Low Level Documentation

Social Media Community Using Optimized Clustering Algorithm

Written by	Hemnath .MR
Document version	0.3
Last reviced date	18-march-2024

Contents

Abstract :

With the rapid expansion of social media platforms and the increasing complexity of online communities, managing and analyzing user-generated content is crucial. This project explores optimized clustering algorithms to navigate social media communities. It categorizes users, posts, and interactions based on shared interests, behaviors, and demographics, enabling personalized recommendations and targeted advertising. The algorithm also incorporates real-time processing to adapt to evolving trends, fostering positive engagement, and ensuring a safer online environment.

Introduction :

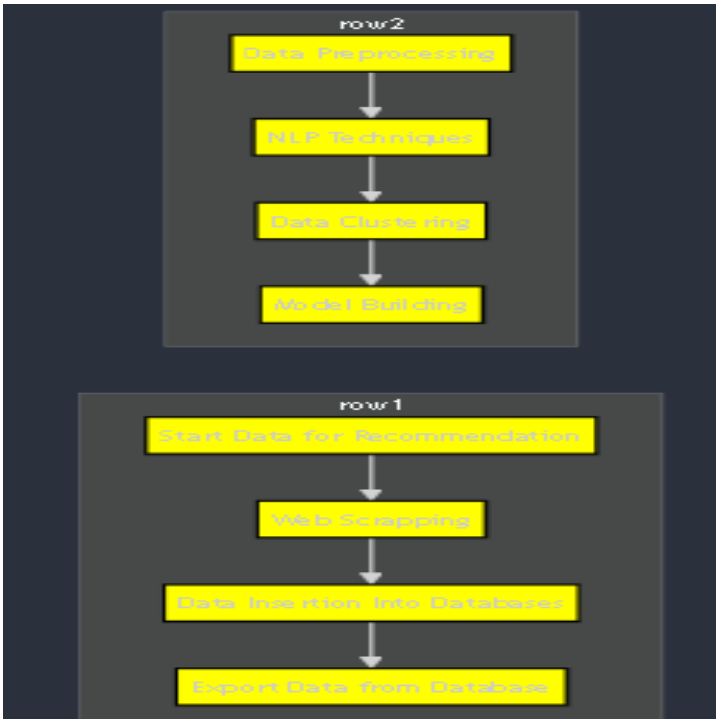
1.1 Why this Low-Level Design Document?

The purpose of this High-Level Design (LLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

Scope :

The LLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The LLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

Architecture



Project Description

3. Architecture Description

3.1. Data Description

For the social media cluster algorithm project, the dataset comprises posts from various social media platforms. The dataset contains text data and is initially in JSON format.

3.2. Data Collection

In order to enhance the dataset, additional data will be scraped from social media platforms to include user engagement metrics, post sentiments, and user demographics.

3.3. Data Transformation

The original JSON dataset will be transformed into a structured format such as CSV. The additional scraped data will be merged with the original dataset to create a comprehensive dataset for analysis.

3.4. Data Storage

a. Database Creation and Connection - A database will be created to store the merged dataset. If the database already exists, a connection will be established to it.
b. Table Creation in the Database - Tables will be created in the database to store the different aspects of the dataset, such as user information, post content, engagement metrics, and sentiment analysis.
c. Insertion of Data into the Database - The transformed and merged dataset will be inserted into the respective tables in the database.

3.5. Data Export

The data stored in the database will be exported as CSV files for further data pre-processing and model training.

3.6. Data Pre-processing

Data pre-processing will involve steps such as handling null values, removing stop words, punctuation, tokenization, and sentiment analysis. Additionally, techniques to handle imbalanced data sets will be applied.

3.7. Data Clustering

Clustering algorithms, such as K-Means, will be used to group the pre-processed data into clusters. The optimal number of clusters will be determined using techniques like the elbow method.

3.8. Model Building

After clustering, the best model for each cluster will be identified using techniques like Grid-Search. AUC scores will be calculated, and the model with the best score will be selected for each cluster. These models will be saved for use in recommendation.

3.9. User Data Collection

Physiological and behavioral data from users, including user engagement with social media content, will be collected.

3.10. User Data Validation

Validation of the user-provided data will be conducted to ensure its accuracy and consistency.

3.11. User Data Insertion into Database

The collected user data will be stored in a database, which can be either MySQL or MongoDB.

3.12. User Data Clustering

The pre-trained model will be used to predict clusters for the user data based on their social media behavior.

3.13. Model Selection and Recommendation

Based on the predicted cluster, the respective model will be used to recommend relevant social media content to the user.

3.14. Output Storage

The recommended content will be saved in the database and used to provide similar recommendations to other users with similar profiles.

3.15. Deployment

The model will be deployed on AWS for real-time social media content recommendation.

4. LOW LEVEL DESIGN (LLD)

3.10. Model Building

After clustering, the best model for each cluster will be selected based on AUC scores.

3.11. User Data Collection

Collection of physiological and behavioral data from users, including user engagement with social media content.

3.12. Data Validation

Validation of the user-provided data.

3.13. User Data Inserting into Database

Storing the collected user data in the database, which can be either MySQL or MongoDB.

3.14. Data Clustering

Using pre-trained models to predict clusters for the user data.

3.15. Model Call for Specific Cluster

Utilizing the respective model based on the predicted cluster to recommend social media content to the user.

3.16. Content Recommendation & Output Storage

After calling the model, recommended content will be saved in the database for future use and similar content recommendations.

3.17. Deployment

Deployment of the model to AWS for real-time social media content recommendation.
This is a workflow diagram for the Social Media Cluster Algorithm.

3.17. User test case

Certainly! Below are the modified unit test cases for the social media cluster algorithm project, presented in a tabular format:

Test Case Description	**Pre-Requisite**	**Expected Result**
Verify Application URL Accessibility	Application URL should be defined	Application URL should be accessible to the user
Verify Complete Application Loading	1. Application URL is accessible 2. Application is deployed	The application should load completely for the user when the URL is accessed
Verify User Sign-up	Application is accessible	The User should be able to sign up in the application
Verify Successful User Login	1. Application is accessible 2. User is signed up to the application	User should be able to successfully login to the application
Verify Visibility of Input Fields on Login	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be able to see input fields on logging in
Verify User's Ability to Edit Input Fields	1. Application is accessible 2. User is signed up to the application 3. User is logged in to	User should be able to edit all input fields

	the application	
Verify Presence of Submit Button	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should get a Submit button to submit the inputs
Verify Presentation of Recommended Results	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should be presented with recommended results on clicking submit
Verify Alignment of Recommended Results with User's Selections	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	The recommended results should be in accordance with the selections the user made
Verify User's Ability to Filter Recommended Results	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	User should have options to filter the recommended results as well
Verify Modification of KPIs based on User's Health Inputs	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	KPIs should modify as per the user inputs for the user's health
Verify Indication of Recipe Details in KPIs	1. Application is accessible 2. User is signed up to the application	The KPIs should indicate details of the suggested recipe

	3. User is logged in to the application	

This table provides a clear overview of the modified unit test cases for the social media cluster algorithm project