

# UR3 CobotOps: Unsupervised Behaviors Clustering

XIU ZHEN HUANG  
06.10.2025


# Overview

- Brief Description of Dataset
- Data Cleaning
- Exploratory Data Analysis (EDA)
- Feature Engineering
- Models Detail
  - KMeans
  - GMM
  - Agglomerative
- Results and Analysis
- Conclusion
- Reference

The full project locates at the [git repository](#).

# Brief Description of Dataset

- UR3 CobotOps is a dataset that contains time-series data from a collaborative robot (cobot) performing various tasks.
- The dataset includes sensor readings, joint positions, and other relevant information that can be used for analysis and modeling.
- The dataset has **7409 instances** and **22 features**.
- I want to perform clustering analysis on this dataset to identify different operational behaviors of the cobot.



## UR3 CobotOps

Donated on 2/28/2024

The UR3 CobotOps Dataset is an essential collection of multi-dimensional time-series data from the UR3 cobot, offering insights into operational parameters and faults for machine learning in robotics and automation. It features electrical...

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate, Time-Series	Engineering	Classification, Regression, Clustering, Other
Feature Type	# Instances	# Features
Real, Categorical, Integer	7409	20

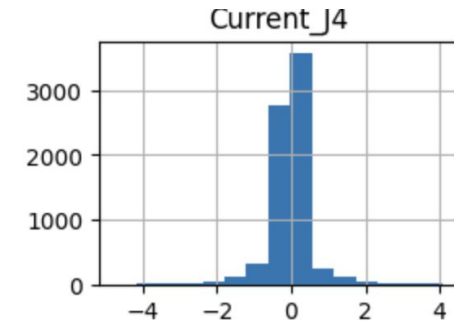
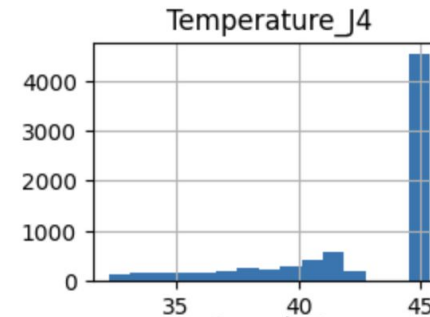
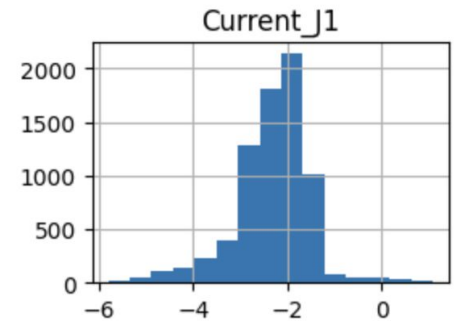
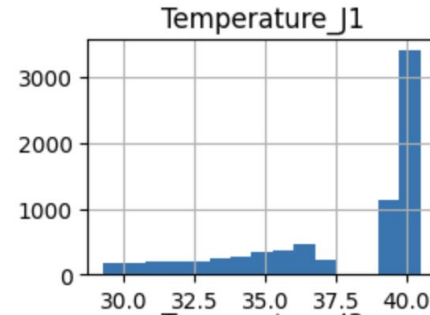
# Data Cleaning

Variable Name	Role	Type	Description	Units	Missing Values
Current_J0	Feature	Continuous			yes
Temperature_T0	Feature	Continuous			yes
Current_J1	Feature	Continuous			yes
Temperature_J1	Feature	Continuous			yes
Current_J2	Feature	Continuous			yes

- After reviewing the variable table and csv file, it shows that there are missing values and binary type variables.
- The following data cleaning processes are performed:
  - Drop unnecessary column - Num
  - Drop columns that all nan
  - Check nan value in cycle and drop the rows with nan cycle value
  - Replace space in column name
  - Drop Timestamp column because it is not useful for this clustering analysis
  - 'Robot\_ProtectiveStop' and 'grip\_lost' are binary value, it's better to convert it to int
  - Use mean instead of nan value in other columns which has continuous column type

