High Level Design (LLD)

Adult Census Income Prediction

Revision Number: 1.5
Last date of revision: 08/7/2024

Jyoti Pandey

Document Version Control

Date Issued	Version	Description	Author
1st July 2024	1.1	First Draft	Jyoti Pandey
2nd July 2024	1.2	Added Workflow charts	Jyoti Pandey
3rd July 2024	1.3	Added Proposed Solution	Jyoti Pandey
6th July 2024	1.4	Added Test Cases	Jyoti Pandey
8th July 2024	1.5	Added KPIs	Jyoti Pandey

Contents

Abstract

Data has always been the backbone of many important decisions. When an assumption is backed up by facts and numbers, the chances of incorrectness and bad decisions decrease and in today's world, Countless decisions in private and public sectors are based on Census data. Census data is the backbone of the democratic system of government, highly affecting the economic sectors. Census-related figures are used to distribute the federal funding by the government into different states and localities.

The above introduction had an aim to increase the awareness about how the income factor actually has an impact not only on the personal lives of people, but also an impact on the nation and its betterment. We will now have a look on the data extracted from the 1994 Census bureau database, and try to find insights about how different features have an impact on the income of an individual and also do some predictive analysis using the modern Data Science and Machine Learning techniques.

1 Introduction

1.1 Why is High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

1. Present all of design aspects and define them in detail
2. Describe all user interface being implemented
3. Describe the hardware and software interfaces
4. Describe the performance requirements
5. Include design features and architecture of the project. List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

2 General Description

2.1 Definitions

Term	*Description*
ACIP	Adult Census Income Predictor
Database	Collection of the Information monitored by the system
Cloud	A datacentre full of servers connected to the internet performing a service
IDE	Integrated Development Environment
UI	User Interface

2.2 Product Perspective

The ACIP is a Machine Learning based classification model which helps us to do predictive analysis on the Income of a person using certain parameters.

2.3 Problem Statement

To create an AI based solution for predictive analysis of a person's annual income and also deploy it in the form of a UI.

The Goal is to predict whether a person has an income of more than 50K a year or not. This is basically a binary classification problem where a person is classified into the >50K group or <=50K group.

2.4 Proposed Solution

Using all the standard techniques used in the life-cycle of a Data Science project starting from Data Exploration, Data Cleaning, Feature Engineering, Model Selection, Model Building and Model Testing and also building a frontend where a user can fill their information in the form input and get the output instantly.

1. User enters the values for age, capital gain, capital loss, hours per week.
2. The User chooses the work class, education marital status, occupation, relationship, race sex, country by clicking on one of the available options.
3. The system presents the set of inputs required from the user.
4. The user gives required information.
5. The system should be able to predict whether income is greater than 50K dollars or not based on the user information.

2.5 Further Improvements

The ACIP can be easily embedded inside any website or an application and can be used to find out whether a person earns more than 50K $ annually or not and can be used by various governmental / non- governmental / private agencies around the world.

This can also be improved further by feeding the model with more data, this can data from various UCI repositories, from various banks, data scraped from internet, etc

