**CENSUS INCOME ANALYSIS**

**Introduction:**

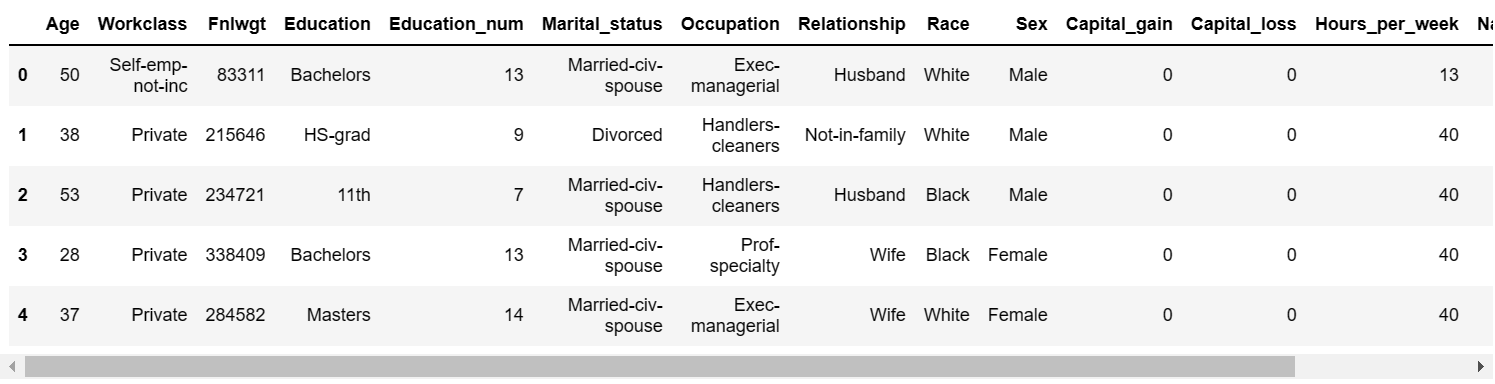
**Description**

In this project, initially we preprocess the data and then develop an understanding of different features of the data by performing exploratory analysis and creating visualizations. Further, after having sufficient knowledge about the attributes, performed a predictive task of classification to predict whether an individual makes over 50K a year or less by using different Machine Learning Algorithms.

**Dataset**

The "Census Income" data was extracted from the [1994 Census bureau database](http://www.census.gov/en.html) by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics).

I obtained the dataset from the below link. <https://raw.githubusercontent.com/dsrscientist/dataset1/master/census_income.csv>



## Lab Environment

## Jupyter notebook

## Tasks Done

1. **Data Preprocessing:**
2. Replaced all the missing values with NA.
3. Removed all the rows that contain NA values.
4. Removed all whitespaces from the columns.
5. **Data Manipulation:**

The original dataset consisted of the following 15 columns.

* Age(integer): the age of each entry.
* Workclass(string): the workclass of each entry.Option include selfemployee,private,local government,federal government.
* Fnlwgt(integer):the final weight of the row.its related to how many unique entries in row characterisitics.
* Education (string): The highest form of education for each entry. Options include: HS-grad, Masters, 10th
* Education.num (integer) :The total number of years spent in education for each entry
* Marital.status (string) : The marital status of each entry
* Occupation (string) :The occupation of each entry
* Relationship (string) :The family-relationship held by each entry. Options include: Mother, Father, Not-in-Family, Unmarried
* Race (string) :The race of each entry
* Sex (string) : The gender of each entry
* Capital.gain (integer) : The capitol gain of each entry
* Capital.loss (integer) :The capitol loss of each entry
* Hours.per.week (integer) : The hours per week worked by each entry
* Native.country (string) : The native country of each entry
* Income (string) : The annual income of each entry as defined categorically as less than or equal to $50,000 (<=50K) and greater than $50,000 (>50K)

**3.Data Visualization:**

a) Built a count-plot for the “workclass” column and filled the bars according to the column.  
 b) Built a distplot for the “Hours\_per\_week” column with density .

c) Built a distplot for the “capital.gain” column with density .

d) Built a heatmap of correlation between the dataset.

e) Divide a dataset as independent and dependent data.

