# Census Income Prediction using Deep Learning
# Project Overview

## Business Context

A census as the total process of collecting, compiling, and publishing demographic, economic, and social data pertaining to a specific time to all persons in a country or delimited part of a country. As part of a census count, most countries also include a census of housing. It is the process of collecting, compiling and publishing information on buildings, living quarters and building-related facilities such as sewage systems, bathrooms, and electricity, to name a few.

## Possible Uses of Census Information

| Census Information | Potential Uses |
|---|---|
| Total Population Size | When two or more census counts are compared for the same location, planners can determine if locales are increasing or decreasing in size. |
| Age | Used to help identify segments of the population that require different types of services. |
| Sex | Sex ratios can be calculated by 5-year age groups to crudely observe migration, especially among the working age cohorts. |
| Marital Status | Used to provide insights into family formation and housing needs. |
| Household Composition and Size | Used to help determine housing needs for related and unrelated households. |
| Educational Attainment and Literacy | Used to provide information on the educational skills of the work force. These measures also help planners select the best strategies to communicate with residents. |
| Location of Residence and Place of Prior Residence | Helps assess changes in rural and urban areas. Place of prior residence helps to identify communities that are experiencing in- or out-migration. |
| Occupation and Labor Force Participation | Helps to provide insights into the labor force of a given locale. The information can be used to develop economic development strategies. |
| Living Quarter Characteristics | Can help planners determine housing and community facility needs |

## Data Description

In this project, we will use a standard imbalanced machine learning dataset referred to as the “Adult Income” or simply the “adult” dataset.

The dataset is credited to Ronny Kohavi and Barry Becker and was drawn from the 1994 United States Census Bureau data and involves using personal details such as education level to predict whether an individual will earn more or less than $50,000 per year.

The dataset provides 14 input variables that are a mixture of categorical, ordinal, and numerical data types. The complete list of variables is as follows:

- Age.
- Workclass.
- Final Weight.
- Education.
- Education Number of Years.
- Marital-status.
- Occupation.
- Relationship.
- Race.
- Sex.
- Capital-gain.
- Capital-loss.
- Hours-per-week.
- Native-country.

The dataset contains missing values that are marked with a question mark character (?).

There are a total of 48,842 rows of data, and 3,620 with missing values, leaving 45,222 complete rows.

There are two class values ‘>50K‘ and ‘<=50K‘, meaning it is a binary classification task. The classes are imbalanced, with a skew toward the ‘<=50K‘ class label.

- ‘>50K’: majority class, approximately 25%.
- ‘<=50K’: minority class, approximately 75%.#

## Data Source

http://www.census.gov/ftp/pub/DES/www/welcome.html

## Tools/Libraries

- Python
- scikit-learn(machine learning library)
- h2o.ai

## Aim

Census Salary Prediction where we have to classify between >50K <=50K.

## How Does it help

- Real Estate Demands
- Basic Amenities
- Fulfilling Infrastructure Demands

## Modular code folder structure

```
input
  |__adult.data
  |__adult.test
  |__adult_data_cleaned_test.csv
src
  |__engine.py
  |__ML_pipeline
            |__get_factors.py
            |__grid_search.py
            |__impute.py
            |__model_building.py
            |__train_valid_split.py
            |__utils.py
            |__validate_model.py

lib
  |__Dataset links.txt
  |__notebooks
            |__Census_data_prediction.ipynb
  |__resources
            |__1_business_objective.docx
            |__2_data_description.docx
            |__3_perceptron_classifier.docx
            |__4_activation_function.docx
            |__5_difference_ml_dl.docx
            |__6_h20_ai_document.docx


output
  |__DeepLearning_model_4
```

Once you unzip the modular_code.zip file you can find the following folders within it.

1. input
2. src
3. output
4. lib

1. The input folder contains all the data that we have for analysis.
2. The src folder is the heart of the project. This folder contains all the modularized code for all the above steps in a modularized manner. It further contains the following.
    a. ML_pipeline
    b. engine.py

The ML_pipeline is a folder that contains all the functions put into different python files which are appropriately named. These python functions are then called inside the engine.py file

3. The output folder contains all the models that we trained for this data saved. These models can be easily loaded and used for future use and the user need not have to train all the models from the beginning.
4. The lib folder is a reference folder. It contains two folders namely
    a. notebooks
    b. resources

    The notebooks folder contain the jupyter notebook that you see in the videos.
    The resources folder contain all the resources that are used in the videos.

## Project Takeaways

- Understanding the problem statement
- Importing the dataset and importing libraries
- Performing basic EDA
- Data cleaning Imputing the null values and if required filling them using appropriate methods.
- Checking data distribution using statistical techniques
- Checking for outliers and how they need to be treated as per the model selection.
- Using python libraries such as matplotlib and seaborn for better and advanced visualizations.
- Splitting Dataset into Train and Test using Stratified Sampling
- Feature Engineering for better decision making by a model
- Training a model using Vanilla DNN
- As per the result, research for other network architectures
- Understanding Class Imbalance Problem and whether any solution needed to tackle it
- Doing Cross Validation to see if the model is overfitting and whether results are somewhat constant.
- Tuning hyperparameters of models to achieve optimal performance and their effect in the results.
- Making predictions using the trained model.
- Gaining confidence in the model using metrics such as Accuracy,Precision,Recall,F1-Score,AUC
- Understanding why Accuracy might be/might not be a good metric to check results
- Selection of the best model based on Feature Importance and the metrics.