CIA-2 Data Mining Project Documentation

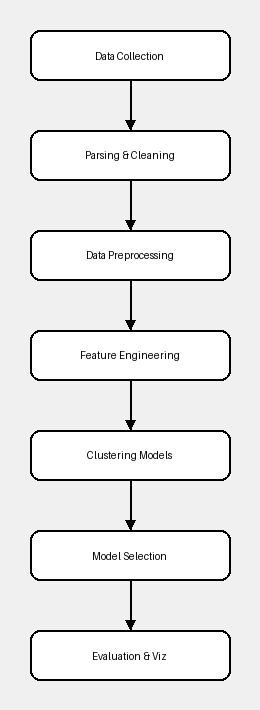
# 1. Analysis of Challenges

* **Unstructured Text Files**:
  + Transactional data (Bills\_1.txt) was unstructured and needed parsing via regular expressions.
* **Data Inconsistencies:**
  + Duplicate rows, missing values, and inconsistent formats in both customer and transaction datasets required preprocessing.
* **Mixed Data Types:**
  + A combination of numerical, categorical, and textual data (customer bio) necessitated tailored preprocessing steps for each type.
* **Textual Noise in Bios:** 
  + Emojis, slang, and inconsistent capitalization in bios were cleaned using custom functions and natural language tools.
* **High-Dimensionality:** 
  + Post-TF-IDF and one-hot encoding led to high-dimensional features, tackled using PCA before clustering.
* **Cluster Validation:** 
  + Evaluating cluster quality required careful use of Silhouette Scores and visual inspections through PCA.

# 2. Application Selection

* The chosen application is:  
  Customer Segmentation for Intelligent Targeted Marketing
* Using clustering techniques, customers are grouped based on transaction behavior and demographic patterns to:
* Enhance personalization
* Increase customer retention
* Improve inventory and marketing efficiency

# 3. Architecture Diagram



# Basic Steps involved:

* Parse the bills.txt and extract the features using regex and append it records in a dataframe using pandas
* Join this newly created dataset with the customer dataset
* Segregate the merged dataset into:
  + Transactional dataset (without bio)
  + Bio dataset (with bio and customer ID)
* Transactional dataset processing:
  + We do the basic preprocessing (Scaling, encoding)
  + Here we find patterns in purchases using apriori algorithm and add it as features of the customers
* Bio dataset processing:
  + After carrying out basic text processing:
    - Lower casing
    - Handling emojis
    - Stopwords handling
  + vectorize bios using TF-IDF
  + Use combined vectors for enhanced segmentation or prediction.
* Visualization and Plot:
  + Other than basic clustering plots:
    - Top items
    - Clusters
    - hourly trends
    - Cluster profiles
    - item affinities
  + These could be future works / extension of the actual project

# 4. Module Description

|  |  |  |
| --- | --- | --- |
| Module | Description | Techniques Used |
| Data Parser | Parses raw bills text file into structured DataFrame | Regex, String Manipulation |
| Preprocessing | Cleans data, encodes categories, scales numerics, prepares bios | StandardScaler, OrdinalEncoder, TF-IDF |
| Feature Engineering | Generates RFM and time-based customer features | Aggregation, DateTime processing |
| Text Cleaning | Removes emojis, normalizes bios, applies spell correction | Regex, emoji lib, difflib |
| PCA Reduction | Applies Principal Component Analysis to reduce dimensionality | PCA from sklearn |
| Clustering Models | Applies multiple clustering algorithms, stores cluster labels | KMeans, DBSCAN, Agglomerative, GMM |
| Evaluation & Plotting | Computes Silhouette Score and visualizes 3D clusters | Matplotlib, sklearn.metrics |

# 5. Data Selection and Preprocessing

Data Source: Two files—customers\_1.csv and Bills\_1.txt

NOTE: since this a small scale project, we assumed that we’ll be taking inputs as:

* Bills as raw text (all bills in a single text file)
* Customer database having credentials and their Bio (assumed it was collected from the store admins)
  + As for their preparation we used prompted and generated it using AI

**Cleaning Steps:**  
 • Removed duplicate transactions and customers  
 • Normalized column names

**Feature Extraction:**  
 • RFM (Recency, Frequency, Monetary) values  
 • Bio-based features using TF-IDF vectorization

**Text Preprocessing:**  
 • Lowercasing, emoji removal, basic spell correction

**Encoding and Scaling:**

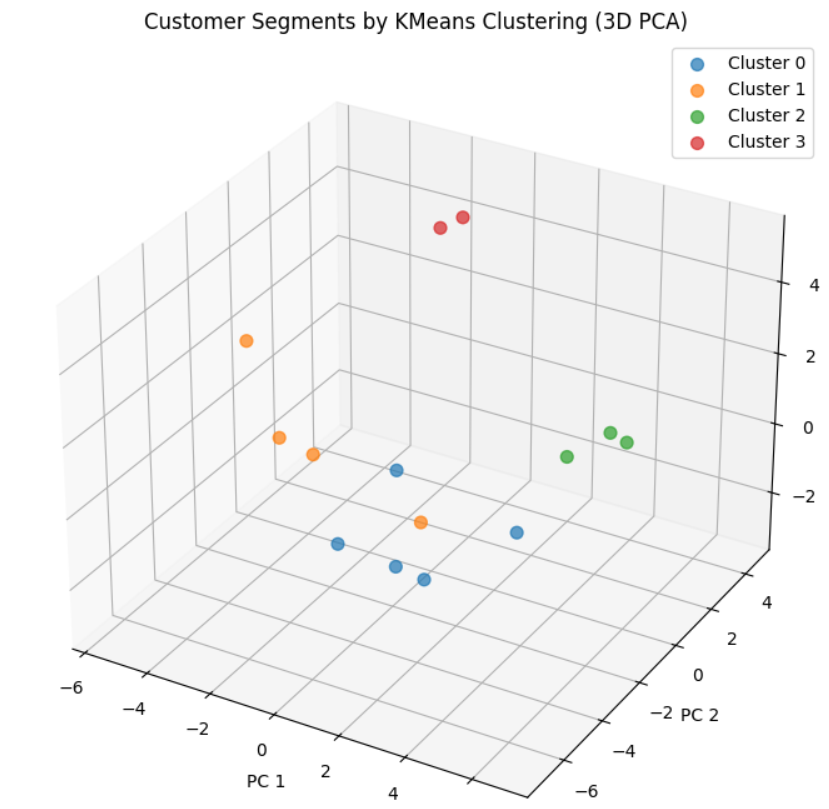
* Categorical data encoded with OrdinalEncoder
  + Encoding with OneHotEncoder caused curse of dimensionality and reduction in silhouette scores
* Numerical data scaled using StandardScaler

**Dimensionality Reduction:**  
 • PCA retained 95% variance to aid clustering

# 6. Performance Evaluation

* Multiple clustering models were evaluated:
* Best Performing Model: KMeans
* **Visualization**: 3D PCA-based cluster visualization clearly showed distinct clusters

**FINAL CLUSTER:**



# 7. Conclusion and Future Work

* The lack of data variety and lack of amount of data constraints the application to fully exploit the extent of this architecture’s abilities
* An actual dataset from real world from, say an e-commerce site would provide far better showcase of this project’s abilities
* This simple set up can be extended to be used in recommendation systems, accurate customer and purchase analysis, etc