

```
[1]: pip install pandas numpy scikit-learn matplotlib seaborn
Requirement already satisfied: pandas in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (2.3.0)
Requirement already satisfied: numpy in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (2.3.0)
Requirement already satisfied: scikit-learn in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (1.7.0)
Requirement already satisfied: matplotlib in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (3.10.3)
Requirement already satisfied: seaborn in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (0.13.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from pandas) (2.25.2)
Requirement already satisfied: pytz>=2020.1 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from pandas) (2025.2)
Requirement already satisfied: scipy>=1.8.0 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from scikit-learn) (1.15.3)
Requirement already satisfied: joblib>=1.2.0 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from scikit-learn) (1.5.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: contourpy>=1.0.1 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from matplotlib) (4.58.4)
Requirement already satisfied: packaging>=20.0 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from matplotlib) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from matplotlib) (3.2.3)
Requirement already satisfied: six>=1.5 in /Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[2]: import glob
```

```
[3]: import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[5]: import pandas as pd
```

```
[6]: # Match only files that start with "CMX1_AL" and end with ".csv"
file_list = glob.glob("/Users/admin/Documents/AMS/clean_datasets/CMX1_AL*.csv")

# Optional: Print matched files
print("Matched CSV files:")
for file in file_list:
    print(file)

# Combine them into a single DataFrame
df = pd.concat([pd.read_csv(file) for file in file_list], ignore_index=True)

# Show result
print(df.shape)
df.head()
```

```
Matched CSV files:
/Users/admin/Documents/AMS/clean_datasets/CMX1_AL_CP1.csv
/Users/admin/Documents/AMS/clean_datasets/CMX1_AL_CP2.csv
(508488, 56)
```

	LOAD 1	LOAD 2	LOAD 3	LOAD 6	ENC_POS 1	ENC_POS 2	ENC_POS 3	ENC_POS 6	CTRL_DIFF2 1	CTRL_DIFF2 2	...	ENC1_POS 3	ENC1_POS
0	3.857422	3.320312	24.230957	3.619385	-335.15767	-234.31094	-401.311863	527761.371165	0.013478	0.000203	...	262.440758	527761.371165
1	3.857422	3.320312	24.230957	3.619385	-335.15213	-234.31158	-401.311453	527783.573847	0.013601	0.001036	...	262.440777	527783.573847
2	4.199219	3.985596	24.212646	3.607178	-335.14666	-234.31181	-401.311073	527805.777216	0.013794	0.001459	...	262.440806	527805.777216
3	4.199219	3.985596	24.212646	3.607178	-335.14170	-234.31153	-401.311273	527827.979212	0.014498	0.001371	...	262.440834	527827.979212
4	4.907227	4.229736	24.291992	3.601074	-335.13690	-234.31128	-401.311323	527850.179835	0.015361	0.001312	...	262.440806	527850.179835