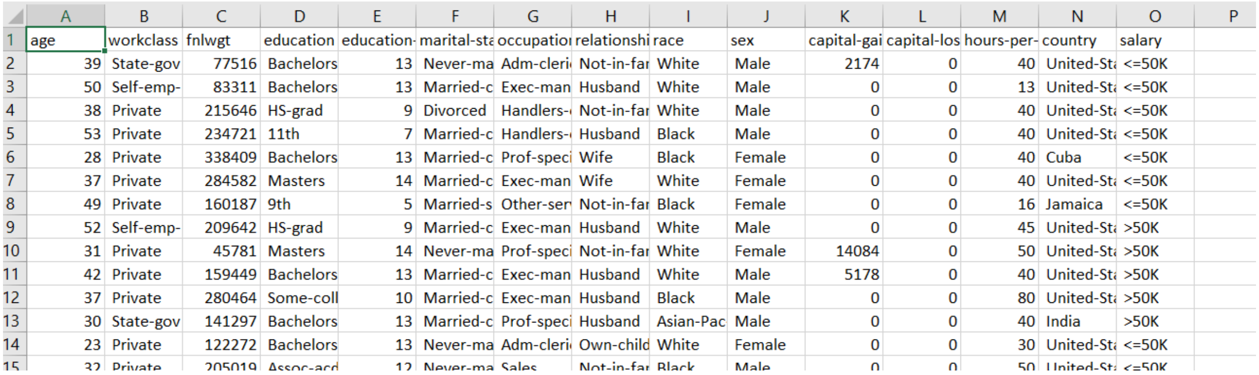
2.6 Data Requirements

Data requirement completely depend on our problem statement. We need the dataset from the census of 1994, to train our model. Each record in the dataset must have the following features which are important to determine one's income:

1 **Age:** Age of the person whom the income has to be determined.
2 **Hours per week:** The number of hours the person work per week.
3 **Capital Gain/Loss:** Whether the person had any capital gain or loss or none.
4 **Country:** Country in which the person lives.
5 **Educational Level:** Highest level of education the person has attained.
6 **Marital Status:** Whether the person is married or single **Occupation:** occupation of the person.
7 **Workclass:** Working class of the person.
8 **Gender:** Gender of the person.
9 **Race:** Race to which the person belongs.

These are the required set of parameters required in order to predict one's annual income.

There are 32561 rows and 15 columns in the dataset.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	age	workclass	fnlwgt	education	education-	marital-sta	occupatio	relationshi	race	sex	capital-gai	capital-los	hours-per-	country	salary	
2	39	State-gov	77516	Bachelors	13	Never-ma	Adm-cleri	Not-in-far	White	Male	2174	0	40	United-Sta	<=50K	
3	50	Self-emp-	83311	Bachelors	13	Married-c	Exec-man	Husband	White	Male	0	0	13	United-Sta	<=50K	
4	38	Private	215646	HS-grad	9	Divorced	Handlers-	Not-in-far	White	Male	0	0	40	United-Sta	<=50K	
5	53	Private	234721	11th	7	Married-c	Handlers-	Husband	Black	Male	0	0	40	United-Sta	<=50K	
6	28	Private	338409	Bachelors	13	Married-c	Prof-speci	Wife	Black	Female	0	0	40	Cuba	<=50K	
7	37	Private	284582	Masters	14	Married-c	Exec-man	Wife	White	Female	0	0	40	United-Sta	<=50K	
8	49	Private	160187	9th	5	Married-s	Other-ser	Not-in-far	Black	Female	0	0	16	Jamaica	<=50K	
9	52	Self-emp-	209642	HS-grad	9	Married-c	Exec-man	Husband	White	Male	0	0	45	United-Sta	>50K	
10	31	Private	45781	Masters	14	Never-ma	Prof-speci	Not-in-far	White	Female	14084	0	50	United-Sta	>50K	
11	42	Private	159449	Bachelors	13	Married-c	Exec-man	Husband	White	Male	5178	0	40	United-Sta	>50K	
12	37	Private	280464	Some-coll	10	Married-c	Exec-man	Husband	Black	Male	0	0	80	United-Sta	>50K	
13	30	State-gov	141297	Bachelors	13	Married-c	Prof-speci	Husband	Asian-Pac	Male	0	0	40	India	>50K	
14	23	Private	122272	Bachelors	13	Never-ma	Adm-cleri	Own-child	White	Female	0	0	30	United-Sta	<=50K	
15	32	Private	205019	Assoc-acd	12	Never-ma	Sales	Not-in-far	Black	Male	0	0	50	United-Sta	<=50K	

2.7 Tools Used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, FastAPI, and a few other libraries were used to build the whole model.