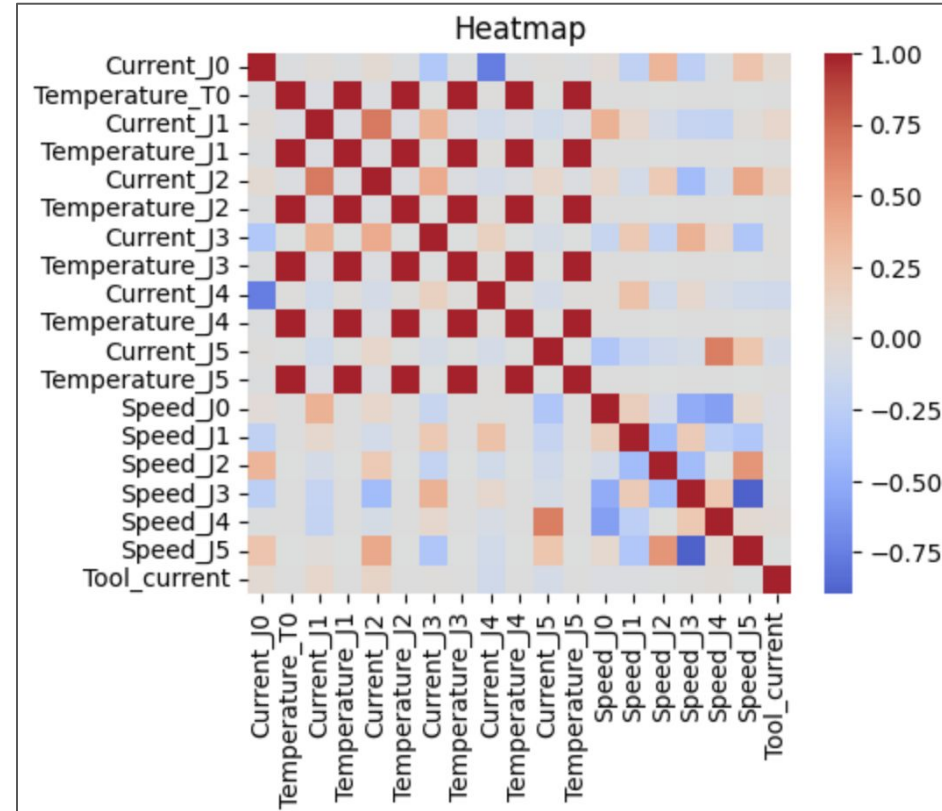
# Exploratory Data Analysis (EDA)

- Most of the sensor readings aren't normally distributed, which showing greatly skewness. Especially **temperature**, this indicates that the temperature data may be clustered into distinct groups based on the sensor readings.



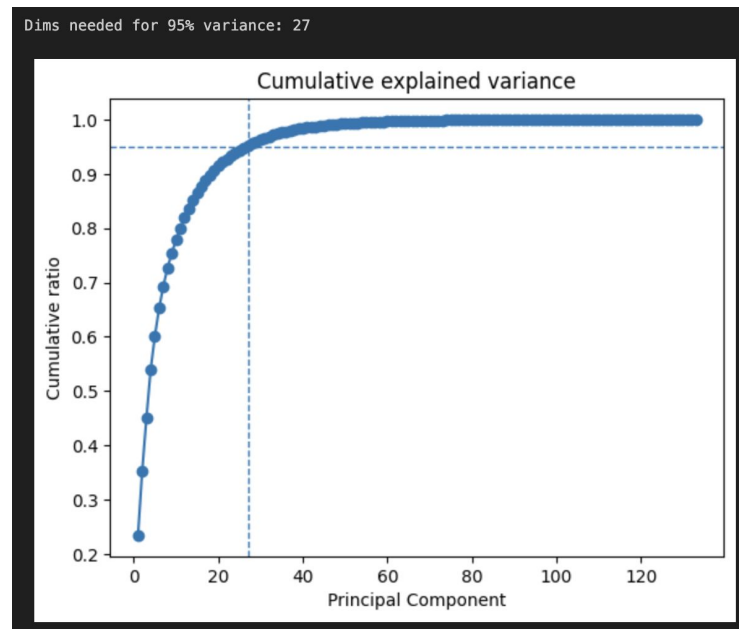
# Exploratory Data Analysis (EDA)

- There are strong correlations **temperature** between the joint positions
- The **speed** in the specific joint positions also strong correlated between each other.
- This suggests that these features may be related and could potentially be used together in clustering analysis.



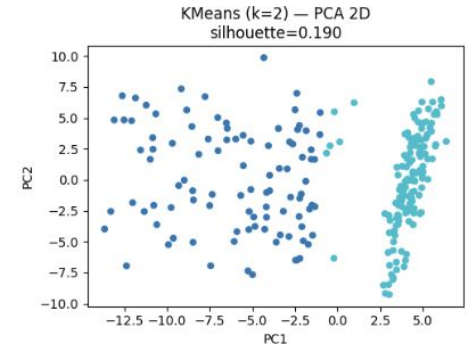
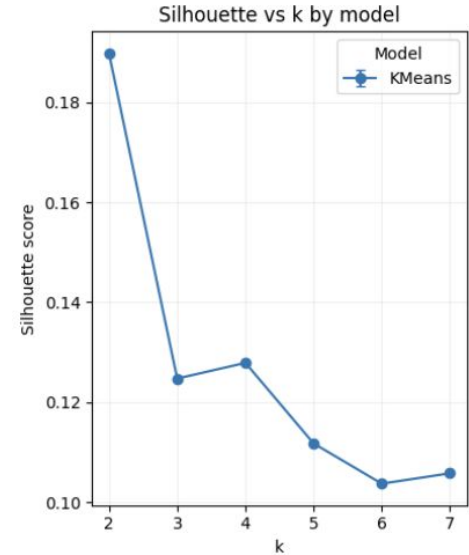
# Feature Engineering

