# **Census Income Project**

Problem Statement:

This data was extracted from the [1994 Census bureau database](http://www.census.gov/en.html" \t "https://medium.com/@lokeshbisen989/_blank) by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics). A set of reasonably clean records was extracted using the following conditions: ((AGE>16) & (AGI>100) & (AFNLWGT>1) && (HRSWK>0)). The prediction task is to determine whether a person makes over $50K a year or less then**.**

# Exploratory Data Analysis (EDA):

Through the head().method we can observed 5 rows. Here we can see there are many features which are in categorical so we have to convert into numeric in order to feed our data to machine learning model. As per our problem statement we have to predict the income is below 50k or beyond. So we can convert into binary and build classification model.

**The data-set contains**32560**rows and**14**features + the target variable (**Income**)**. 6 are integers and 9 are objects.

As per description we have seen in ‘count’ there is no null values in the dataset. And, we can also see ‘mean’ it shows variation among the features and values are on different scales so we have to scale the features in similar scale.

As we have checked statistical descriptions which shows only numeric data and dataset contains categorical values as well as. Therefore we have to check null values.

As per above we can not see any null values in our dataset. But in the 2 features Native\_country and Occupation there is ‘?’ empty which will be consider as a null values. Hence we have to replace with authentic values.

We can see “Native\_country” contains 583 missing values and “Occupation” contains 1816 missing values as count is huge. Although we can use simple imputer or other method like KNN imputer to fill null values. But if we will fill it probably it may biased towards the single variable. However, we have large data so we can remove it using ‘Drop’ function.

**Let’s take a more detailed look from Data Visualizations.**

**Data Preprocessing:**

Encoding Categorical Variables : As we have many features contains categorical variable so we are using pandas get\_dummies function to convert into numeric and 2 variables “Income” and “Sex” columns converting into binary using label encoder.

As we have to scale only 4 features so I am stored it in a variable. And using standard scaler. later all 4 scale features replaced. Now our dataset has been scaled and ready for the model building.

# Model Building:

Now we will train several Machine Learning models and compare their results. Note that because the dataset does not provide labels for their testing-set, we need to use the sklearn train test split and divide into test and train. Before that we have to divide into dependent and independent variables.

Data is now divided in independent and dependent.Our data set divided into train and test. Now we will continue with model building.

Now its time to train out model.Our Model has been train now checking the accuracy.

As our data was imbalanced and in this case we have to consider F1-score and there are 2 models giving the scores above 90% and rest of below 90%. In order to check our model is overfitted or not we are checking the cross validation for 2 models which are giving the scores above 90%.

Saving Model .

# **Concluding Remarks :**

We started the our project to import various libraries and imported the dataset from GitHub. Observing the many important points like problem type and how many columns contains int ,float and object values. As per statistic observations we found huge variations among the features and we have used standard scaler to scale the variables. Besides this, we have identified there are 2 columns where instead the NaN it is in “?” so we deleted such rows. During this process we used seaborn and matplotlib to do the visualizations and converted categorical features into numeric using label encoder and pandas get\_dummies function. Afterwards we started training different different machine learning models, picked one of them (Xgboost) and applied cross validation on it and we tried to tune model using hyperparameter tuning . To conclude, There are many other ways also to improve the model accuracy like doing a more extensive feature engineering, by comparing and plotting the features against each other and identifying and removing the noisy features, Along with resampling the data in case of imbalance or more extensive hyperparameter tuning on several machine learning models .