**Census Income Project**

**ABOUT THE DATASET**

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)).

Independent variables:

Age

Workclass

Fnlwgt

Education

Education number

Marital status

Occupation

Relationship

Sex

Race

Capital gain

Capital loss

Hours per week

Native country

**DEPENDANT VARIABLE (TARGET VARIABLE)**

Here, the dependant variable is income since we have to figure out the salary status of an individual whether it is more than 50k or not

**Description about the project:**

## Description of fnlwgt (final weight)

The weights on the Current Population Survey (CPS) files are controlled to independent estimates of the civilian non-institutional 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:

A single cell estimate of the population 16+ for each state.

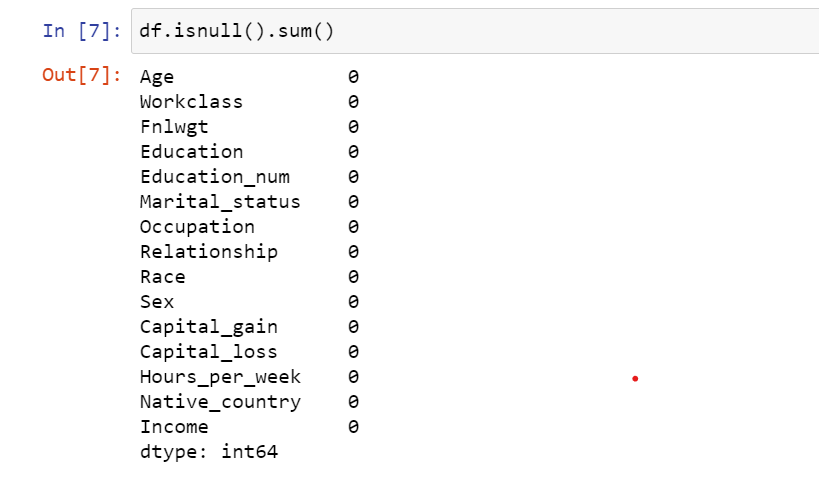
Controls for Hispanic Origin by age and sex.

Controls by Race, age and sex.

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.

**Finding missing values:**

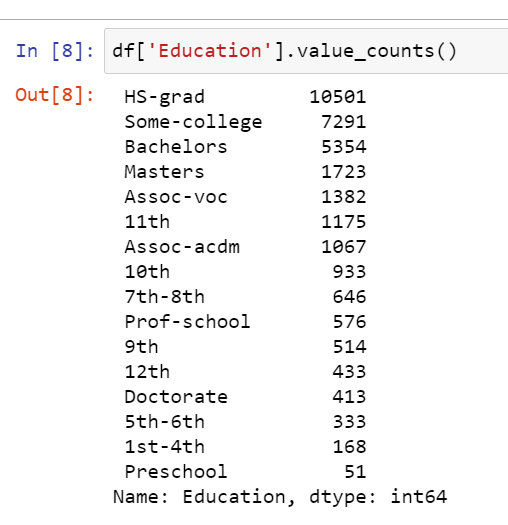
Missing Data can occur when no information is provided for one or more items or for a whole unit. Missing Data is a very big problem in a real-life scenarios. Missing Data can also refer to as NA(Not Available) values in pandas. In DataFrame sometimes many datasets simply arrive with missing data, either because it exists and was not collected or it never existed



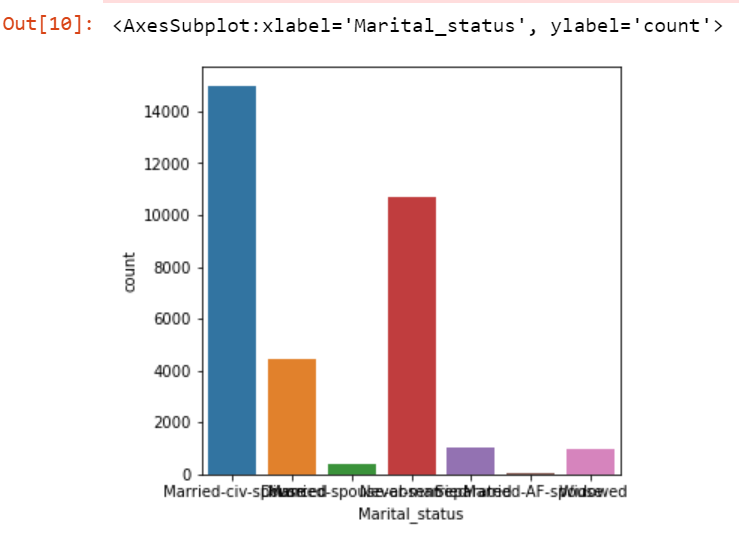
Here, none of the values were missing hence the result came out to be zero