- Create 133 cycle-level features (mean, std, min, max, madiff, p01, p99) for each feature
- Standardize the data (cycle-level features) using StandardScaler.
- Perform PCA and determine the number of components to retain based on explained variance ratio
- I decided to retain 27 components based on the explained variance ratio, which captures 95% of the variance in the data.



# Models Detail - KMeans

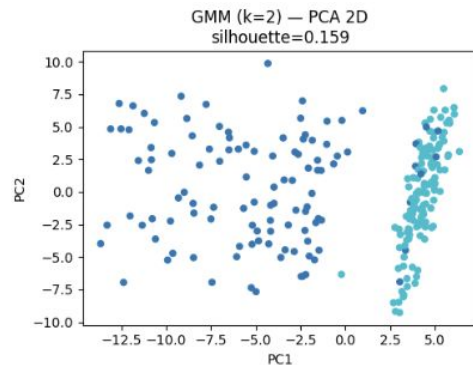
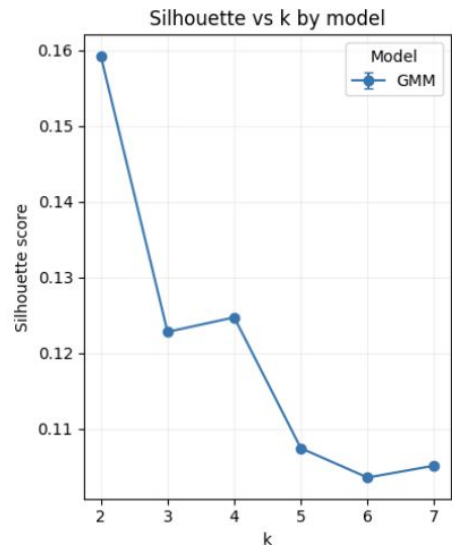
- The highest silhouette is at  $k=2$  ( $\sim 0.19$ ), which is low—so KMeans finds only a very mild cluster structure.
- It drops sharply after  $k=2$  and stays around 0.10–0.13 for  $k=3$ – $7$ , indicating no strong relation to increase  $k$  under KMeans.
- Separation mostly along PC1, which consist with the low overall silhouette and suggesting PC1 (likely temperature-driven) dominates the split.





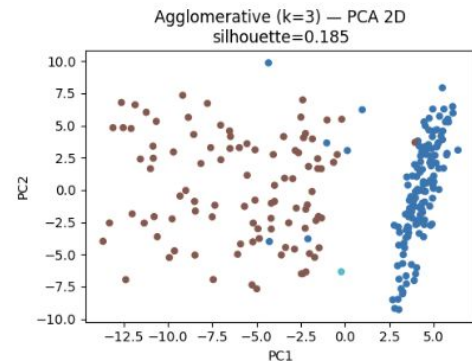
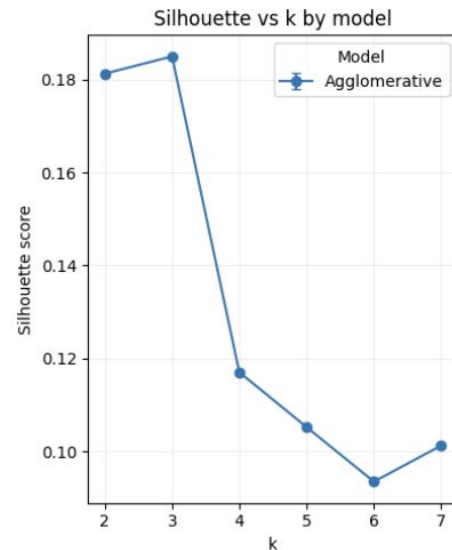
# Models Detail - GMM

- Best  $k = 2$ , weak separation. The highest silhouette is at  $k=2$  ( $\sim 0.16$ ), which is low—so GMM finds only a faint cluster structure.
- No benefit from more components. Silhouette drops for  $k \geq 3$  and stays around 0.105–0.12, indicating over-partitioning without uncovering new meaningful groups.