NumPy
Pandas
matplotlib
seaborn
scikit learn
mongoDB
FastAPI
amazon web services
git
GitHub

1. Visual Studio code was used as IDE
2. For visualization tasks, matplotlib, seaborn were used
3. AWS was used for deployment of the model
4. Fast API and HTML were used for building the web application and server to run the code
5. MongoDB was used to storage and retrieval of data
6. GitHub is used as version control system
7. NumPy and Pandas were used to clean and interpret data
8. Scikit learn was used to cross validate and compare different models.
9. Random Forest Classifier was used to build the final model

Hardware Requirements:

1. Windows Server, Linux, or any operating system that can run as a webserver, capable of delivering HTML5 content.
2. Minimum 1.10 GHz processor or equivalent.
3. Between 1-2 GB of free storage
4. Minimum 512 MB of RAM
5. GB of hard-disk space

2.8 Constraints

The front-end must be user friendly and should not need any one to have any prior knowledge in order to use it.

2.9 Assumptions

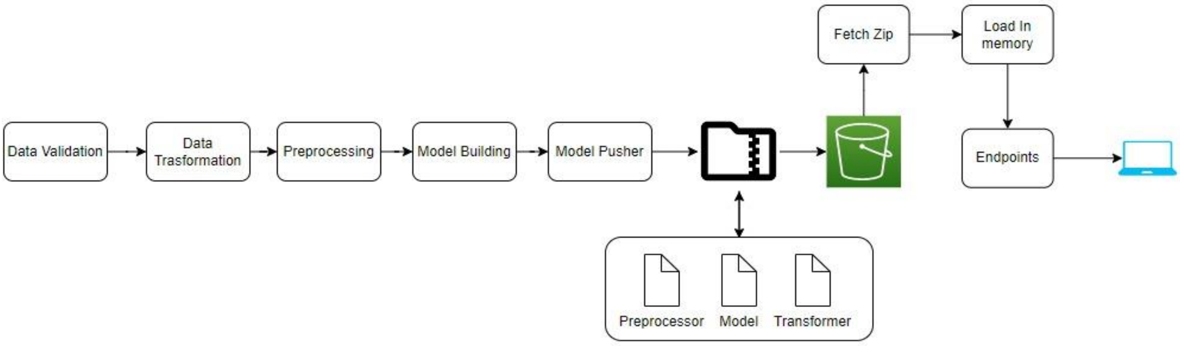
The main objective of this project is to implement the use case as previously mentioned (2.3 problem statement) for new dataset that comes through the UI. It is assumed that all aspects of this project have the ability to work together as the designer is expecting and also the data on which our model is trained is as correct as possible.

3 Design Details

3.1 Process Flow

For accomplishment of the task, we will use a trained Machine Learning model. The process flow diagram is shown below:

Proposed Methodology:



Data Ingestion:

Uploaded the data to MongoDB database and retrieved the data from MongoDB by creating connection in data ingestion component.

Data Validation:

In this component, data is validated by checking that all the columns are retrieved from MongoDB database.

Data Transformation:

In this component, data has checked and proper techniques are used to handle missing values, incorrect values, categorical feature handling, outliers handling, feature extraction, feature manipulation.

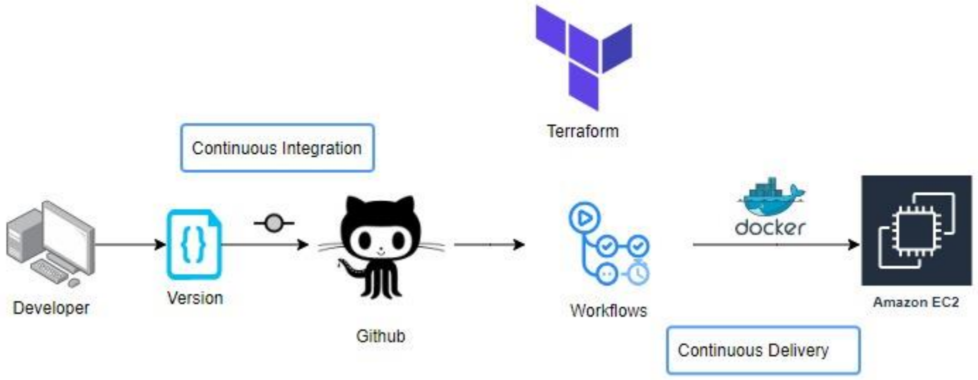
Model Builder:

In this component of project, feature selection has been done and a model is created which performs the best among all the classification Machine Learning algorithms.

