# Semantic Document Search and Clustering System

## Problem Statement

Traditional file search relies on keyword matching and filename recall. This approach fails when users do not remember exact terms, when documents are semantically related but use different wording, and when directories accumulate unstructured content. The objective of this project is to build a local semantic document indexing and retrieval system that extracts text, generates embeddings, enables semantic search, detects near-duplicates, clusters documents by topic, and evaluates clustering quality.

## Core Functional Requirements

1. Document Indexing (Batch Mode): Extract text from PDF, TXT, and DOCX files. Generate embeddings using a local model. Store file path, text, embedding, and file hash persistently. 2. Semantic Search: Accept natural language queries, compute query embeddings, and return top-k similar documents using cosine similarity. 3. Near-Duplicate Detection: Detect exact duplicates using hashing and near-duplicates using embedding similarity thresholds. 4. Document Clustering: Apply at least two unsupervised clustering methods (e.g., K-means, Agglomerative). Compare results and compute silhouette scores. 5. Cluster Interpretation: Extract representative keywords or summaries for each cluster.

## Technical Stack

Language: Python Suggested Libraries: sentence-transformers, scikit-learn, numpy, pandas, pdfplumber, python-docx, sqlite3 Vector indexing tools such as FAISS are optional.

## Deliverables

1. Working CLI-based system (indexing, search, duplicate detection, clustering). 2. Technical report (4–6 pages) explaining embedding strategy, similarity metric, clustering comparison, evaluation, limitations, and performance discussion. 3. Demonstration using a real-world dataset.

## Learning Objectives

Students will understand embedding representations, cosine similarity in high-dimensional space, unsupervised clustering techniques, evaluation metrics, and practical NLP pipeline design.