Adult Census Income Prediction

Low Level Design

Written by	Meghana Munjeti
Document Version	0
Last Revised Date	07/10/2023

Contents

3.3. Data Preprocessing

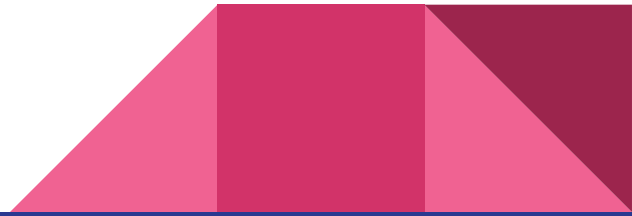
3.4. Model Building

3.5. Model Selection

3.6. Front-end

3.7. Deployment

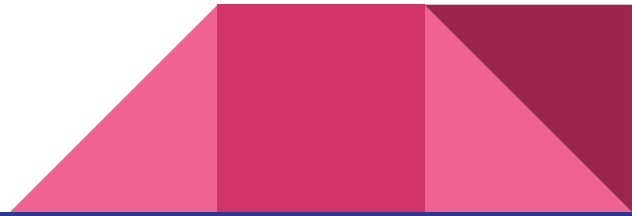
4. Unit Test Case



1. Introduction

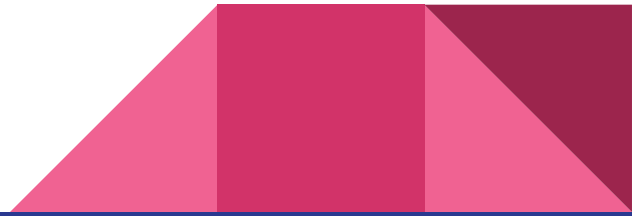
1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