**EDA ( exploratory data analysis):**

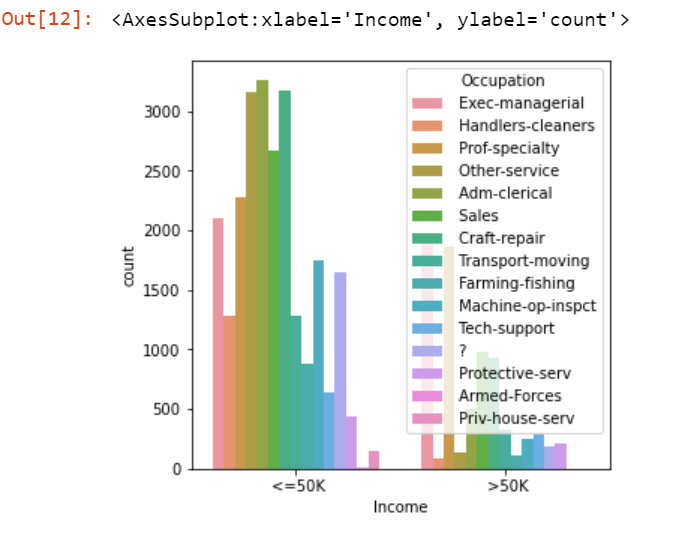
EDA in data science is quite like the service advisor doing a rough inspection of your car, asking a few preliminary questions, setting expectations and then taking the car in for service. It is one of the first things done with the data, so it is a critical phase, as many inferences and consequent actions depend on this exploration



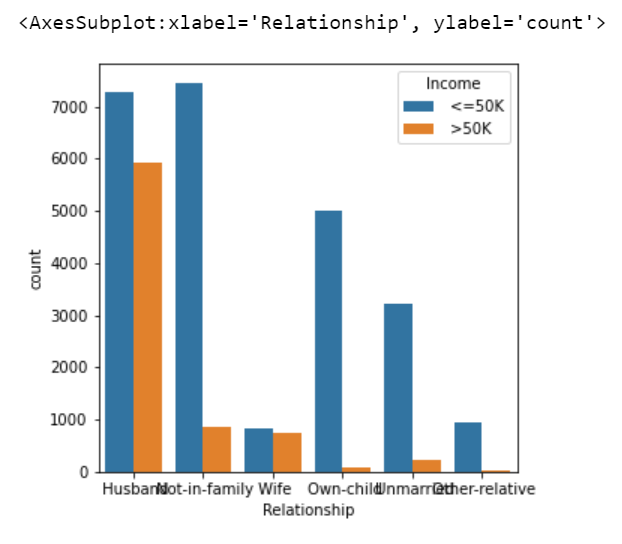
The above data represents value of people been educated upto a specific degree



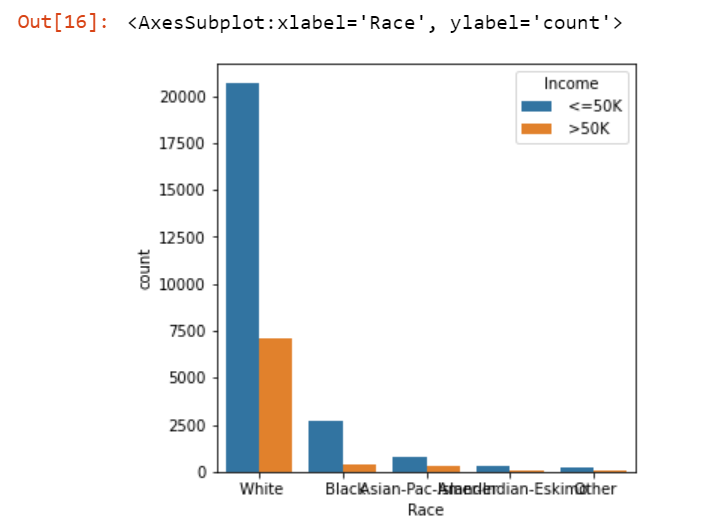
Here, the graphical representation shows the marital status of different people. As we can clearly see that the ratio of married and currently living with spouce is the highest and married and having an affair is the lowest.



