



**Adult Census Income Prediction**

HIGH LEVEL DOCUMENT DESIGN

01.12.2021

**─**

Project By :

Hrishik V

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| DATE ISSUED | VERSION | DESCRIPTION | AUTHOR |
| December 1, 2021 | 1 | Initial HLD V1 | Hrishik V |
|  |  |  |  |
|  |  |  |  |

**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.

# 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 : o Security o Reliability o

Maintainability o Portability o Reusability o Application compatibility o Resource utilization o 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.

# General Description

**2.1** **Definitions**

|  |  |
| --- | --- |
| **Term** | **Description** |
| BB | Bank Bot |
| Rules | The rules which define the flow of converstion |
| BBF | Basic Banking functions |
| CBF | Chat bot functions |
| UI | User Interface |
| Flask | A Python based UI builder |

**2.2** **Product Description**

The BB is a rule based chat bot whose goal is to provide friendly access to banking services.

**2.3** **Problem Statement**

In today’s world banking is also a platform which is almost automated. However many

people face different issues during online banking and also many are not aware of

correct safety measure for their accounts. To create a bot using Natural Language

Processing and deep learning which will be able to give answers for various banking

related issues like your bank id, transaction details, Security services, loan policies, your

account status etc.

**2.4** **Proposed Solution**

Using the rule based chat bot system as a base and designing the front end where the user can input and get the output at the least possible time.

**2.5** **Further Improvements**

The BB can easily be embedded into any internet banking services like net banking, mobile banking or can integrated as chat-based banking service.

This can also be improved further by moving from rule-based approach to a learning based approach where the recorded conversation between customers and bank representatives can be used as a base.

**2.7** **Tools used**

Python programming language and frameworks such as NumPy,

Pandas, Flask, and a few other libraries were used to build the whole model.



* Pycharm is used as theIDE
* Heroku was used for deployment of the model
* 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
* Pandas was used to generate the dataset

**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

* 1. **Constraints**

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

* 1. **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.