# Low Level Design (LLD)

# Income Prediction

Revision Number: 0

Manojprabakaran.M

## Document Version Control

| Date Issued | Version | Description | Author |
|---|---|---|---|
| 12th Dec 2023 | 1.0 | First Draft | Mnaojprabakaran |
| | | | |

# Contents

## Abstract

Income prediction is a fundamental task in the field of machine learning, with significant realworld applications. In this study, we focus on the predictive analysis of individuals' incomes, aiming to classify whether a person's income falls below or above the 50k threshold. Our research utilizes a dataset consisting of several key features, including sex, education, workclass, capital gains, capital losses, and the type of work performed. The primary objective of this project is to design a robust and accurate income prediction model, which can provide valuable insights for various socio-economic applications.

By leveraging algorithms and data analysis, we aim to deliver a tool that can assist policymakers, businesses, and researchers in making informed decisions related to income disparities, education, and employment. The insights gained from this research can contribute to a better understanding of socio-economic factors affecting individuals' incomes, ultimately helping create a fairer and more equitable society.

# 1 Introduction

## 1.1 Why this Low-Level Design Document?

The purpose of this document is to present a detailed description of the Income Prediction system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

## 1.2 Scope

- This software system will be a Web application, this system will be designed to predict a income.
- It aims to provide comprehensive Income information, including age,sex,capital gain, capital loss and other factors.

## 1.3 Out of Scope

Delineate specific activities, capabilities, and items that are out of scope for the project.

- Many things are not under the scope such as cities , married status etc .

5

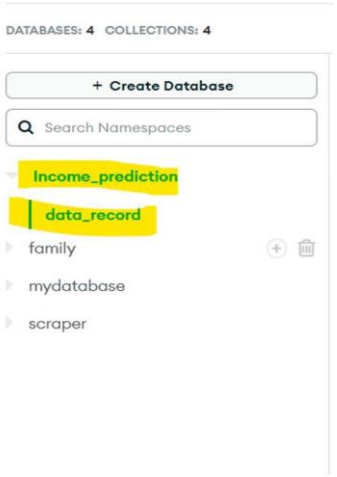
# 2 Technical specifications

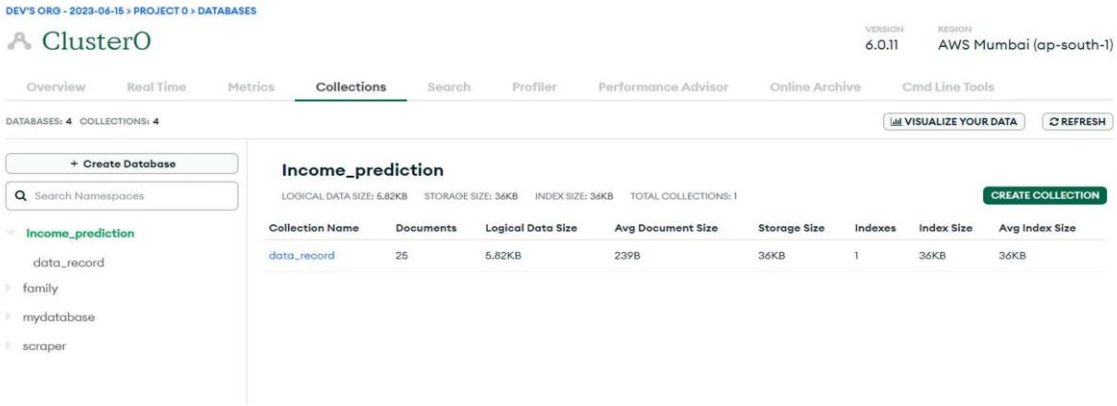
## 2.1 Web-based specs

- Client-side technologies: HTML, CSS
- Server-side technologies: Flask, Python • Database: MongoDB for storing new test data.
- Data files at One location

### 2.1.1 Database overview

The application uses Income_prediction database and it consists collections Data_record.



## 2.2 Logging

We should be able to log every activity done by the user.