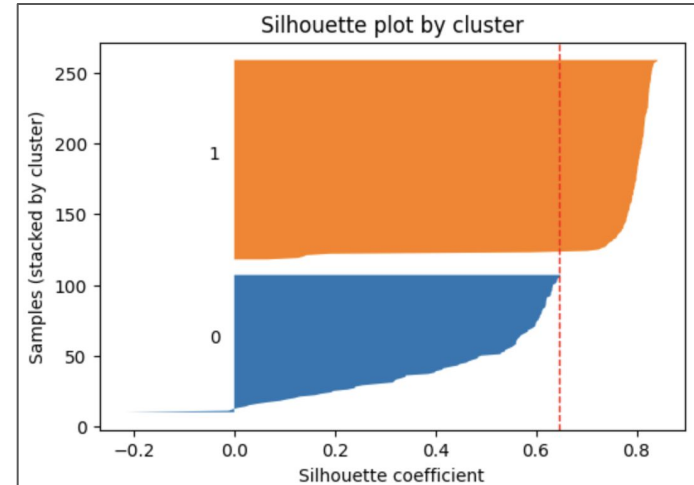
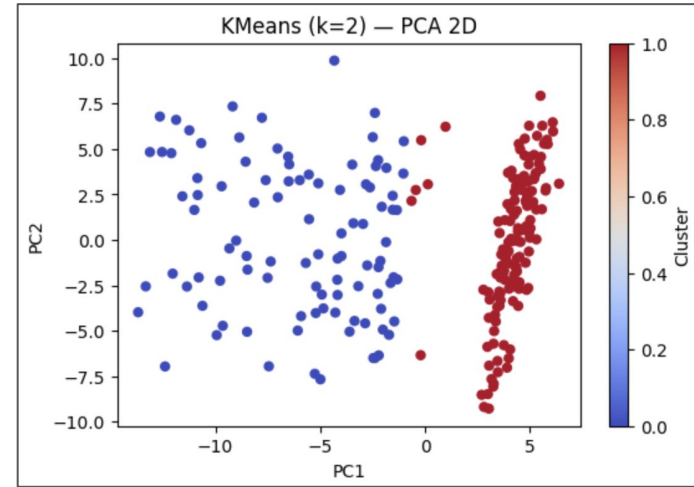
# Models Detail - Agglomerative

- Silhouette peaks around 0.185 at  $k=3$ . That small gain means the third cluster likely just carves the diffuse cloud rather than revealing a truly new, well-separated group.
- Silhouette falls to  $\sim 0.10$ – $0.12$  as  $k$  increases, indicating over-segmentation—the algorithm is splitting existing structure without improving separation.

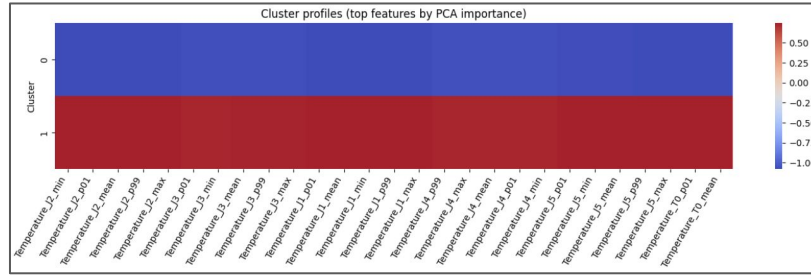


# Results and Analysis

- The KMeans ( $k = 2$ ) has the highest silhouette score when clustering.
- PCA 2D show a clear split along PC1
- It consistent with **temperature-driven variation that we observed earlier**; cluster 1 is much tighter/cleaner than cluster 0.
- Sample silhouette is high (red dashed line around  $\sim 0.65$ ), it means that clusters are reasonably well separated.
- Cluster 1 shows uniformly high silhouettes (well-defined); Cluster 0 has lower and more spread silhouettes, it indicates boundary or mixed-behavior points.



# Conclusion



model	k	silhouette
Agglomerative	3	0.184942
GMM	2	0.159262
KMeans	2	0.189810

- In this project, I compared KMeans, GMM, and Agglomerative clustering algorithms with different **cluster numbers (k=2 to 8)** using silhouette score for evaluation.
- Among all tested configurations, The best model was **KMeans(k=2)** achieved an overall silhouette = 0.189810, indicating it found something, but only a very mild cluster structure.
- We can see **the clusters split data as two type highly related about temperature**.
- Future work could involve exploring more data preprocessing techniques, such as **frequency features**, etc., to capture more complex patterns in the time-series data.

# Reference

- [UR3 CobotOps. UCI Machine Learning Repository](#)
- [Selecting the number of clusters with silhouette analysis on KMeans clustering](#)
- [AgglomerativeClustering](#)