Low Level Design Document

Adult Census Income Prediction

Revision Number : 1.0

Last date of revision : 20/06/2022

Ritesh Naik

Darshan Naik

Document Version Control

Date Issued	Version	Description	Author
20-June-2022	1.0	Initial Release	Ritesh Naik

Contents

Abstract

Inequality in wealth and income is a major source of worry, particularly in the United States. One reasonable motivation to lessen the world's rising level of economic disparity is the possibility of reducing poverty. The notion of universal moral equality promotes long-term development and improves a country's economic stability. Governments in several countries have been working hard to address this issue and find the best answer possible. The goal of this project is to demonstrate how machine learning and data mining techniques can be used to solve the problem of income inequality. The Adult Dataset from Kaggle was utilized for this.

Based on a set of attributes, classification has been done to forecast whether a person's annual income in the United States fits into the income categories of greater than 50K Dollars or less than 50K Dollars.

1. Introduction

1.1 Purpose of the Document

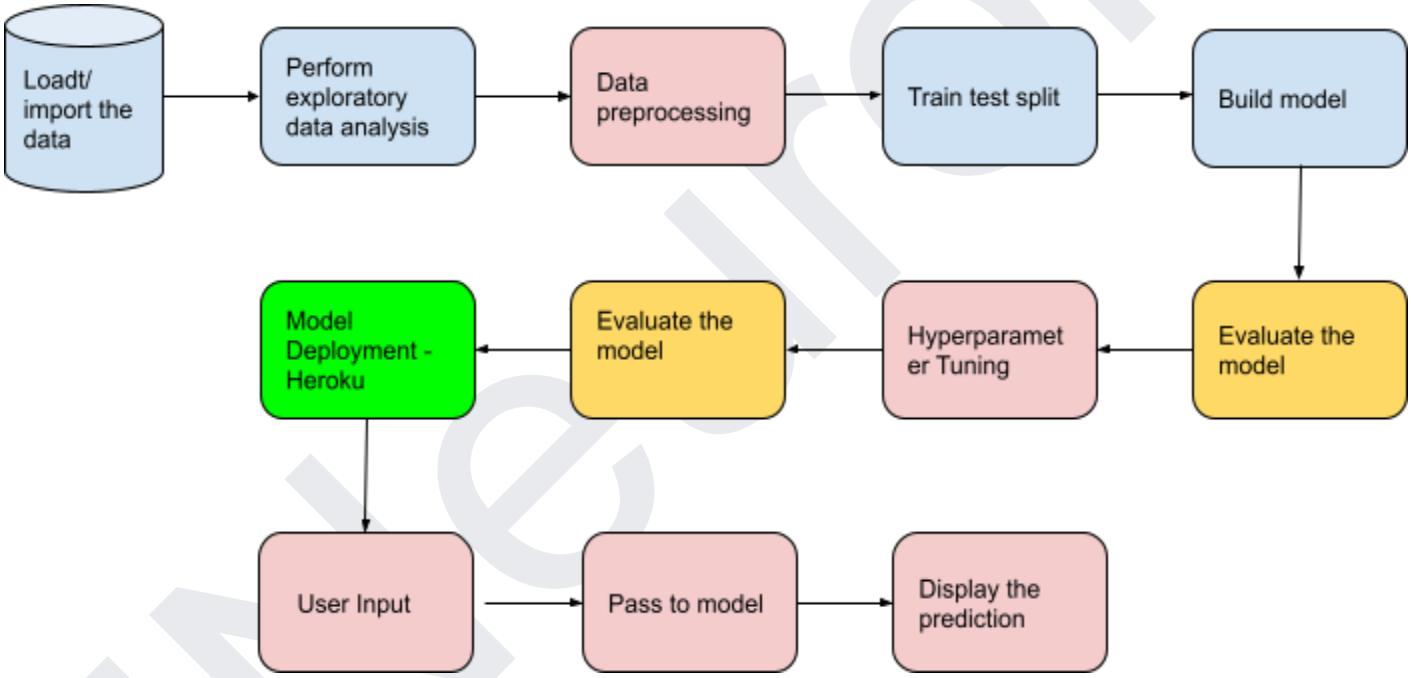
A low-level design document (LLDD), often known as an LLD, aims to describe the internal logical structure of the actual programme code for Adult Census Income Prediction. It explains the function, characteristics, and user interface of the system. LLD explains class diagrams that show the relationships and methods between classes and programme specifications. In order for the programmer to create the programme directly from the document, it describes the modules.

A person is essentiv ally classified into the >50K group or the =50K group in this binary classification problem.

1.2 Scope of LLD

A Web application will be used to run this programme. Based on the person's educational background, prior or present employment experience, geographic area, and a number of other factors, this method will be developed to forecast whether the person will have an annual income of more than $50,000 or less than $50,000.

2. Architecture



3. Architecture Information

3.1. Load / Import the data

We have used the Adult Census Dataset. The dataset contains 15 feature columns

and 32561 rows

3.2. Perform Exploratory Data Analysis(EDA)

After loading the data, understand the data by checking for the missing values, average values, any correlation among features/ target variable. Do some visualization of the relationship between the features/target variable.

3.3. Data Preprocessing

To handle null values, punctuation, categorize the categorical characteristics to the proper format, encode the categorical features, and scale the numerical features of the dataset, pre-processing is performed. Finally, the machine learning model is constructed using the most pertinent features.

3.4. Test Train Split

After we process the data, we will split the data into train & test dataset. This is to train the model, & Ideally 80% data into train dataset & 20% dataset into test dataset.

3.5. Build Model

When the data has been split, the dataset is used to train a variety of models, and GridSearchCV is used to pass a grid of parameters to each model so that it can choose the parameters that are best for it.

3.6. Evaluate the model

Which model is the best is determined using the F1 score and the AUC score of the models.

3.7. Hyperparameter tuning

Tune the hyperparameter for best model accuracy.

3.8. Evaluate the model

Check the model accuracy again & (repeat step 3.7) until the ideal accuracy is achieved (i.e Above 85%). We need to pass the test dataset to the model & check its accuracy

3.9. Model Deployment

Deploy the model to Heroku platform which is a cloud PaaS & help us to run the app in the cloud. This can be used by anyone anywhere in the world.

3.10. User Input

Collect the user input (independent variables) by webform.

3.11. Pass user input to model

Pass these independent variables to model for prediction

3.12. Display the prediction

Display the prediction to the user!

4. Unit Test Cases

Test Case description	Prerequisite	Expected Result
Check to see if the user can access the Application URL	Application URL must be defined	User should be able to access the website when he enter the URL
Verify whether app loads or not	1. Application URL accessible 2. Application is deployed	Application must load without any error
Check whether user can edit all the fields	User is must be able to edit the fields	User is able to edit the input
Check whether app warns the user if the user enters irrelevant data	App shows warning if the desired data is not entered	App warns the user when irrelevant data is entered
Check whether user able to click on “predict” button	Button is accessible	User is able to hit the “predict” button
Verify whether user is presented with prediction after clicking on submit	1) User is able to edit the fields 2)User is able to click on predict button	User should be presented with prediction results on clicking submit
Verify whether the predicted results are in accordance to the selections user made	Application is accessible	The results that are anticipated should match the user's choices.

THANK YOU!