

|  |
| --- |
|  |
|  |
| October 25  iNeuron.ai  Authored by: Darshita Paghadal |



## **Census-Income Dataset Analysis** Income Prediction

**Abstract**

The economic well-being of a Nation is highly driven by the income of the residents.

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.

Not only the above, the census data is also used for post census population estimates and projections, economic and social science research, and many other such applications. Hence, the importance of this data and its correct predictions is very clear to us.

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.

|  |  |
| --- | --- |
| CONTENT | |
| 1. INTRODUCTION | 4 |
| 1.1 WHY THIS HIGH-LEVEL DESIGN DOCUMENT | 4 |
| 1.2 SCOPE | 4 |
| 1.3 DEFINCATIONS | 5 |
| 1. GENEREAL DESCRIPTION | 6 |
| 2.1 PROBLEM STATEMENT | 6 |
| 2.2 PRPOSED SOLUTION | 6 |
| 2.3 FURTHER IMPROVEMENT | 6 |
| 2.4 DATA REQUIREMENTS | 6 |
| 2.5 TOOLS USED | 7 |
| 2.6 HARDWARE REQUIREMENTS | 8 |
| 1. DEDIGN DETAILS | 9 |
| 3.1 PROCESS FLOW | 9 |
| 3.2 MODEL TRAINING AND EVALUATION | 10 |
| 3.3 EVENT LOGS | 10 |
| 3.4 ERROR HANDLING | 10 |
| 1. PERFOMANCE | 11 |
| 4.1 REUSABILITY | 11 |
| 4.2 APPLICATION COMPATIBILITY | 11 |
| 4.3 RESOURCE UTILIZATION | 11 |
| 1. CONCLUSION | 13 |

# 1. Introduction

## **1.1 Why this High-Level Design Document?**

The purpose of this High-Level Design (HLD) Document is to add the 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 the design aspects and define them in detail
* Describe the user interface being implemented
* Describe the hardware and software interfaces
* Describe the performance requirements
* Include design features and the 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 the 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.

1.3 Definitions

Term Description

|  |  |
| --- | --- |
| UGV  Database  IDE  A WS | Unmanned Ground Vehicle  Collection of all the information monitored by this system  Integrated Development Environment  Amazon Web Services |

# 2. General Description

## **2.1 Problem Statement**

## 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. I 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. Though the data is quite old, and the insights drawn cannot be directly used for derivation in the modern world, but it would surely help us to analyze what role different features play in predicting the income of an individual.

## Problem statement is we need to predict income (weather it can be more than 50k or less than 50k) based on given different criteria like age, work class, education, occupation, capital gain, capital loss data, working hours per day.

## **2.2 Proposed Solution**

## The solution proposed here to predict data is machine learning based solution using different algorithms. Different algorithms are tried and experimented on given data set and find proposed solution. As there is low amount of data available, decision tree-based algorithm works better. In case of large amount of data, we need to find other solution.

## **2.3 Further Improvement**

## Having small amount of model prediction can be underfitted so in a further improvement case, we need to feed more data to our model and need better training to generate better solution.

## Other improvement, neural-network can be implemented for better prediction.

## **2.4 Data Requirements**

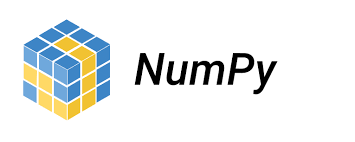
## Data requirements completely depends on a problem statement.

## We need dataset to train a model.

## Dataset should be large enough to train model considering all criteria.

## **2.5 Tools Used**

## Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Seaborn, Flask and Matplotlib, Visual Studio Editor are used to build the whole model.



# 



## Visual Studio is used as IDE.

## For visualization of the plots, Matplotlib, Seaborn and Plotly are used.

## Scikit Learn for atomized pipelines creation and generate model file.

## Flask library is used to generate a web application for user customized prediction through web interface.

## AWS is used for deployment of the model.

2.6 Hardware Requirements

## In summary, to get started with machine learning, you will need a computer with a powerful CPU and/or GPU, a minimum of 8GB of RAM, and large storage capacity. You will also need to install Python and various machine learning libraries and frameworks.

# 3. Design Details

3.1 Process Flow

3.2 Model Training and Evaluation

3.3 Event log

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

## Initial Step-By-Step Description:

## The System identifies at what step logging required

## The System should be able to log each and every system flow.

## Developer can choose logging method. You can choose database logging/ File logging as well.

## System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.4 Error Handling

## Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

# 4. Performance

## Proposed solution is providing 85% accuracy to predict income will be more than 50k or less than based on the input data. Also, model retraining is very important to improve the 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 the Python to ensure proper transfer of information with prediction pipelines.

4.3 Resource Utilization

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

# 5. Conclusion

## Though the data is quite old, and the insights drawn cannot be directly used for derivation in the modern world, but it would surely help us to analyze what role different features play in predicting the income of an individual.