Model Pusher:

This component is used after evaluating the best model in order to maintain the performance of model for near future and pushes the best model to system.

Deployment Process:



3.2 Event Log

The system should log every event so that the user will know what process is running internally.

Initial step-by-step description:

1. The system identifies at what level logging is required
2. The system should be able to log each and every system flow
3. Developer can choose logging method. You can choose database logging/ File logging as well
4. System should not hang even after so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3 Error Handling

Errors should be encountered; an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal intended usage.

4 Performance

The ACIP tool is used to predict whether a person earns above or below 50K dollars per annum or not. So, this is made keeping in mind that if it will be used by various governmental/ non-governmental/ private agencies then it is supposed to be as accurate as possible. So that it doesn't mislead authorities. Also, model retraining is very important to further enhance its performance.

4.1 Reusability

The code written and the components used should have the ability to be reused with no problems.

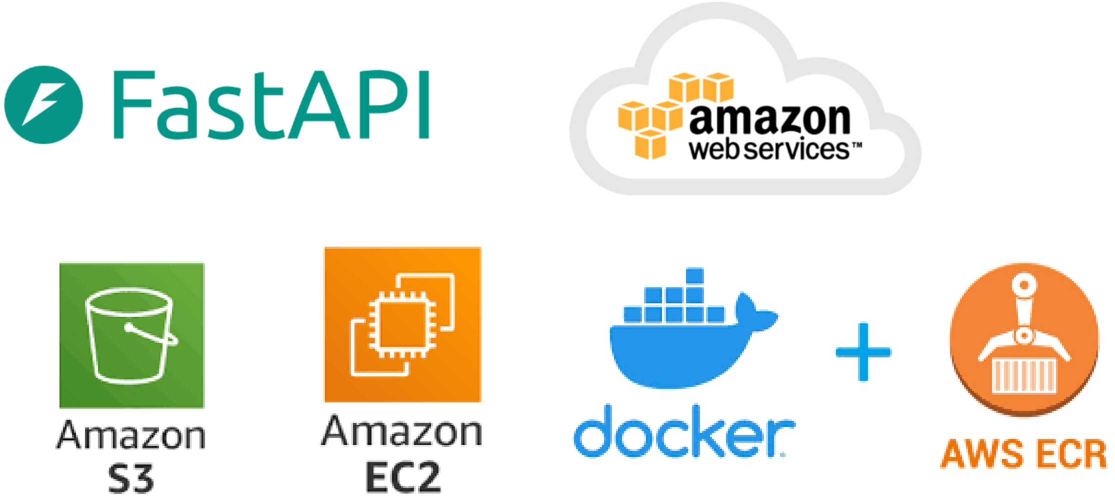
4.2 Application Compatibility

The different components for this project will be using Python as an interface between them, each component will have its own task to perform, and it is the job of Python to ensure proper transfer of information.

4.3 Resource Utilization

When any task is performed, it will likely use all the processing power available to it until finished.

4.4 Deployment



5 Dashboards

Dashboards will be implemented to display and indicate certain KPIs and relevant indicators for the unveiled problems that if not addressed in time would cause catastrophes of unimaginable impact.

As and when, the system starts to capture the historic/ periodic data for a user, the dashboards will be included display charts over time with progress on various indicators or factors.

5.1 KPIs (Key Performance Indicators)

1. Key Performance Indicators of ACIP.
2. Latency or the amount of time the application takes to display results for some specific input.
3. The processing power our application takes to run
4. The memory and RAM our application takes to run on a web server.
5. Comparison of F1 score of model prediction value and actual value.
6. User enters the values for age, capital gain, capital loss, hours per week.
7. The User chooses the work class, education marital status, occupation, relationship, race, sex, country by clicking on one of the available options.
8. Capital Loss and Capital Gain present or not. If present, is it low or high.

6 Conclusion

The ACIP will give the income predictions of a person instantly and has the potential to help various organisations, agencies, companies, etc around the world in various tasks.