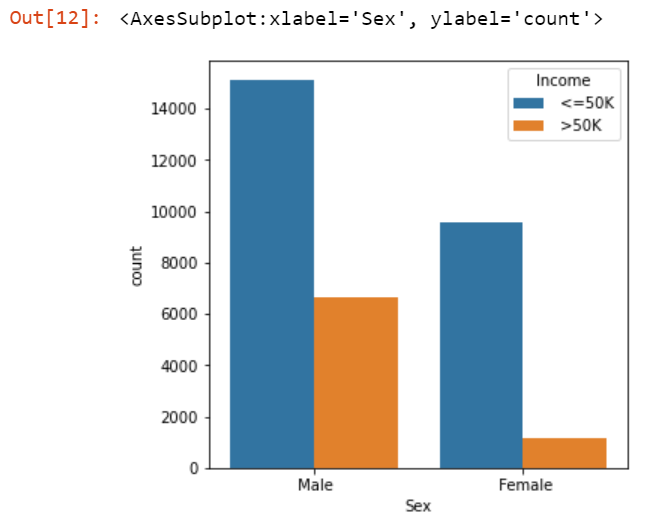
Here, in this graph income is been compared with respect to the occupation they are been lying into.



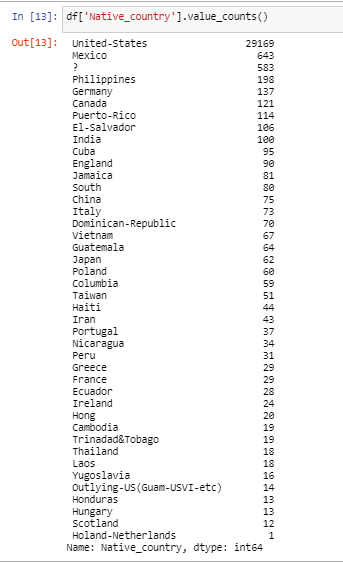
On comparing income with the relationship status we can figure out that highest income which is more than 50k is been carried out by a male who is not living in the family and lowest income is of female who is unmarried and owns a child.



While comparing race with respect to income, we can have an estimated idea that people falling in others category have the least income.



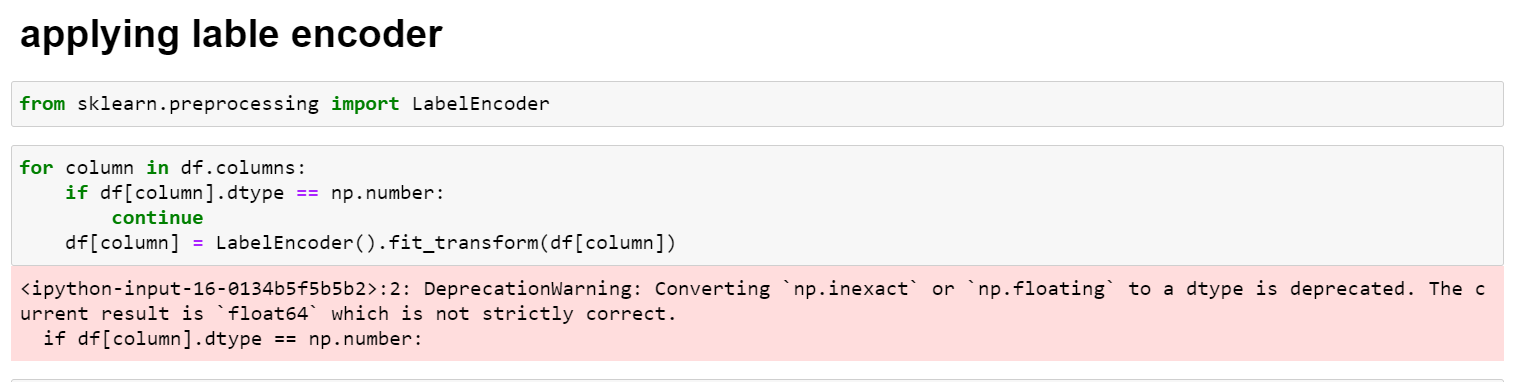
As comared to males, females are having lower income.



