yahoo\_dataset\_documentation

A comparison of blue lines

AI-generated content may be incorrect.A black screen with white text

AI-generated content may be incorrect.A computer screen shot of a black screen

AI-generated content may be incorrect.

- the reason the results look very similar is that the original Yahoo dataset was **already mostly scaled between 0 and ~0.8**. MinMaxScaler scales data to a 0-1 range, so the visual difference appears minimal.

-The main visible change is that the **highest peak now reaches exactly 1.0** instead of ~0.8

-This is important to ensure all values are precise.

- Many algorithms (like neural networks) expect exact 0-1 input ranges

Split the dataset into train\_data and test\_data

.reshape(-1,1)

-Reshape for MinMaxScaler in 2D (shapes: n\_rows, 1 columns)

.flatten()

- convert back into a 1D array and stored in the scaled\_yahoo variables

-make it easier to work with the scaled data as a single sequence

데이터 전처리

.ravel()

-값을 1D 배열로 추출

(numpy array)

데이터 전처리

A screen shot of a computer

AI-generated content may be incorrect.A computer code with many colored text

AI-generated content may be incorrect.A computer screen with text on it

AI-generated content may be incorrect.

-investigate the impact of having a smaller (3), medium (5), and larger (7) number of clusters on the time series segments.

-As we can see the values between small and medium parameter will suit the best for each segment length.

-So, n\_clusters chosen for KMeans’s paremeter is 4

Hyperparameter tuning & clustering

-Basic statistical features

-extract features from the dataset by different segments like mean, standard deviation, maximum value, minimum value, high values(spikes) and medium peaks.

-using Sliding Window/Overlapping Window technique to extract the features.

-This feature extraction part is crucial because clustering algorithms like KMeans work by grouping data points based on their numerical features.

-This allows the clustering algorithm to compare different segments based on these features and group together segments that exhibit similar patterns.

특징 추출

A screen shot of a computer program

AI-generated content may be incorrect.A computer screen shot of a program code

AI-generated content may be incorrect.

extract\_enhanced\_features(segment) function was included to address limitations in basic statistical feature extraction and improve clustering performance.

Visualization &data handling

-Apply **Principal Component Analysis (PCA)**

**Implementation:** Applied PCA to reduce feature dimensionality while retaining 95% of variance.

- **Visualization:** Enables 2D visualization of high-dimensional cluster relationships

**-Noise Reduction:** Filters out less informative features while preserving signal

-**Improved Clustering:** Removes redundant features that could confuse clustering algorithms

특징 추출

특징 추출

**1. Why Enhanced Features Are Necessary**

Basic statistical features (mean, std, max, etc.) provide a simple summary of time series segments but **fail to capture critical temporal dynamics** like:

* **Trends**: Directional movement (increasing/decreasing)
* **Periodicity**: Repeating patterns or cycles
* **Peak characteristics**: Precise spike detection
* **Frequency components**: Hidden oscillations

The enhanced features bridge this gap by incorporating domain-specific insights from time series analysis research

**Model Evaluation:**

**1.Silhouette Score**

**Implementation:** Calculated silhouette score as primary clustering quality metric

-**Rationale:**

* **Intrinsic evaluation:** Does not require ground truth labels[8](https://www.numberanalytics.com/blog/silhouette-score-clustering-evaluation)
* **Intuitive interpretation:** Quantifies clustering quality in easily understood terms
* **Widely accepted:** Standard metric in clustering literature

**2. Confidence Scoring**

**Implementation:** Custom confidence metric based on distance to cluster centers:

confidence = 1 / (1 + distance\_to\_center)

**Rationale:**

* Provides interpretable measure of assignment certainty
* Higher scores indicate points closer to cluster centers
* Useful for identifying potential misclassifications or boundary cases

**Prediction Methodology**

**1.New Data Processing Pipeline**

**Implementation:**

1. Apply same preprocessing (MinMaxScaler with fitted parameters)
2. Extract identical feature set from new segments
3. Transform using fitted PCA model
4. Predict cluster assignment using trained K-means model

**Rationale:**

* **Consistency:** Ensures new data follows identical preprocessing pipeline
* **Generalization:** Leverages learned cluster structure for classification
* **Practical utility:** Enables real-time classification of new time series segments

**Model Interpretation**

**A screenshot of a graph

AI-generated content may be incorrect.**

**1. Training Data with Cluster Assignments (Top Left)**

* The colored backgrounds show how your time series has been segmented and assigned to different clusters.
* There are clear, contiguous regions where segments are grouped together, indicating the model is capturing distinct patterns or regimes in the data.
* Some clusters dominate larger portions, while others are more localized, suggesting the presence of both common and rare patterns.

**2. PCA Visualization of Clusters (Top Middle)**

* Each point represents a segment in reduced feature space (first two principal components).
* The clusters are well-separated, with minimal overlap.
* This separation indicates that your extracted features are effective at distinguishing different types of segments, and the clustering algorithm is grouping similar patterns together.

