



Low Level Design Document

Introduction

This Low Level Design (LLD) document details the implementation plan for **ClusterVision - Customer Segmentation Visualizer**. The project is a Streamlit-based tool for unsupervised K-Means clustering on retail customer data, providing interactive cluster visualization, evaluation (Elbow/Silhouette), and exportable results.

1. System Components

Component	Description / Responsibility
Data Loader	Loads and preprocesses customer data (CSV upload)
Clustering Engine	Performs K-Means clustering, computes metrics
Visualization Module	Renders cluster plots, Elbow, and Silhouette graphs
Streamlit UI	User interface for parameter input, visualization, export
Exporter	Exports clustered data/results as CSV

2. Class/Interface Overview

Class/Module	Key Methods / Attributes	Relationships
DataLoader	<code>load_data()</code> , <code>preprocess()</code>	Used by ClusteringEngine
ClusteringEngine	<code>fit()</code> , <code>predict()</code> , <code>compute_metrics()</code>	Uses DataLoader, Exporter
Visualizer	<code>plot_clusters()</code> , <code>plot_elbow()</code> , <code>plot_silhouette()</code>	Used by Streamlit UI
Exporter	<code>export_csv()</code>	Used by Streamlit UI

Key Methods Example:

```
class ClusteringEngine:
    def fit(self, data, n_clusters): ...
    def predict(self, data): ...
    def compute_metrics(self, data, max_k): ...
```

3. Data Structure Overview



Data Model	Fields / Structure
CustomerDataFrame	Pandas DataFrame: customer_id, features...
ClusterResult	DataFrame: customer_id, features..., cluster_label
MetricsResult	Dict: {'elbow': [inertia], 'silhouette': [score]}

4. Algorithms / Logic

K-Means Clustering & Evaluation Flow:

```
# Pseudocode
data = DataLoader.load_data(file)
data = DataLoader.preprocess(data)
for k in range(2, max_k):
    model = KMeans(n_clusters=k).fit(data)
    inertia_list.append(model.inertia_)
    silhouette_list.append(silhouette_score(data, model.labels_))
best_k = user_selected_k
final_model = KMeans(n_clusters=best_k).fit(data)
labels = final_model.labels_
Visualizer.plot_clusters(data, labels)
Exporter.export_csv(data, labels)
```

5. Error Handling

Scenario	Handling Approach
Invalid/empty file upload	Show Streamlit error message, abort processing
Non-numeric data in features	Skip/notify user, request correction
K-Means convergence failure	Show error, suggest parameter adjustment
Export failure	Show error, allow retry
Visualization rendering error	Show error, log details

End of Document