Low-Level Document

Adult Income Census Prediction Application

Written By	**Mohammad Sohail Parvez**
Document Version	0.1
Last Revised Date	

Contents

1. Introduction……………………………………………….1
   1.1 What is Low-Level Design Document?...................1
   1.2 Scope……………………………………………….1
2. Architecture………………………………………………..2
3. Architecture Description………………………………….2
   3.1 Data Collection………………………………………..2
   3.2 Exploratory Data Analysis……………………………2
   3.3 Data Preprocessing…………………………………...2
   3.4 Feature Engineering…………………………………..3
   3.5 Model Creation………………………………………...3
   3.6 Cross validation………………………………………..3
   3.7 Model Dump…………………………………………….3
   3.8 Data from User………………………………………...3
   3.9 Data Validation………………………………………….3
   3.10 User Data inserting into Database…………………4
   3.11 Model Call/.pkl file loaded……………………………4
   3.12 Docker……………………………………………………4
   3.13 Deployment……………………………………………….4
4. Unit Test Cases………………………………………………..4

1. Introduction

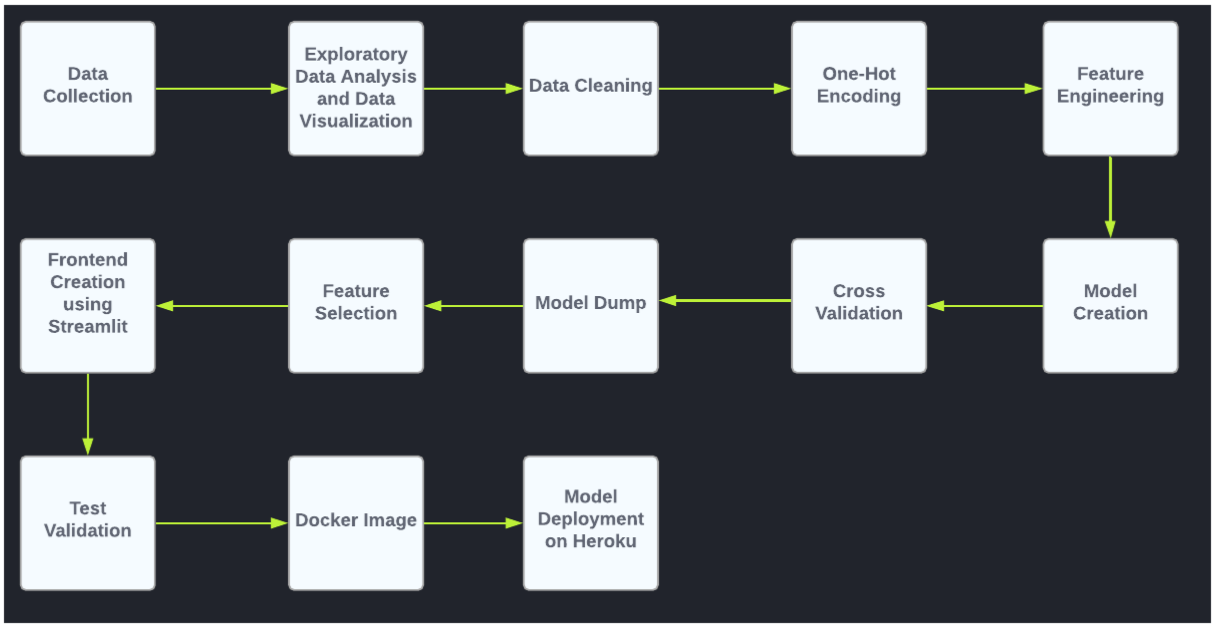
1. 1 What is a Low-Level Design Document?

The goal of LLD or a low-level design document(LLDD) is to give the internal logical design of the actual program code for the adult income prediction estimation 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 Collection

The dataset is downloaded from Kaggle. The dataset contains 15 columns and 32k+ rows. The dataset includes age, workclass , fnlwgt , education, education-num ,marital-status ,occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week ,country ,salary. This is given in the comma separated value format(.csv).

3. 2 Exploratory Data Analysis

In eda, we have seen various insights from the dataset like checking top 5 rows and bottom 5 rows. There are no null values in the dataset.

3. 3 Data Preprocessing

3.3.1 Data Preprocessing 1

In this process some column values contain '?' init. So, I replaced it with 'np.nan' values. After replacing it, I dropped the 'nan' values.

3.3.2 Data Preprocessing 2.

Some of the columns are categorical dtype. So, encoding the features as needed

3.3.3 Data preprocessing 3

Checking the correlation between a binary variable and continuous variables using point biserial correlation. The feature which has negative value with the target feature is dropped

3. 4 Feature Engineering.

New features are created regarding the ‘workclass’ and ‘marital-status’ column and drop the old one in the dataset.

3. 5 Model Creation.

After cleaning the data and completing the feature engineering. We have split data into training and testing sets. The dataset is imbalance, so we applied SMOTE technique to oversample our data and implemented various classification algorithms like Logistic Regression, Random Forest, Gradient Boosting, Decision Tree, SVM, XGBoost, Catboost, KNN and also calculated their accuracies on test and train data.

3.6 Cross-validation

For cross-validation, we implemented the ‘RepeatedStratifiedKFold’ algorithm. Which have good accuracy.

3.7 Model Dump

After comparing all accuracies and checking all regression metrics, CatboostClassifier is chosen as our best model by their results, so we have dumped these models in a pickle file format.

3.8 Data From User

Here we will collect user's requirement to classify their income using age, sex, capital gain, capital loss, hours per week, country, employment type.

3.9 Data Validation

Here Data Validation will be done, given by the user

3.10 User Data Inserting into Database

Collecting the data from the user and storing it into the database. The database can be the Cassandra Cloud Version.

3.11 Model Call/.pkl file loaded

Based on the User input, it will be thrown to the backend in the dictionary format.We are loading our pickle file in the backend and predicting whether a person has an income over 50K or not.

3.12 Docker

Implemented docker in our project so that anyone can run our docker image which is uploaded as an open source on a docker hub. So it is useful for a developer to pull images from the docker and run on any platform.

3.13 Deployment

The model is deployed on Heroku.

4. Unit Test Cases

Test Cases 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 the user is giving standard input.	1. Handled test cases at backends.	Users should be able to see successfully valid results.
Verify whether user is able to see input fields	1. Application is accessible	User should be able to see input fields

Verify whether user is able to edit all input fields	2.Application is accessible. User is logged in to the application	User should be able to edit all input fields
Verify whether gets predict or classify the income is >50K or <=50K	1. Application is accessible. 2. User is logged in to the application	Users should get the Submit button to submit the inputs.
Verify whether the user is presented with recommended results on clicking submit.	1.Application is accessible. 2.User is logged in to the application	User should be presented with recommended results on clicking submit
Verify whether the recommended results are in accordance to the selections user made	1.Application is accessible. 2.User is logged in to the application	The recommended results should be in accordance to the selections user made
Verify whether user has options to filter the recommended results as weel	1. Application is accessible 2. User is logged in to the application	User should have options to filter the recommended results as well.