5 rows × 56 columns

```
[7]: import glob
import pandas as pd

# Step 1: Get all files starting with 'CMX1_AL' and ending in '.csv'
file_list = glob.glob("/Users/admin/Documents/AMS/clean_datasets/CMX1_AL*.csv")

# Step 2: Print file names
print("Matched CSV files:")
for file in file_list:
    print(file)

# Step 3: Read each file into a separate DataFrame (optional but helpful for checking)
dfs = []
for file in file_list:
    df = pd.read_csv(file)
    print(f"\nRead file: {file}")
    print(f"Shape: {df.shape}")
    dfs.append(df)

# Step 4: Combine all into a single DataFrame
```

```

combined_df = pd.concat(dfs, ignore_index=True)

# Final check
print(f"\nTotal combined shape: {combined_df.shape}")

Matched CSV files:
/Users/admin/Documents/AMS/clean_datasets/CMX1_AL_CP1.csv
/Users/admin/Documents/AMS/clean_datasets/CMX1_AL_CP2.csv

Read file: /Users/admin/Documents/AMS/clean_datasets/CMX1_AL_CP1.csv
Shape: (433666, 56)

Read file: /Users/admin/Documents/AMS/clean_datasets/CMX1_AL_CP2.csv
Shape: (74822, 56)

Total combined shape: (508488, 56)

[12]: # Step 1: Read and combine the files
file_list = glob("Users/admin/Documents/AMS/clean_datasets/CMX1_AL*.csv")
dfs = [pd.read_csv(file) for file in file_list]
df = pd.concat(dfs, ignore_index=True)

# Step 2: Check and clean the data
print(df.info()) # Check for non-numeric columns or missing values

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 508488 entries, 0 to 508487
Data columns (total 56 columns):
 #   Column      Non-Null Count   Dtype  
 ---  --          --          --      
 0   LOAD|1      508488 non-null  float64 
 1   LOAD|2      508488 non-null  float64 
 2   LOAD|3      508488 non-null  float64 
 3   LOAD|6      508488 non-null  float64 
 4   ENC_POS|1   508488 non-null  float64 
 5   ENC_POS|2   508488 non-null  float64 
 6   ENC_POS|3   508488 non-null  float64 
 7   ENC_POS|6   508488 non-null  float64 
 8   CTRL_DIFF2|1 508488 non-null  float64 
 9   CTRL_DIFF2|2 508488 non-null  float64 
 10  CTRL_DIFF2|3 508488 non-null  float64 
 11  CTRL_DIFF2|6 508488 non-null  float64 
 12  TORQUE|1    508488 non-null  float64 
 13  TORQUE|2    508488 non-null  float64 
 14  TORQUE|3    508488 non-null  float64 
 15  TORQUE|6    508488 non-null  float64 
 16  DES_POS|1   508488 non-null  float64 
 17  DES_POS|2   508488 non-null  float64 
 18  DES_POS|3   508488 non-null  float64 
 19  DES_POS|6   508488 non-null  float64 
 20  CTRL_DIFF|1 508488 non-null  float64 
 21  CTRL_DIFF|2 508488 non-null  float64 
 22  CTRL_DIFF|3 508488 non-null  float64 
 23  CTRL_DIFF|6 508488 non-null  float64 
 24  CTRL_POS|1  508488 non-null  float64 
 25  CTRL_POS|2  508488 non-null  float64 
 26  CTRL_POS|3  508488 non-null  float64 
 27  CTRL_POS|6  508488 non-null  float64 
 28  VEL_FFW|1   508488 non-null  float64 
 29  VEL_FFW|2   508488 non-null  float64 
 30  VEL_FFW|3   508488 non-null  float64 
 31  VEL_FFW|6   508488 non-null  float64 
 32  CONT_DEV|1  508488 non-null  float64 
 33  CONT_DEV|2  508488 non-null  float64 
 34  CONT_DEV|3  508488 non-null  float64 
 35  CONT_DEV|6  508488 non-null  float64 
 36  CMD_SPEED|1 508488 non-null  float64 
 37  CMD_SPEED|2 508488 non-null  float64 
 38  CMD_SPEED|3 508488 non-null  float64 
 39  CMD_SPEED|6 508488 non-null  float64 
 40  TORQUE_FFW|1 508488 non-null  float64 
 41  TORQUE_FFW|2 508488 non-null  float64 
 42  TORQUE_FFW|3 508488 non-null  float64 
 43  TORQUE_FFW|6 508488 non-null  float64 
 44  ENC1_POS|1   508488 non-null  float64 
 45  ENC1_POS|2   508488 non-null  float64 
 46  ENC1_POS|3   508488 non-null  float64 
 47  ENC1_POS|6   508488 non-null  float64 
 48  ENC2_POS|1   508488 non-null  float64 
 49  ENC2_POS|2   508488 non-null  float64 
 50  ENC2_POS|3   508488 non-null  float64 
 51  ENC2_POS|6   508488 non-null  float64 
 52  CURRENT|1   508488 non-null  float64 
 53  CURRENT|2   508488 non-null  float64 
 54  CURRENT|3   508488 non-null  float64 
 55  CURRENT|6   508488 non-null  float64

dtypes: float64(56)
memory usage: 217.2 MB
None

[17]: # Step 6: (Optional) PCA-transformed DataFrame
pca_df = pd.DataFrame(pca_result, columns=[f'PC{i+1}' for i in range(pca_result.shape[1])])
print(pca_df.head())

          PC1        PC2        PC3        PC4        PC5        PC6        PC7 \
0 -0.586248  0.591217 -1.043815  1.421009  1.015342  0.637605 -0.298792
1 -0.647474  0.361379 -1.184273  0.973734  0.849235  0.680482 -0.310499
2 -0.687792  0.245844 -1.351410  0.699288  0.762038  0.626944 -0.341337
3 -0.700564  0.234058 -1.404067  0.582696  0.519399  0.708996 -0.231750
4 -0.734257  0.171871 -1.578804  0.380016  0.298085  0.734427 -0.116616

          PC8        PC9        PC10       ...        PC47        PC48 \
0  0.261996  0.083196 -0.143298  ...  2.866752e-09  8.107761e-11
1 -0.013234  0.316249 -0.048727  ... -6.429611e-09 -4.129454e-09
2 -0.269607  0.593161 -0.141380  ... -4.983088e-09 -7.841564e-09
3 -0.474337  0.852417 -0.066667  ... -4.522539e-09  2.057903e-09
4 -0.780256  1.307289 -0.135314  ... -4.436336e-09 -6.769707e-09

          PC49        PC50        PC51        PC52        PC53 \
0 -5.577756e-10 -6.677141e-10  1.024794e-24 -1.116897e-09 -1.129061e-09
1 -1.072786e-09  1.748658e-09 -1.138417e-24  1.379273e-09  1.344207e-09

