 -1.804342  0.940045  0.750306  0.321394  0.845386 -1.062811
4 -3.439316 -1.948464  0.951263  1.003525  0.372551  0.862805 -1.207829

        PC8       PC9      PC10      PC11  price actual
0 -0.882678 -1.921679 -0.832382 -0.884281         65.41
1 -0.863688 -1.850541 -0.963886 -0.791646         64.92
2 -0.859786 -1.585825 -0.897589 -0.875170         64.48
3 -0.716736 -1.185771 -0.887189 -0.939449         59.32
4 -0.789941 -0.925156 -0.895755 -0.876701         56.04
```

```
Saved to: /content/energy_pca_reduced.csv
```