**3. Test Data with Predicted Clusters (Top Right)**

* The test data is also segmented and colored by predicted cluster.
* Most test segments are assigned to a single cluster (orange), suggesting that the test data is more homogeneous or that one cluster center is much closer to the test segments.
* This could mean the test data contains mostly one type of behavior, or the model is less sensitive to rare patterns in the test set.

**4. Prediction Confidence Scores (Bottom Left)**

* The confidence scores (based on the inverse distance to the cluster center) are steadily increasing.
* This suggests that as you move through the test segments, they are increasingly similar to their assigned cluster center, indicating stable and confident predictions.

**5. Cluster Centers in PCA Space (Bottom Right)**

* The cluster centers are well-separated in the PCA-reduced feature space.
* Distinct, non-overlapping lines for each cluster center further confirm that the clusters are capturing different underlying segment characteristics.

**Overall Assessment**

* **Clustering is meaningful and stable:** Your model finds distinct, well-separated clusters in the data, and assigns both training and test segments with high confidence.
* **Features are effective:** The clear separation in PCA space and the stability of confidence scores suggest that your feature extraction and preprocessing steps are working well.
* **Test data is less diverse:** The dominance of one cluster in the test set may indicate a lack of variability in the test data, or that the model could be improved to better capture rare or anomalous patterns.

**Potential Next Steps**

* **Inspect cluster characteristics:** Analyze the average pattern in each cluster to interpret what kind of behaviors or events each cluster represents.
* **Check for overfitting:** If the test data is always assigned to one cluster, consider whether your training data is representative, or if the cluster count should be adjusted.
* **Enhance features:** If you want to capture more subtle or rare patterns, try extracting additional features or using a different segment size.

**Conclusion:**  
Your clustering model is robust, with clear and interpretable clusters. The visualization confirms that the model is effectively grouping similar time series segments and making stable predictions on new data. The next step would be to interpret what each cluster means in the context of your application.

**References**

1 **Neptune.ai Blog** - "Exploring Clustering Algorithms: Explanation and Use Cases." Available: <https://neptune.ai/blog/clustering-algorithms>

2 **Neptune.ai Blog** - "K-Means Clustering Explained." Available: <https://neptune.ai/blog/k-means-clustering>

3 **Number Analytics** - "A Deep Dive into Silhouette Score for Clustering Evaluation." Available: <https://www.numberanalytics.com/blog/silhouette-score-clustering-evaluation>

[4](https://neptune.ai/blog/clustering-algorithms) **Scikit-learn Documentation** - "MinMaxScaler." Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>

[5](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html) **Scikit-learn Documentation** - "Clustering." Available: <https://scikit-learn.org/stable/modules/clustering.html>

[6](https://help.llama.ai/release/native/demand/demand-topics/clstr_featureextract_clustering.htm) **Supply Chain Help Center** - "Feature extractions for clustering." Available: <https://help.llama.ai/release/native/demand/demand-topics/clstr_featureextract_clustering.htm>

[7](https://neptune.ai/blog/k-means-clustering) **Displayr** - "K-Means Clustering | Definition, Algorithm & How-To Guide." Available: <https://www.displayr.com/what-is-k-means-cluster-analysis/>

[8](https://www.numberanalytics.com/blog/silhouette-score-clustering-evaluation) **Displayr** - "Comprehensive Guide to Cluster Analysis: Methods, Uses & Tools." Available: <https://www.displayr.com/understanding-cluster-analysis-a-comprehensive-guide/>

[9](https://scikit-learn.org/0.15/modules/generated/sklearn.preprocessing.MinMaxScaler.html) **Wikipedia** - "Cluster analysis." Available: <https://en.wikipedia.org/wiki/Cluster_analysis>

[10](https://www.displayr.com/what-is-k-means-cluster-analysis/) **Number Analytics** - "7 Surprising Benefits of Dimensionality Reduction in Data Science." Available: <https://www.numberanalytics.com/blog/7-surprising-benefits-dimensionality-reduction-data-science>

11. Silhouette Score – “ADeep Dive into Silhouette Score for Clustering Evaluation**”:**

<https://www.numberanalytics.com/blog/silhouette-score-clustering-evaluation>

12. Dimensionality Reduction – “7 Surprising Benefits of Dimensionality Reduction in Data Science**”:** [**https://www.numberanalytics.com/blog/7-surprising-benefits-dimensionality-reduction-data-science**](https://www.numberanalytics.com/blog/7-surprising-benefits-dimensionality-reduction-data-science)

13. Sliding Window technique –“Sliding Window Technique — reduce the complexity of your algorithm”: <https://medium.com/@data-overload/sliding-window-technique-reduce-the-complexity-of-your-algorithm-5badb2cf432f>