**Low Level Design (LLD)**

Adult Census Income Prediction

By :- Deepak yadav

**Abstract**

The population of the world is going to increase day by day. but job opportunity goes to decrease. That time we want to do some prediction like who is earning more, what is his/her education background, income background,..etc. Therefore I had made a model for the predict the income of the person who earning more that 50K and less than 50K based on some data adult census income data set.

# Project overview

|  |  |
| --- | --- |
| **Project Title** | **Adult Census Income Prediction** |
| **Technologies** | **Machine Learning Technology** |
| **Domain** | **Finance** |

# Problem Statement:

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.

.

# Approach:

The classical machine learning tasks like Data Exploration, Data Cleaning,

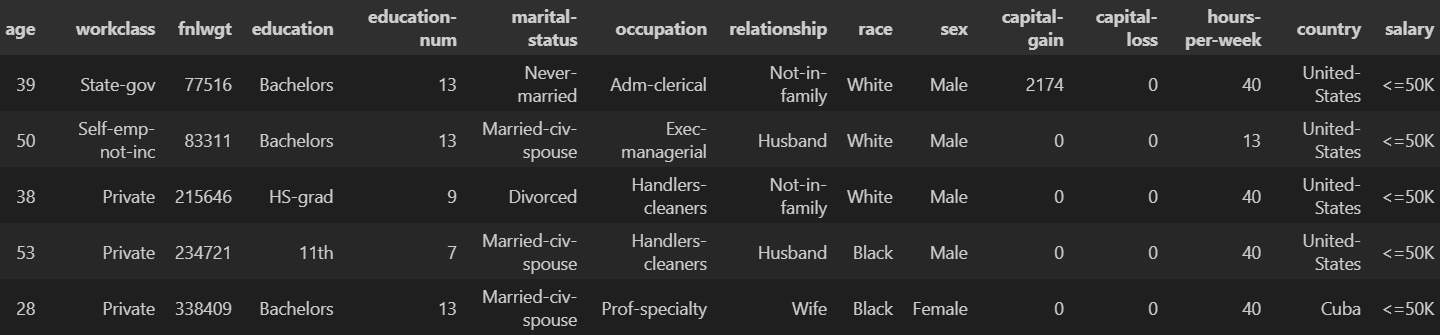
Feature Engineering, Model Building and Model Testing. Try out different machine

learning algorithms that’s best fit for the above case.

## **Adult census income prediction dataset overview**

Consists of 15 different tables. Age has age of the of people ,education columns has education information of the person like that there are 15 different columns and all are have some specific information. We need to predict the salary columns info.