- The System identifies at what step logging required
- The System should be able to log each and every system flow.
- Developers can choose logging methods. You can choose database logging/ File logging as well.

6

- System should not be hung even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

## 2.3 Database

System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

- MongoDB database design to Income prediction data efficiently.
- Indexing and query optimization for fast data retrieval.

## 2.4 Deployment

- Deployment on cloud infrastructure (i.e., AWS) for scalability and availability.
- Continuous integration and continuous deployment (CI/CD) pipelines for automated updates.

AWS


EBS

CODEPIPELINE

7

## 3 Technology stack

| | |
|---|---|
| Front End | HTML/CSS |
| Backend | Flask/Python |
| Database | MongoDB |
| Deployment | AWS |

# 4 Proposed Solution

## 4.1 Key Features

### 4.1.1 Age Features:

- The system predicts a users age based on available data, contributing to demographic analysis.

### 4.1.2 Workclass Classification:

- Users workclasses are categorized, providing insights into employment sectors and income disparities.

### 4.1.3 Gender-Based Income Prediction Features:

- The system predicts income levels based on gender, highlighting gender-related income disparities.

### 4.1.4 Education Impact Analysis:

- Users' education levels are evaluated for their influence on income, offering insights into the significance of education in income disparities.

### 4.1.5 Occupation-Based Income Prediction:

- The system predicts income categories based on the occupation, shedding light on the role of professions in determining income.

### 4.1.6 Race and Income Correlation:

- Income categories are associated with race, providing valuable insights into racial income disparities.

### 4.1.7 Capital Loss and Gain Analysis:

- Users' financial gains and losses are considered in income prediction, offering a holistic view of financial well-being.

### 4.1.8 Hours Worked Impact:

- The number of hours worked per week is analyzed for its influence on income, providing insights into labor market dynamics.

### 4.1.9 Country-Specific Income Prediction:

- Income categories are determined based on the user's country of residence, aiding in international socio-economic analysis.

# 5 Architecture Description

## 5.1 Data Description

- The This dataset contain a row/column (32561,15)

### 5.1.1 Data Insertion:

- Data Ingestion from the data by using the pandas lib.

### 5.1.2 Data Transformation:

- In this transformation process. Under this crucial process is done in this data cleaning process has been persuaded, and many process can be handle under this process such as missing values , replace nan values and creating a pipeline . Data Preprocessing steps we could use are Null value handling, stop words removal, punctuation removal, Imbalanced data set handling, Handling columns with standard deviation zero or below a threshold, etc.

### 5.1.3 Data File Station:

- In this class all the file are created inside the artifacts folder. Stored file location and type of the file

### 5.1.4 Model Training:

- Under this model are trained and model training are gaves as a model pickle file which is used for training validation . we will find the best model for each algo, algorithms will be passed with the best parameters derived from Grid-Search. We will calculate the Accuracy scores for models and select the model with the best score. Similarly, the models will be selected for each score. All the models for every cluster will be saved for use in Recommendation.

### 5.1.5 Utils:

- Are used to handle all the genric code which is used again and again.

### 5.1.6 Data Storage:

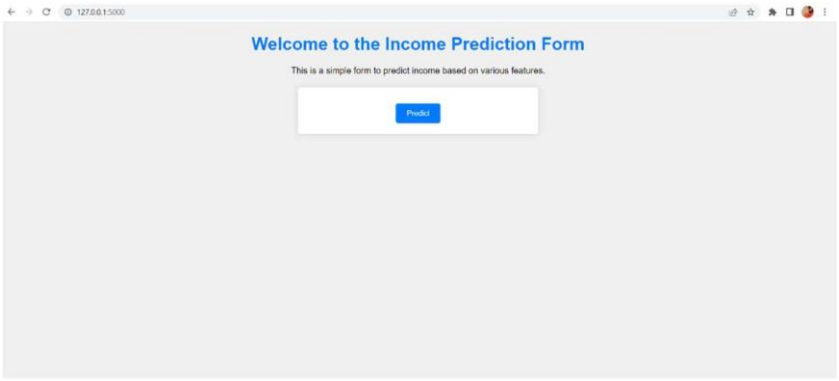
- Data storage is used to store new data into MongoDB . Database creation and connection.