Here, we can configure number of individuals in their respective countries.



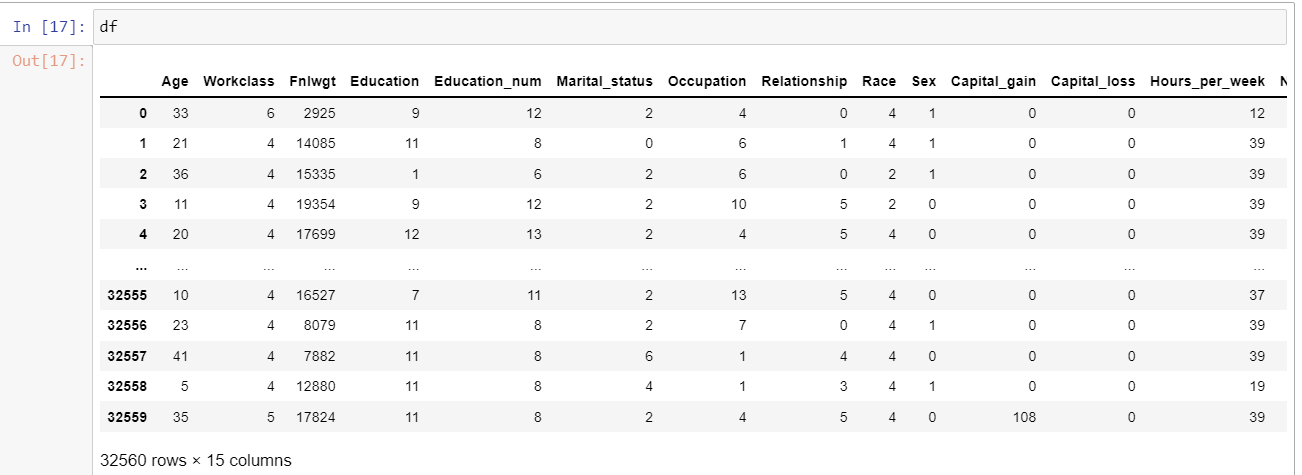
**Label encoder:**

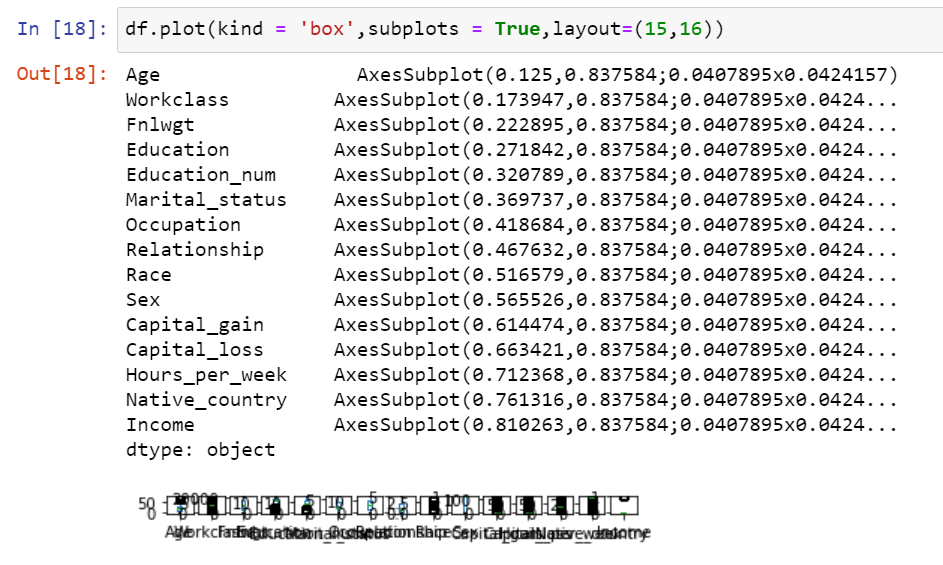


**Label Encoding**

It refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

Our data set after label encoding looks like this:

