# Census Income Dataset

**Problem Definition:**

The dataset named Adult Census Income is available in kaggle and UCI repository.

*Link:* [*https://www.kaggle.com/uciml/adult-census-income*](https://www.kaggle.com/uciml/adult-census-income)*.*

This 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). A set of reasonably clean records was extracted using the following conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1) && (HRSWK>0)).

*The prediction task is to determine whether a person makes over $50K a year*.

Description of fnlwgt (final weight):

The weights on the Current Population Survey (CPS) files are controlled to independent estimates of the civilian noninstitutional population of the US. These are prepared monthly for us by Population Division here at the Census Bureau. We use 3 sets of controls. These are:

1. A single cell estimate of the population 16+ for each state.
2. Controls for Hispanic Origin by age and sex.
3. Controls by Race, age and sex.

We use all three sets of controls in our weighting program and "rake" through them 6 times so that by the end we come back to all the controls we used. The term estimate refers to population totals derived from CPS by creating "weighted tallies" of any specified socio-economic characteristics of the population. People with similar demographic characteristics should have similar weights. There is one important caveat to remember about this statement. That is that since the CPS sample is actually a collection of 51 state samples, each with its own probability of selection, the statement only applies within state.

Using the python language, I have attempted to fit a few machine learning models and

find the best model to describe the data.

**Data Analysis:**

Initially I imported the required libraries and then imported the data and visualized it.

Data required cleaning, transformation, integration. There were null values in dataset, and it had redundant columns as well. The missing values were shown as '?'. They were replaced by NAN values.

The dataset had 15 columns. The missing values were only present in the categorical columns. Several predictive supervised models like KNN or other methods could be used to predict those values and impute them.

I converted the problem to a dichotomous classification problem and hence the dependent column ‘income’ which is to be predicted has been replaced with 0 and 1. One redundant column ‘education.num’ was removed, which was an ordinal representation of ‘education’.

After removing the unnecessary data points and redundant attributes, it was necessary to select the set of attributes really contributing to the prediction of the income.

To find the correlation between all the other attributes with the dependent variable a heatmap was created, attribute ‘fnlwgt’ had negative correlation with the dependent variable, so it should be dropped.

A statistical test called ‘Point Biserial Correlation’ was also used to measure the relationship between a binary variable x and a continuous variable y. It works on the same concept of a correlation coefficient. The 'fnlwgt' had a negative correlation with 'income' and hence that column was dropped. For feature selection, except ‘fnlwgt’ all columns were selected.

Now to measure the correlation between 2 categorical variables, chi-square estimate was used. The chi-square gives a contingency table and calculates the p-value and the chi-square estimate. The hypothesis of the chi-square test is

H0: variable not related.

H1: variables related

I had set the value of alpha as 0.01. If the p-value is less than alpha, it will reject H0 and hence the variables are related.

According to chi-square estimate all the categorical variables contribute to 'income'. Hence, all of them were kept.

Fitting the categorical columns into machine learning models without encoding them would cause a problem, and hence they were encoded by using get\_dummies. It gave the dummy variables. It simply gets all your data on the same scale.

Now, there were 103 columns. Fitting this huge set of attributes is not recommended and hence the next step was feature selection. Also, since almost all variables are either 0 or 1, we normalize the continuous variables to be between 0 and 1.

After using the min max normalization which by default scales all the variables between 0 and 1, chi square test was used for feature selection. According to chi square test, only 65 attributes contribute to the dependent variable. Continuous variables were added to the set of selected attributes.

So, there were 103 attributes including numerical variables and after feature selection, there were 65 attributes.

### **Exploratory Data Analysis (EDA):**

It is a good practice to understand the data first and try to gather as many insights from it and EDA is done for the same. Exploratory data analysis (EDA) is used by to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

All other parameters were analyzed with respect to dependent variable ‘income’.

* For ‘Income with respect to people’ a graph and pie chart was drawn.

There were total of 30162 observations out of which 7508 (24.9%) people earning more than 50K and 22654 (75.1%) earning less than or equal to 50K.

The percentage division was clearly understood by the pie chart.

* In ‘Income with respect to sex’ it was seen that

5% women are in >50 category and the rest are 95% men

38% women are in <=50 and the rest are 62% men

69% of all men earn <=50K while only 31% earn more than 50K

Only 10% of all women in the sample earn >50K while the rest 90% earn less than 50K.

* ‘Frequency distribution of income with respect to age’ and ‘Frequency distribution of workclass with respect to income’ were also done.
* According to ‘Histogram of hours per week’ around 18K people work between 35 to 50 hours a day.
* ‘Age distribution for income’ shows that

Maximum people around age 23 earn <=50K.

Maximum people around age 37 earn >50K.

**Building Machine Learning Models:**

As seen in the first graph i.e. ‘Income with respect to people’, there is a class imbalance problem. The pie chart clearly denotes that more than 50% of the dataset is occupied by one type of observation. Models do not fit well when there is a class imbalance. There are some methods like oversampling, undersampling and mixture. So, to do class oversampling SMOTE (Synthetic Minority Oversampling Technique) is used. It creates new samples along the lines of the existing samples. The Machine Learning models to find the best solution are:

**1.Logistic Regression:**

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.

We can call a Logistic Regression a Linear Regression model, but the Logistic Regression uses a more complex cost function, this cost function can be defined as the ‘Sigmoid function’ or also known as the ‘logistic function’ instead of a linear function.

The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

After fitting the model, we found the model accuracy, and the AUC was 0.902. I generated the confusion matrix and it does somewhat good.

**2.Decision Tree:**

Decision tree is one of the predictive modelling approaches used in statistics, data mining and machine learning. A decision tree is a branched flowchart showing multiple pathways for potential decisions and outcomes. The tree starts with what is called a decision node, which signifies that a decision must be made. From the decision node, a branch is created for each of the alternative choices under consideration.

Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a [non-parametric](https://machinelearningmastery.com/parametric-and-nonparametric-machine-learning-algorithms/) supervised learning method used for both classification and regression tasks.

After fitting the model, the AUC was 0.883.

**3.Naive Bayes:**

Naive Bayes is a simple but surprisingly powerful algorithm for predictive modeling.

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values.

It is called naive Bayes or idiot Bayes because the calculation of the probabilities for each hypothesis are simplified to make their calculation tractable.

A naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable. Basically, it’s “naive” because it makes assumptions that may or may not turn out to be correct.

After fitting the model, the AUC was 0.819.

**4.Random Forest:**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

After fitting the model, the AUC was 0.911.

**Conclusion:**

All the models were compared in the end by plotting all the curves on the graph.

Random Forest performs the best with area of 0.91 under the curve. It covers the maximum area and hence is a best model.

The project is available in my GitHub repository.

*Link:* [*https://github.com/sahil0801/Projects.git*](https://github.com/sahil0801/Projects.git)

Any comment or suggestions for improvement will be helpful.

Thank You!!