### 5.1.7 Prediction Pipeline:

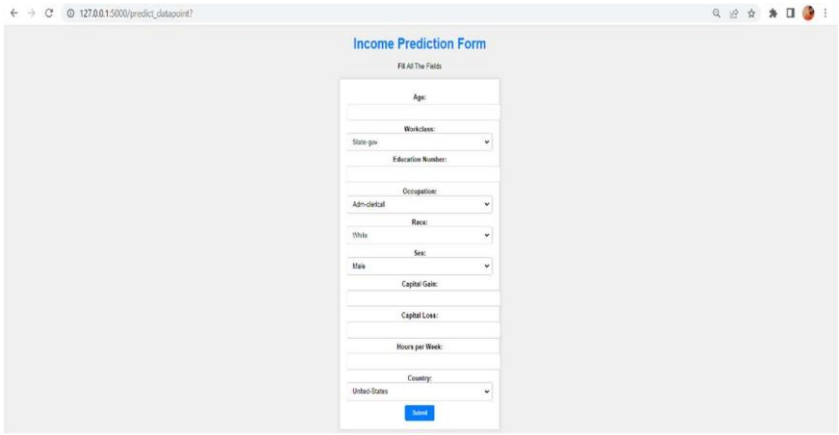
- Are used for predicting new data and also used for creating dataframe

# 6 User I/O workflow

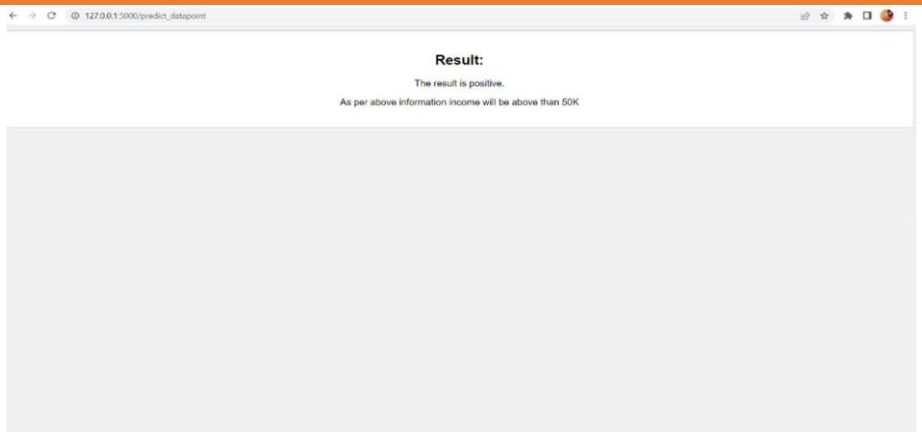
- Home page




- Form Page




- Result Page




User Module:

- Hit the URL
- Home page will open and click on Predict buttom page where all the field to filled
- Result page will pop up and show result.
- Directly moved to the form

## 7 Test cases

| Test case | Steps to perform test case | Module | Pass/Fail |
|---|---|---|---|
| 01 | Hit the URL | Application module, index function is executed because '/' URL is GET method and it return Home page of application | Pass |
| 02 | Click on Button Predict as User | Application module, Moved to Form page because predict button is POST method | Pass |
| 03 | After Click Predict Button it will automatically redirect you form page | Form page is redirect to user and able to access form page | Pass |
| 04 | Verify user is able to fill all the options (form page) | There are option easy to fill by drop down menu, user has access to fill the fields | Pass |
| 05 | Verify whether user is presented with recommended results on clicking submit Button | User should be presented with recommended results on clicking submit | Pass |
| 06 | Verify whether the recommended results are in accordance to the selections user made | Admin module,passed | Pass |
| | | | |
| | | | |

| | | | |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| 13 | | | |
| | | | |
| | | | |