High-Level Design Document: Author Prediction System for Avila Bible

Revision Number: 1.0
Last Date of Revision: [04-01-2025]

Document Version Control

Date Issued	Version	Description	Author
04-01-2025	1.0	HLD - V1.0	Apurav

Here's an example High-Level Design (HLD) document for your **Author Prediction System for the Avila Bible** formatted similarly to your provided structure:

Contents

1. Abstract

This document outlines the high-level design for the Author Prediction System for the Avila Bible. The system uses machine learning techniques to classify text passages from the Avila Bible into one of 12 author categories based on textual features. It focuses on a scalable and efficient solution leveraging supervised learning for accurate predictions.

2. Introduction

The Author Prediction System aims to identify the authorship of text passages in the Avila Bible using a supervised learning approach. Given a dataset with pre-defined features, the system trains a classification model to predict the authors of the text passages efficiently.

3. Scope

- **Predict the author of a text passage using pre-extracted features.**
- **Ensure accurate classification into one of 12 author categories.**
- **Deploy a user-friendly web interface for users to input data and receive predictions.**

4. Definitions

- **Supervised Learning: A machine learning approach using labeled datasets for training.**
- **Render: A cloud platform used for deploying applications.**

5. General Description

The system comprises a pre-built dataset with labeled features, a machine learning model for prediction, and a deployed web interface on Render for end-user interaction. It provides fast and accurate predictions with minimal computational overhead.

6. Technical Requirements

- **Programming Language: Python**
- **Libraries: Scikit-learn, Pandas**
- **Model Types: Random Forest, CatBoostRegressor**

7. Data Requirements

- **Dataset with pre-extracted features and labels representing text passages and their authors.**
- **Dataset split into training, validation, and testing subsets.**

8. Tools Used

- **Jupyter Notebook**
- **Scikit-learn**
- **Render (for deployment)**

9. Hardware Requirements

The system can be deployed on standard hardware infrastructure including servers or cloud-based platforms such as AWS, Azure, or Google Cloud Platform, Render etc.

10. Constraints

- **Relies on the quality and diversity of the pre-built dataset.**
- **Model deployment depends on Render's hosting environment.**

11. Assumptions

- **Dataset accurately represents the authors' writing styles.**
- **Features provided in the dataset are relevant and sufficient for classification.**

12. Design Details

12.1 Process Flow

1. **Load pre-built dataset with extracted features.**
2. **Train supervised learning models on the dataset.**
3. **Evaluate model performance on validation and test sets.**

4. **Deploy the best-performing model on Render.**
5. **User inputs features, and the system predicts the author.**

12.2 Model Training and Evaluation

- **Model Choices: Random Forest, Gradient Boosting.**
- **Evaluation Metrics: Accuracy, F1 Score, Confusion Matrix.**

12.3 Deployment Process

- **Deploy the model API on Render.**
- **Use Flask to create an endpoint for predictions.**
- **Set up a frontend interface for user interaction.**

12.4 Event Log

- **Log model predictions and user queries for debugging and analysis.**

12.5 Error Handling

- **Handle missing or incorrect input features gracefully.**
- **Provide meaningful error messages for invalid requests.**

12.6 Performance

- **Optimize model for low latency on Render.**

12.7 Reusability

- **Design modular components for feature extension or new datasets.**

12.8 Application Compatibility

- **Accessible via web browsers with no additional software requirements.**

12.9 Resource Utilization

- **Minimize resource usage during training and deployment to fit Render's hosting constraints.**

12.10 Deployment

- **Deploy application on Render.**

12.11 KPIs (Key Performance Indicators)

- **Prediction Accuracy: ≥ 99%**
- **Deployment Uptime: ≥ 99%**

13. Conclusion

This document provides the high-level design for the Author Prediction System for the Avila Bible. By leveraging pre-built features, supervised learning, and a cloud-based deployment approach, the system ensures efficient and accurate predictions for research and academic purposes.

14. Reference

- **Documentation for Scikit-learn.**
- **Render platform deployment guides.**