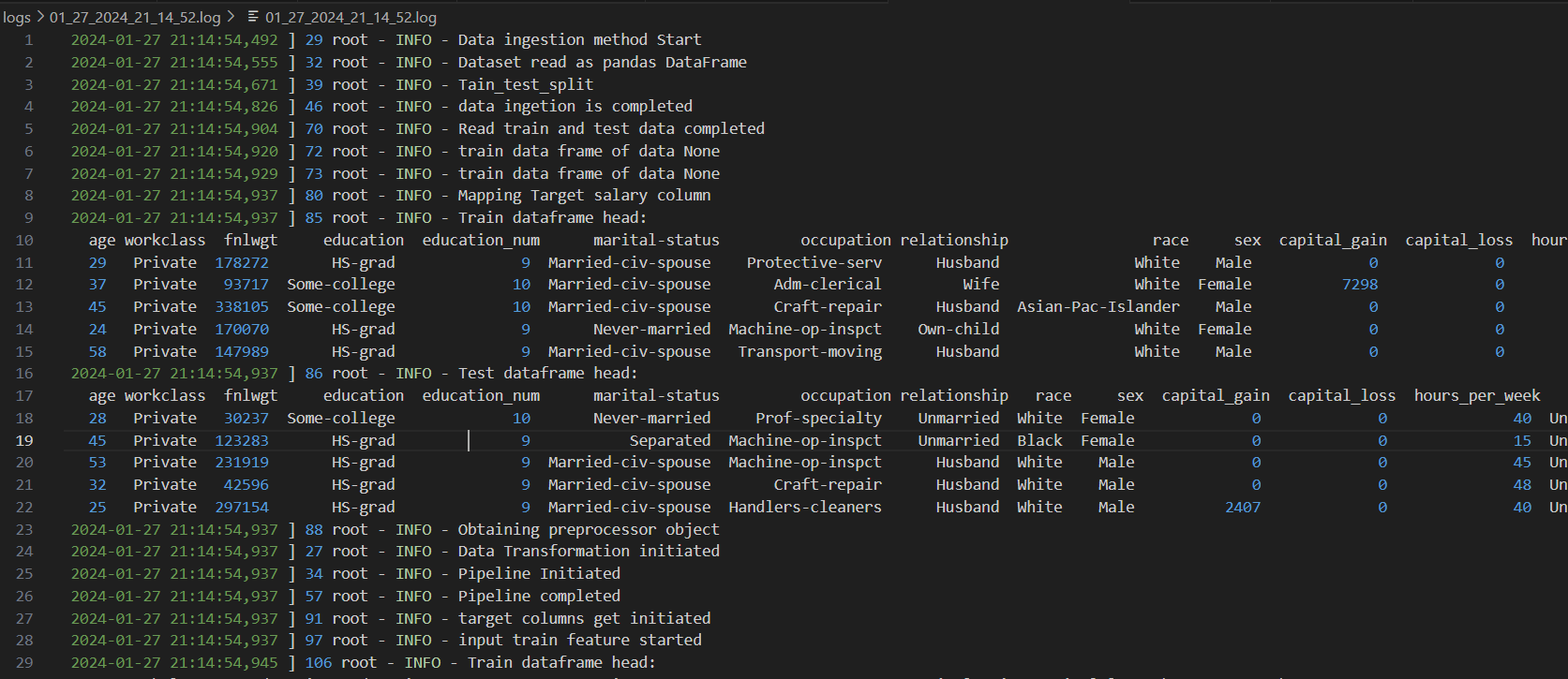
There has 32561 entities into the dataset we select 30% data for the test.



## **Logging**

We should be able to log every activity done by the user.

* The System identifies at what step logging required
* The System should be able to log each and every system flow.
* Developers can choose logging methods. You can choose database logging/ File logging as well.
* System should not be hung even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.
* Log look like this.

****

# Exception

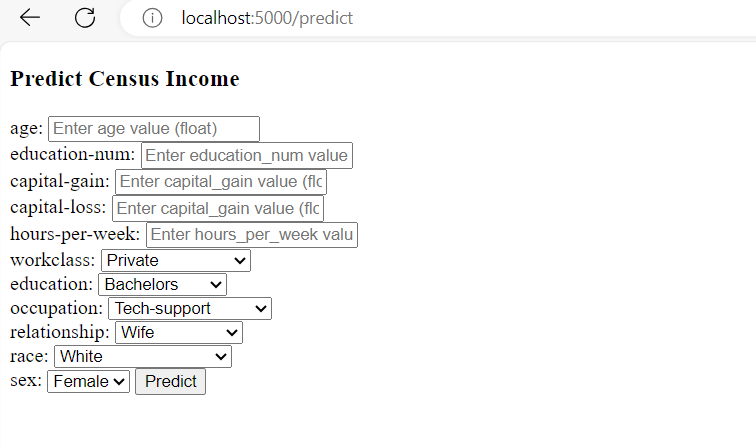
We should be able to use exception in every activity done by the user.

* The System identifies at which step exception occur in the code
* The System should be able to find error at every system flow.

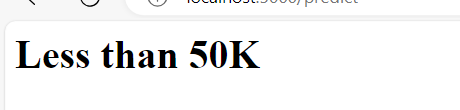
# This is the url page:-



# This is the input page on url

****

**This is the output page**

****

**Deployment**

1. AWS

\

# Technology stack

|  |  |
| --- | --- |
| **Front End** | HTML |
| **Backend** | Python flask |
| **Data set** | Kaggle/ineuron |
| **Deployment** | AWS |