Scaling Techniques Documentation

Customer Churn Analysis and Clustering

# Introduction

In this client churn project, our team conducted a series of data processing, including data scaling. By ensuring that numerical features are at the same scale through this processing, the contribution of each feature in the model during training is more equitable, resulting in more comprehensive results.

In this project, the StandardScaler in the sklearn.preprocessing module was used to standardize the data for subsequent clustering and classification processing.

# Scaling method used

Standardization (Z-score standardization). Z-score, also known as standard score, is used to evaluate the distance between a sample point and the population mean. The main application of z-score is to measure the standard deviation between the original data and the overall mean of the data.

**Technique:** StandardScaler() from scikit-learn

**Formula:** z = (x - μ) / σ

x is the original value, μ is the mean of the feature, σ is the standard deviation.

The effect of doing so is to convert all features into a standard normal distribution with a mean of 0 and a standard deviation of 1. In this project, our team scaled the tenure, Monthly Chambers, and other numerical features involved in modeling.

# Code

## Save the raw cleaned data as a CSV file for easy tracking or sharing in the future.

## 

from sklearn.preprocessing import StandardScaler

Import standardization tools for scaling features in Z-score format (with a mean of 0 and a standard deviation of 1).

scaler = StandardScaler() # Create a StandardScaler object

data\_scaled = scaler.fit\_transform(data) # Fit and transform the dataset

Perform standardized operations:

Converting all features in the data to the same scale is beneficial for subsequent modeling and clustering.

## 

Divide the training and testing sets:

test\_size=0.2： Use 20% of the data for testing and 80% for training;

random\_state=66： Set random seeds to ensure consistent partitioning each time.

## 

Display the partitioned data and confirm that the partitioning is correct



The result is shown in the figure.

## 

Afterwards, the divided data will be saved for future use

# Reason for scaling

Due to its dependence on Euclidean distance, K-Means clustering is extremely sensitive to data scale; If scaling is not performed, features with larger values will dominate in distance calculation. After scaling, the clustering effect of the model is better and the distribution between clusters is more balanced.

# Conclusion

After being processed by StandardScaler, the data is used for KMeans clustering and training/testing set partitioning, resulting in clearer clustering boundaries, more balanced feature contributions, and providing effective input for distance based algorithms.