```

```
2 8.248495e-11 7.852311e-10 -2.127840e-24 2.034034e-09 5.209366e-10
3 -7.855682e-10 1.167899e-09 1.946197e-25 -1.756460e-10 -5.138499e-10
4 1.848596e-10 7.127582e-10 -9.358792e-25 6.953181e-10 -1.000926e-09
```

```
PC54 PC55 PC56
0 1.579586e-09 5.896974e-10 -2.265390e-09
1 -1.642634e-09 -5.868560e-10 3.256788e-09
2 -2.373288e-09 -5.088174e-10 -1.177369e-09
3 1.032863e-09 1.489072e-09 4.635862e-09
4 7.175055e-11 1.504187e-09 2.116681e-09
```

[5 rows x 56 columns]

```
[20]: # Step 1: Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df_numeric) # Make sure df_numeric only has 56 numeric features

[21]: # Step 2: Apply PCA with n_components=15
pca = PCA(n_components=15)
pca_result = pca.fit_transform(scaled_data)

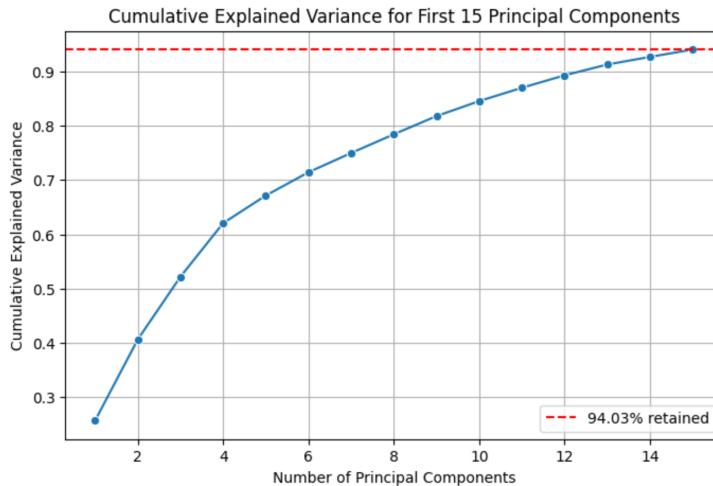
[22]: # Step 3: Explained variance
explained_var_ratio = pca.explained_variance_ratio_
cumulative_explained_var = np.cumsum(explained_var_ratio)

[23]: # Step 4: Print compression stats
original_dims = scaled_data.shape[1] # Should be 56
compressed_dims = 15
compression_ratio = compressed_dims / original_dims
variance_retained = cumulative_explained_var[-1]

print(f"Original dimensions: {original_dims}")
print(f"Reduced dimensions: {compressed_dims}")
print(f"Compression ratio: {compression_ratio:.2f} (i.e., {(100 - compression_ratio * 100):.0f}% compressed)")
print(f"Total variance retained: {variance_retained:.2%}")

Original dimensions: 56
Reduced dimensions: 15
Compression ratio: 0.27 (i.e., 73% compressed)
Total variance retained: 94.03%
```

```
[24]: # Step 5: Plot cumulative variance
plt.figure(figsize=(8, 5))
sns.lineplot(x=range(1, len(cumulative_explained_var)+1), y=cumulative_explained_var, marker='o')
plt.axhline(variance_retained, color='red', linestyle='--', label=f'{variance_retained:.2%} retained')
plt.title("Cumulative Explained Variance for First 15 Principal Components")
plt.xlabel("Number of Principal Components")
plt.ylabel("Cumulative Explained Variance")
plt.legend()
plt.grid(True)
plt.show()
```



```
[25]: #compression analysis
original_dims = scaled_data.shape[1]
compressed_dims = 15
compression_ratio = compressed_dims / original_dims
variance_retained = cumulative_explained_var[-1]

print(f"Original dimensions: {original_dims}")
print(f"Reduced dimensions: {compressed_dims}")
print(f"Compression ratio: {compression_ratio:.2f} ({(100 - compression_ratio * 100):.0f}% compressed)")
print(f"Total variance retained: {variance_retained:.2%}")

Original dimensions: 56
Reduced dimensions: 15
Compression ratio: 0.27 (73% compressed)
Total variance retained: 94.03%
```

[]:

回 ↑ ↓ ← → ⌂