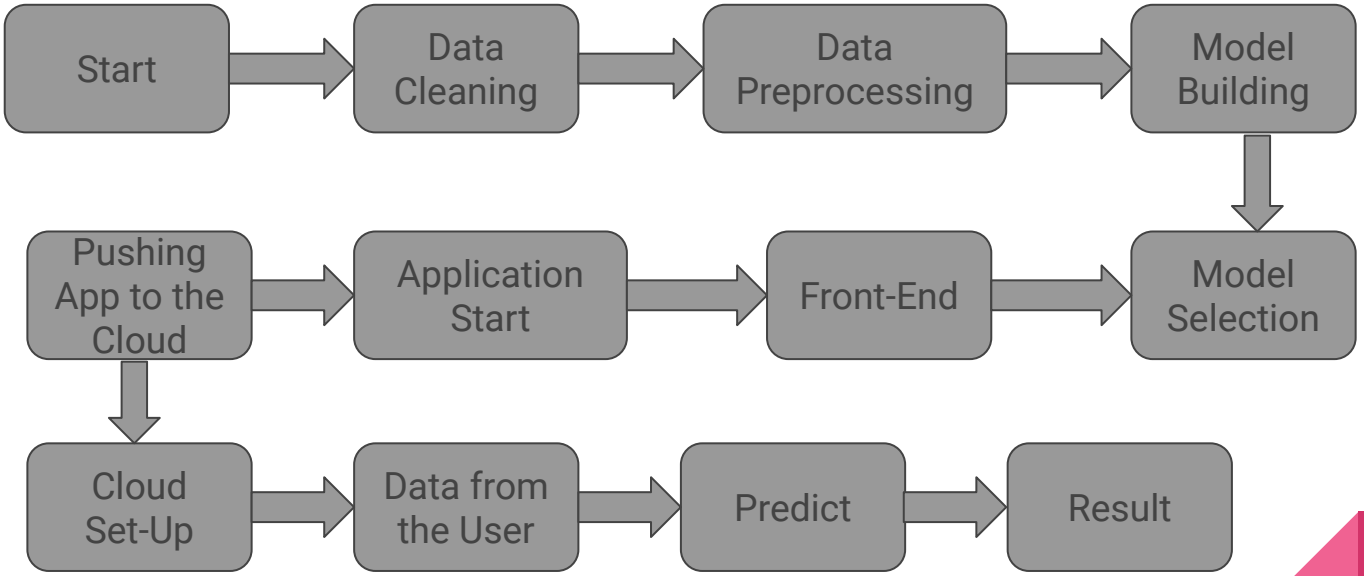


1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work



2. Architecture



3.Architecture Description

3.1.Data Description

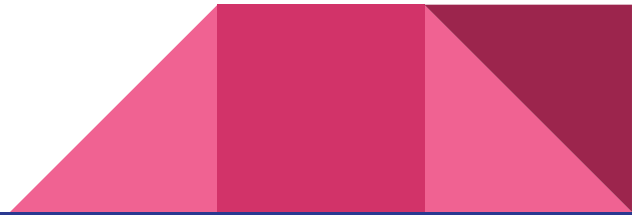
The following is a description of the variables in the Adult Census Income Dataset:

- age: The age of the individual in years.
- workclass: The individual's work class, such as Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- fnlwgt: The final weight for the individual.
- education: The individual's education level, such as Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-voc, Assoc-acdm, 9th, 7th-8th, 12th, Masters, 1st-4th, 5th-6th, 10th.

- education-num: The individual's education level in years.
- marital-status: The individual's marital status, such as Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- occupation: The individual's occupation, such as Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Machine-op-inspect, Adm-clerical, Transport-moving, Handlers-cleaners, Farming-fishing, Armed-Forces.
- relationship: The individual's relationship to the householder, such as Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- race: The individual's race, such as White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- sex: The individual's sex, such as Female, Male.
- capital-gain: The individual's capital gain in the past year.

- capital-loss: The individual's capital loss in the past year.
- hours-per-week: The number of hours the individual worked per week in the past year.
- native-country: The individual's native country.
- income: The individual's income in the past year.

The Adult Census Income Dataset is a valuable resource for researchers and practitioners who are interested in studying income inequality and other social issues. The dataset can be used to train machine learning models to predict income, to identify factors that contribute to income inequality, and to develop policies to reduce income inequality.



3.2.Data Cleaning

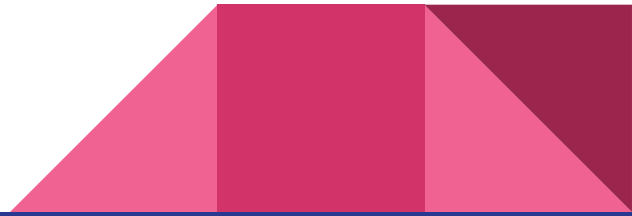
We Clean the dataset by removing missing values and duplicates.

3.3.Data Preprocessing

We Standardize the numerical data and convert the categorical data into numerical data by using label encoding

3.4.Model Building

We apply various classification models like Logistic Regression, Random Forests on the data



3.5.Model Selection

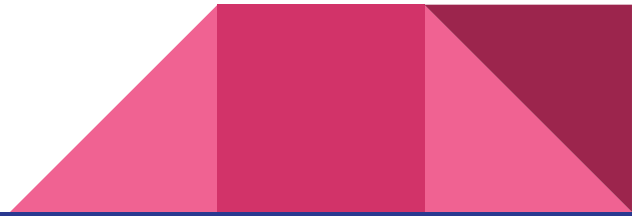
We select the model which gives highest f1-Score

3.6.Front-End

We use Flask to develop Web-Application and HTML for Front-End

3.7.Deployment

We will deploy our model in AWS Cloud Platform



4.Unit Test Case

Test Case Description	Prerequisite	Expected result
Verify whether the Application URL is accessible to the user	1. Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed

Verify whether user is able to see input fields on	1. Application URL is accessible 2. Application is deployed	User should be able to see input fields on logging in
Verify whether user is able to edit all input fields	1. Application URL is accessible 2. Application is deployed	User should be able to edit all input fields
Verify whether user gets Predict button to submit the inputs	1. Application URL is accessible 2. Application is deployed	User should get Predict button to submit the inputs
Verify whether user is presented with result	1. Application URL is accessible 2. Application is deployed	User should be presented with recommended results on clicking Predict