Customer Journey Analysis Using Clustering and Dimensionality Reduction

Enhancing User Experience

Phase 2: Data Preprocessing and Model Design

2.1 Overview of Data Preprocessing

After completing the initial data exploration, this phase focuses on preparing the dataset for analyzing customer journeys using clustering and dimensionality reduction. This involves cleaning, transforming, and scaling the data to ensure it is suitable for training the deep autoencoder model. The primary goal of this phase is to handle missing values, outliers, and data inconsistencies, and to apply appropriate transformations such as feature scaling, encoding, and dimensionality reduction to uncover actionable insights about user behavior. The focus is to enable a seamless user experience by identifying behavioral patterns and improving decision-making.

2.2 Data Cleaning: Handling Missing Values, Outliers, and Inconsistencies

Cleaning the dataset is a critical step to ensure that the input data is accurate and ready for modeling. In this phase, we address the following issues:

- **Missing Values :** Missing data were identified using statistical methods such as descriptive statistics (mean, median) and visualization techniques like heatmaps. The following strategies were employed:
    - **Numerical Features**: Missing values were imputed using the mean (for approximately normal data) or median (for skewed data) to maintain data integrity and prevent downstream bias in clustering.
    - **Categorical Features**: Missing categorical values were imputed using the mode (the most frequent value) to ensure consistency across customer segments and maintain the categorical distribution.
    - **Advanced Techniques:** For datasets with complex missing data patterns, iterative imputation methods like K-nearest neighbors (KNN) imputation were applied to improve accuracy.

- **Outliers**: Outliers, detected using visualization methods like boxplots and statistical methods like Z-score, were handled by:
    - **Capped**: Limiting extreme values to a defined threshold (winsorization) to reduce their impact while retaining the majority of the data points.
    - **Removed**: Removing records with extreme outliers that significantly deviated from the distribution to avoid skewed clustering results, especially in highly sensitive metrics like conversion rates.
    - **Robust Scaling:** In scenarios where outliers were not removed, robust scaling methods (e.g., Median Absolute Deviation) were applied to minimize their influence on the model.

- **Inconsistencies**: Duplicate rows were identified and removed to avoid over representation of specific customer behaviors. Contradictory entries (e.g., conflicting demographic or behavioral information) were flagged for manual review or corrected to improve data reliability.

2.3 Feature Scaling and Normalization

Scaling ensures that features are comparable in magnitude, which is critical for clustering algorithms and autoencoder models. This step prevents features with larger ranges from dominating the learning process.

- **Standardization**: Applied to most numerical features using Z-score normalization (mean = 0, standard deviation = 1) to align the scales of features like time spent on-site and transaction value.
- **Normalization**: Skewed features, such as session durations or purchase frequency, were normalized using Min-Max scaling to range [0, 1].
- **Categorical Features**: Categorical variables were encoded using One-Hot Encoding, transforming them into binary columns to ensure the model correctly interprets customer attributes.

2.4 Feature Transformation and Dimensionality Reduction

Transforming features helps improve the performance of the deep learning model by reducing noise or irrelevant information and highlighting important patterns. This phase also includes applying dimensionality reduction techniques to handle high-dimensional data.

- **Encoding Categorical Variables**: One-Hot Encoding was used to convert categorical variables (e.g., device type, user location) into binary representations, avoiding unintended ordinal relationships.
- **Dimensionality Reduction**: Given the high number of features in the dataset, it was important to reduce the dimensionality to speed up the training process and prevent overfitting. Several dimensionality reduction techniques were considered:
    - **Principal Component Analysis (PCA)**: Reduced dimensionality while retaining maximum variance. For example, behavioral data such as clickstreams and navigation patterns were condensed into fewer principal components.
    - **Feature Selection**: Techniques like Variance Thresholding removed redundant features with low variance, ensuring only relevant aspects of customer journeys were considered.
    - **t-SNE or UMAP**: For visualization purposes, additional techniques such as t-distributed Stochastic Neighbor Embedding (t-SNE) or Uniform Manifold Approximation and Projection (UMAP) were applied to better understand customer behavior clusters in lower-dimensional spaces.

2.5 Autoencoder Model Design

With the data cleaned and transformed, we now turn to the model design. The focus in this project is on using an **autoencoder** for deep clustering. Autoencoders are unsupervised neural networks that learn to represent input data in a compressed latent space. The architecture of the autoencoder was designed as follows:

- **Encoder Architecture**: The encoder takes the preprocessed and transformed data as input and compresses it into a latent feature space. The encoder has the following layers:
  - An input layer that takes the feature vector.
  - Dense layers with progressively fewer neurons (e.g., 64, 32, 8) to compress the data into latent space, capturing essential customer journey patterns.
- **Decoder Architecture**:
  - Mirrors the encoder, expanding latent features back to original dimensions.
  - Output layer uses sigmoid activation to constrain reconstructed values within [0, 1].
- **Loss Function**: Mean Squared Error (MSE) minimized reconstruction errors to ensure accurate representation of customer journeys.
- **Optimizer**: Adam optimizer provided efficient gradient optimization for the learning process.

2.6 Model Training and Validation

The autoencoder was trained on a split dataset:

- **Training and Validation**: Performance was monitored to ensure generalization and avoid overfitting.

- **Hyperparameter Tuning**: Adjusted learning rate and batch size for optimal results.

After training, the encoder extracted latent features representing compact customer journey insights for clustering. Additionally, early stopping and dropout were implemented to prevent overfitting during training.

2.7 Clustering for User Segmentation

Using the latent features extracted by the autoencoder, clustering techniques such as K-means, hierarchical clustering, or DBSCAN were applied to group customers based on behavioral patterns. These clusters revealed segments with similar behaviors, such as frequent buyers, infrequent users, or users requiring personalized recommendations. Further analysis of these clusters helped pinpoint areas for improving user experience by identifying high-value users or churn risks.

2.8 Conclusion of Phase 2

This phase prepared the dataset by cleaning, handling missing values and outliers, and transforming features. Scaling and encoding ensured compatibility with the autoencoder. Dimensionality reduction improved efficiency by removing noise and redundancy. The trained model produced latent features ready for clustering, setting a solid foundation for identifying patterns and enhancing user experience through targeted strategies. The insights derived through clustering have the potential to revolutionize decision-making, ensuring a user-centric approach to business outcomes. Real-time adaptability and CLV integration further expand the practical applications of these findings.