



# Adult Census Income Prediction

HIGH LEVEL DOCUMENT DESIGN

25/07/2024

---

Project By : Kushal Jangra

# Document Version Control

DATE ISSUED	VERSION	DESCRIPTION	AUTHOR
25 <sup>th</sup> July'24	1	Initial HLD V1	Ritik Sharma

# 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 this High Level 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 :

- Present all of design aspects and define them in detail
- Describe all user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- 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

### 1. Definitions


<i>Term</i>	<i>Description</i>
ACIP	Adult Census Income Predictor
Database	Collection of the Information monitored by the system
Cloud	A datacenter full of servers connected to the internet performing a service
IDE	Integrated Development Environment
UI	User Interface
Anvil	A Python based UI builder
Heroku	A cloud service

### 2. Product Description

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.

### 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.

#### 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.

#### 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 / nongovernmental / 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

#### 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 :

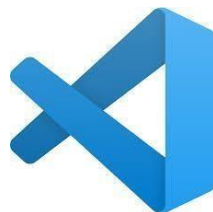
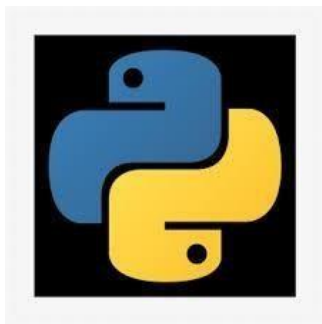
- **Age** : Age of the person whom the income has to be determined.
- **Hours per week** : The number of hours the person work per week.
- **Capital Gain/Loss** : whether the person had any capital gain or loss or none.
- **Country** : country in which the person lives.
- **Educational Level** : highest level of education the person has attained.

- **Marital Status** : whether the person is married or single
- **Occupation** : occupation of the person.
- **Workclass** : working class of the person.
- **Gender** : gender of the person.
- **Race** : race to which the person belongs.

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

## 2.7 Tools used

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



- Google colab and Visual Studio code were used as IDE
- For visualization tasks, matplotlib, seaborn and plotly were used

- Heroku was used for deployment of the model
- Anvil and Flask were used for building the web application and server to run the code
- Apache Cassandra was used to storage and retrieval of data
- Github is used as version control system
- NumPy and Pandas were used to clean and interpret data
- Scikit learn was used to cross validate and compare different models.
- CatBoost Classifier was used to build the final model

