Document Version Control: 1

**Abstract:**

This High-Level Design Document outlines the architecture and design considerations for the Adult Census Income Prediction project. It defines the scope, technical requirements, design details, and deployment process.

**1. Introduction:**

1.1 Why this High-Level Design Document?

This document serves as a guide for understanding the architecture and design decisions of the Adult Census Income Prediction project.

**1.2 Scope:**

The scope of this project includes developing a machine learning model to predict an individual's income based on demographic information. The document outlines the high-level design aspects of the project.

**1.3 Definitions:**

N/A

**2. General Description:**

2.1 Product Perspective:

The project involves building a predictive model using machine learning techniques. It interfaces with data preprocessing, training, evaluation, and deployment components.

**2.2 Problem Statement:**

The goal is to predict whether an individual's income is above or below a certain threshold using demographic features. This requires developing a robust and accurate machine learning model.

**2.3 Proposed Solution:**

The project aims to use historical census data to train a classification model. This model can then predict income based on demographic features of individuals.

**2.4 Further Improvements**:

Future enhancements may include refining the model, addressing bias, and incorporating additional features for better accuracy.

**2.5 Technical Requirements:**

- Programming languages: Python

- Machine learning libraries: Scikit-learn, TensorFlow

- Data preprocessing tools: Pandas, NumPy

- Version control: Git

**2.6 Data Requirements**:

The project requires a labeled dataset containing demographic features and corresponding income labels. Data preprocessing will handle missing values and categorical encoding.

**2.7 Tools Used:**

2.7.1 Hardware Requirements:

- Standard PC with sufficient RAM and CPU for data processing and model training.

- GPU (optional) for faster training of deep learning models.

**2.7.2 ROS (Robotic Operating System):**

ROS is not applicable to this project.

**2.8 Constraints:**

- Availability of quality labeled data.

- Model complexity based on hardware limitations.

**2.9 Assumptions:**

- The dataset is representative of the target population.

- Features have been preselected based on their relevance.

**3. Design Details:**

**3.1 Process Flow:**

**3.1.1 Model Training and Evaluation:**

- Data preprocessing: Handle missing values, one-hot encode categorical variables.

- Model selection: Choose appropriate classification algorithm.

- Model training: Train the selected model using the preprocessed data.

- Model evaluation: Assess the model's performance using various metrics.

**3.1.2 Deployment Process:**

- Deploy the trained model to a production environment.

- Implement monitoring to track the model's performance.

**3.2 Event Log:**

Not applicable.

**3.3 Error Handling:**

Implement error handling mechanisms to catch data preprocessing and deployment errors.

**3.4 Performance:**

The performance of the model will be evaluated using standard classification metrics such as accuracy, precision, recall, and F1-score.

**3.5 Reusability:**

The machine learning pipeline can be reused for similar prediction tasks with appropriate modifications.

**3.6 Application Compatibility:**

The model can be used in any environment that supports the required programming languages and libraries.

**3.7 Resource Utilization:**

The model's resource utilization will depend on the complexity of the chosen algorithm and the size of the dataset.

**Deployment:**

The trained model will be deployed in a production environment. Monitoring mechanisms will be put in place to ensure the model's ongoing performance and accuracy.