Low-Level Design

Author Prediction system

Contents

1. Introduction

The Low-Level Design (LLD) document outlines the internal logical design of the Author Prediction System for the Avila Bible. It provides a detailed description of the system's architecture, modules, and interactions between components.

2. What is Low-Level Design Document?

The LLD document describes the class diagrams, methods, and relations between classes and program specifications. It serves as a reference for developers to code the system's functionalities based on the document's specifications.

3. Scope

The LLD covers the component-level design process, including data structures, software architecture, source code, and algorithms. It defines the data organization and refinement process during system design.

4. Architecture

The architecture of the Author Prediction System consists of the following components:

- Data Loader Module
- Feature Extractor Module
- Model Training Module
- Model Prediction Module
- User Interface Module

5. Architecture Description

The Author Prediction System architecture is designed to load author data, extract features, train prediction models, interact with users, and deploy the system for production use.

6. Data Description

The system uses a dataset containing predefined features of text passages attributed to authors of the Avila Bible. This dataset is split into training and test sets.

7. Data Transformation

The input dataset is processed into a suitable format for model training and evaluation. Since features are pre-built, minimal transformation such as splitting and normalization is performed.

8. Data Insertion

Processed data is stored in a database for managing prediction logs. The database schema includes tables for input features, prediction results, and user activity.

9. Export Data

Stored data, including prediction logs and features, can be exported as Txt files for further analysis or debugging.

10. Data Preprocessing

Preprocessing involves ensuring data integrity and consistency. Given that features are already available, the focus is on handling missing values or outliers.

11. Model Training

The system trains machine learning models like Random Forest or Gradient Boosting on the training dataset. Hyperparameter tuning and evaluation are performed to ensure optimal model performance.

12. Model Prediction

The trained model is used to predict the author of new text passages based on input features. Predictions include the author’s name.

13. User Interaction

The system allows users to input features via a web interface, which communicates with the API. It includes user-friendly displays for predictions.

14. Deployment

The Author Prediction System is deployed on Render. The system is configured for reliability.

15. Unit Test Cases

Unit test cases are written to verify the functionality of each module, including data loading, feature extraction, model training, prediction. These tests ensure the system behaves as expected under different scenarios.