#### 2.7.1 Hardware Requirements

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

### 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.

### 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.

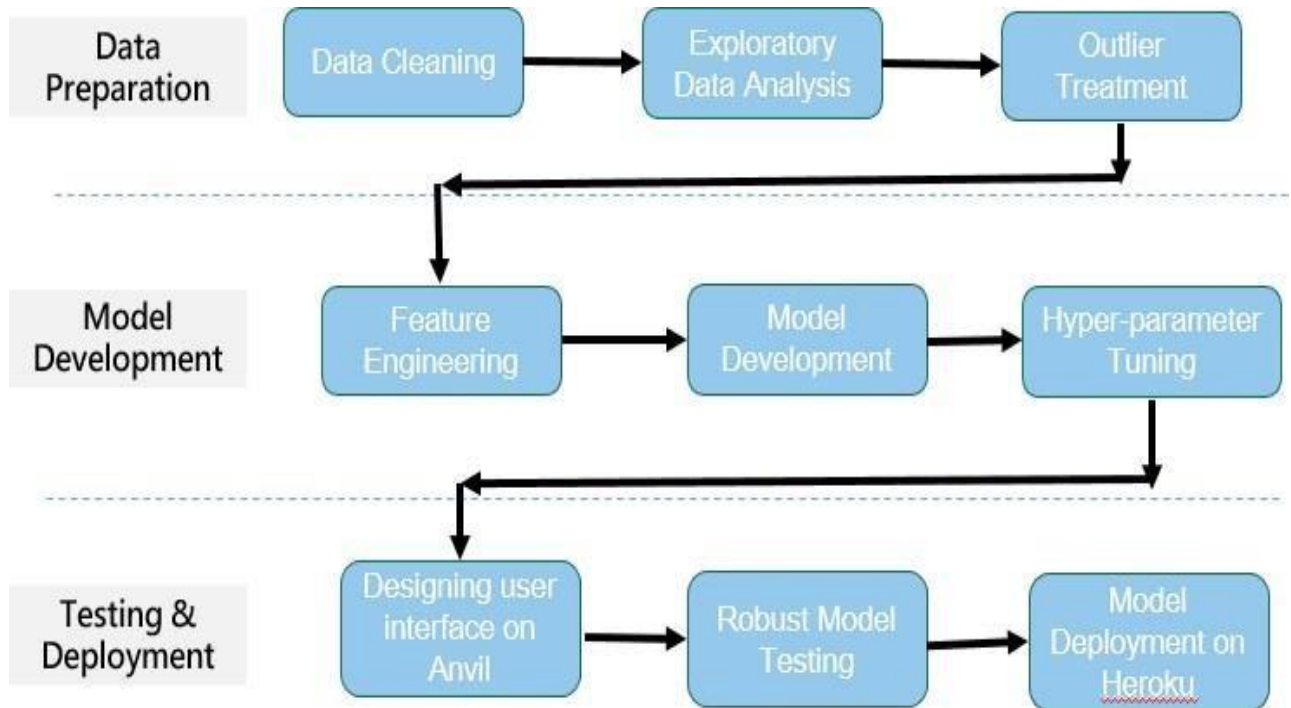
## 3. Design Details

### 3.1 Process Flow

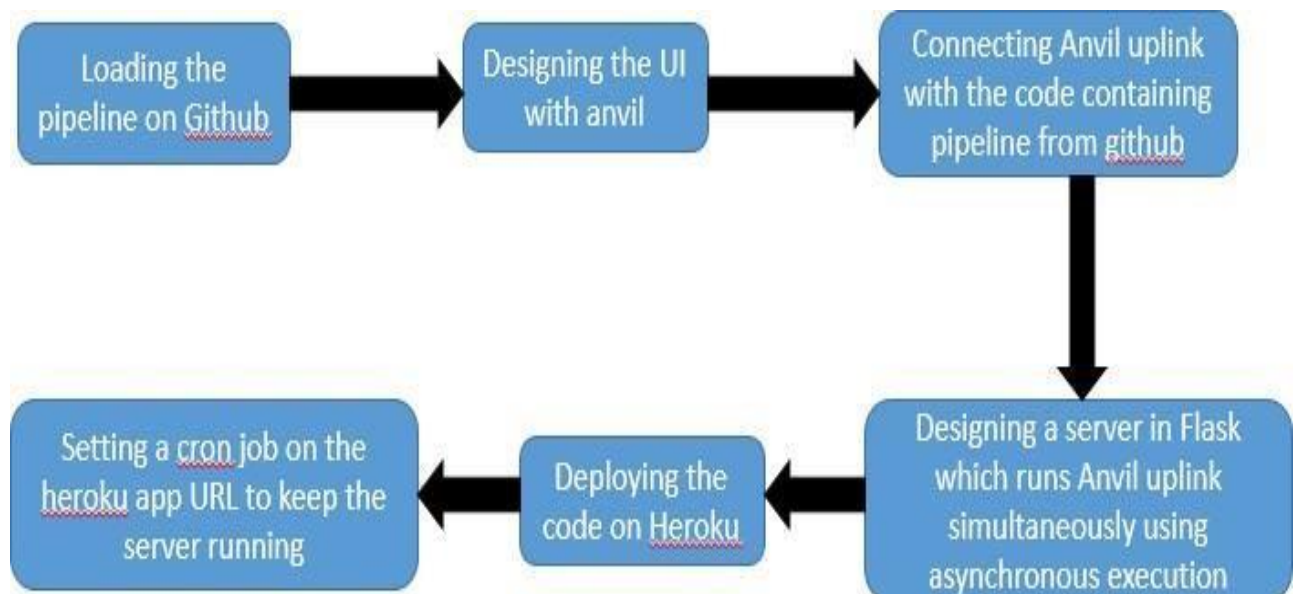
For accomplishment of the task, we will use a trained Machine Learning model.

The process flow diagram is shown below :

## Proposed Methodology



## 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 chose logging method. You can chose 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.

#### 1. Reusability

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

#### 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.

#### 3. Resource Utilization

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

## 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.

### 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.

## 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.