**4.** Encoding and skewing techniques:

Building a dataset encoding a data in to integer and find skew.then finding a outlier a dataset.

5. Scaling Techniques:

* Build a standard scaler to transform the dependent and independent variables to training and testing data

**6.OverSampling** Techniques:

* **Pre-processing pipeline using import SMOTE.**

**7.Machine learning models:**

**a)** Built a RANDOM FOREST CLASSIFIER model as follows:

i) Divided the dataset into training and test sets.  
ii) Built a linear model on the test set where the dependent variable and independent variable .

iii) Predicted the values on the train set and found the error in prediction.  
iv) Calculated the R2 score and random state.

**b)**  Built a LOGISTIC REGRESSION model as follows:

i) Divided the dataset into training and test sets.  
ii) Built a logistic regression model where the dependent variable and independent variables.  
iii) Predicted the values on the test set.  
iv) Calculated the accuracy score,confusion matrix and classification report.

**c)** Built a GRADIENT BOOSTING CLASSIFIER model as follows:

i) Divided the dataset into training and test sets.  
ii) Built a random forest classifier model where the dependent variable and independent variables.  
iii) Predicted the values on the test set.  
iv) Calculated the accuracy score,confusion matrix and classification report.

**d)** Built a SUPPORT VECTOR MACHINE model as follows:

i) Divided the dataset into training and test sets.  
ii) Built a support vectore machine classifier model where the dependent variable and independent variables.  
iii) Predicted the values on the test set.  
iv) Calculated the accuracy score,confusion matrix and classification report.

**e)** Built a DECISION TREE CLASSIFIER model as follows:

i) Divided the dataset into training and test sets.  
ii) Built a decision tree classifier model where the dependent variable and independent variables.  
iii) Predicted the values on the test set.  
iv) Calculated the accuracy score,confusion matrix and classification report.

**f)** To checking the models using cross validation to finding a best score.

# **8. HYPER PARAMETER TUNING:**

Building a model selection using GRIDSEARCH CV Method to finding a best score of random forest classifier of parameters are

‘criterion': 'gini',

'max\_depth': 8,

'max\_features': 'auto',

'n\_estimators': 200

accuracy\_score=84.8.

### **Conclusions**

The goal of this project was to determine whether a person makes over $50,000 a year (in 1994 U.S. Dollars). Exploratory data analysis revealed that hours worked per week, sex,education level, capital gains, capital losses, and age were all associated with income.

Using these quantitative data (once biological sex was mapped to 1 or 0), several supervised machine learning models were built to predict income level (which was entered in the data as either "<=50K" or ">50K").

* The primary criterion for model performance was precision,f1score,recall and support.the secondary criterion was accuracy.
* A support vector machine classification model with a precision of 77.2% and an accuracy of 83.9% was the most successful of the models built. While this model has a low false positive rate, its false negative rate is higher; the model suffers from low recall. This is a decided drawback of the model.
* A random forest model with an accuracy of 82.6% is an attractive option. While its precision is noticeably lower at 68.5%, its recall is higher, meaning that it correctly classifies more high-income records than the support vector machine classifier did. While the random forest model does not meet my primary criterion for model success, it is an attractive option if fewer false negatives are a priority.
* A Gradient boosting model with an accuracy of 87.2% is an attractive option. While its precision is noticeably lower at 87.2%, its recall is higher, meaning that it correctly classifies more high-income records than random forest model does not meet my primary criterion for model success, it is an attractive option if fewer false negatives are a priority.
* A decision tree classifier with an accuracy of 85% is an attractive option. While its precision is noticeably lower at 85%, its recall is higher, meaning that it correctly classifies lesser than the gradient boosting did. While the random forest model does not meet my primary criterion for model success, it is an attractive option if fewer false negatives are a priority.