1. Purpose of Feature Scaling

Feature scaling was applied during the data preparation phase to ensure that all numerical features contributed equally to machine learning algorithms. This step is particularly important for distance-based models such as K-Means clustering, where variables with larger numeric ranges can dominate the clustering process.

Without scaling, features such as tenure or monthly charges could disproportionately influence the distance calculations, leading to biased cluster formation.

2. Scaling Method Used

The project applied **Standardization (StandardScaler)** from the Scikit-learn library.

Standardization transforms each feature so that:

- Mean ≈ 0
- Standard deviation ≈ 1

The transformation formula is:

$$X_{scaled} = \frac{X - \mu}{\sigma}$$

Where:

- X= original feature value
- μ= feature mean
- σ= feature standard deviation

This method was selected because it performs well when features have different units or scales and is widely used in clustering and neural network models.

3. Features Scaled

All input features were scaled after categorical encoding. The target variable (Churn) was excluded from scaling because it represents the classification outcome rather than an input feature.

Examples of scaled features include:

- tenure
- MonthlyCharges
- Service-related encoded variables
- Contract-related encoded variables

4. Implementation in Python

The following code snippet shows how scaling was implemented:

```
from sklearn.preprocessing import StandardScaler

# Separate features and target
X = df_clean.drop("Churn", axis=1)
y = df_clean["Churn"]

# Initialize scaler
scaler = StandardScaler()

# Fit scaler on training data and transform
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Key implementation details:

- The scaler was fitted only on the training dataset.
- The same transformation was applied to the test dataset.
- This approach prevents data leakage and ensures reliable model evaluation.

5. Benefits of Scaling in This Project

Applying feature scaling provided several advantages:

- Improved clustering performance and stability.
- Balanced feature contribution during distance calculations.
- Faster convergence during model training.
- Better compatibility with future predictive modeling (ANN).

6. Outcome

After scaling:

- All features shared a comparable scale.
- The dataset became suitable for K-Means clustering.
- Model performance and segmentation quality improved.

Feature scaling therefore served as a critical preprocessing step in ensuring accurate and reliable analytical results.