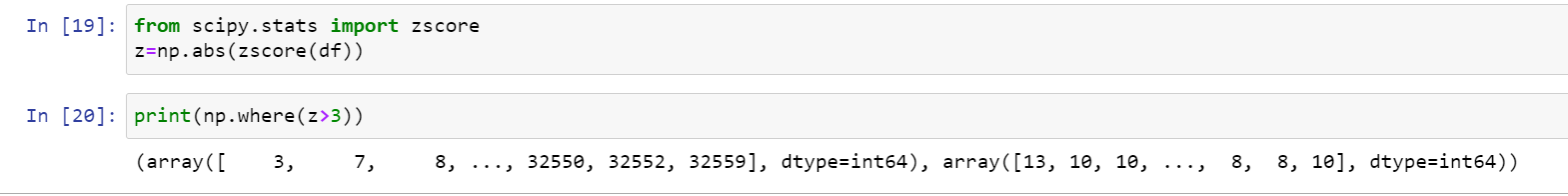




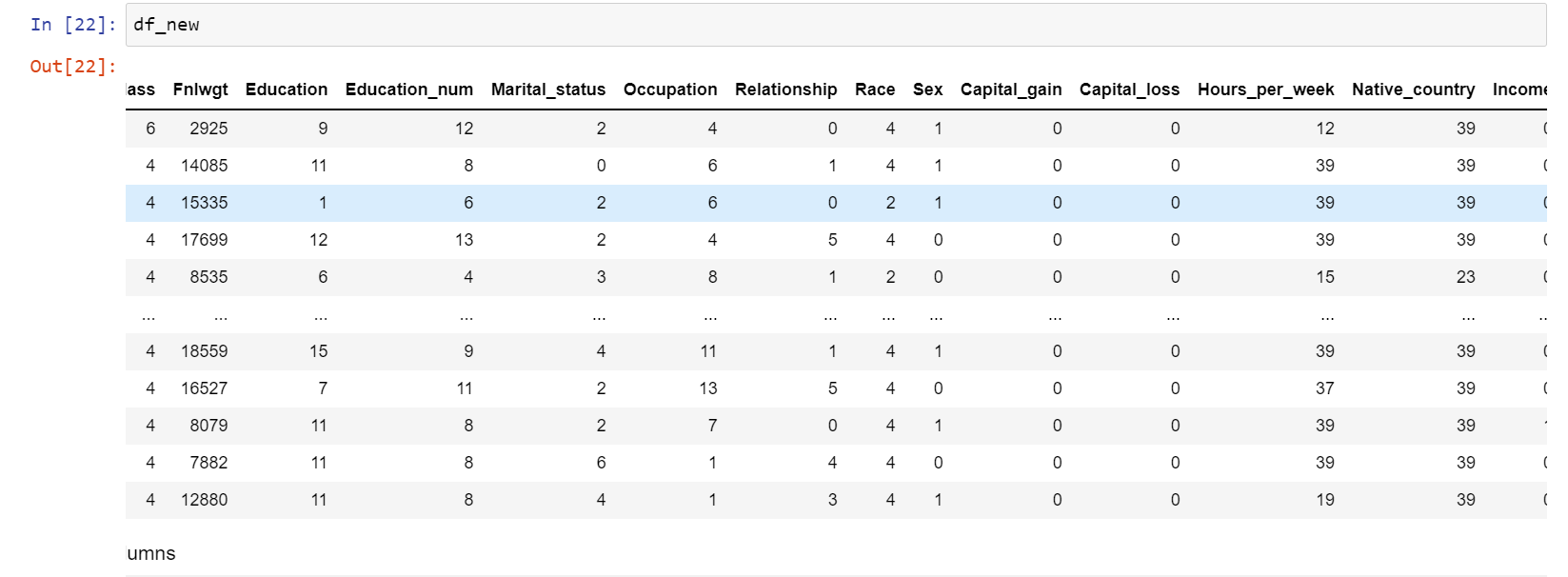
**Outliers**

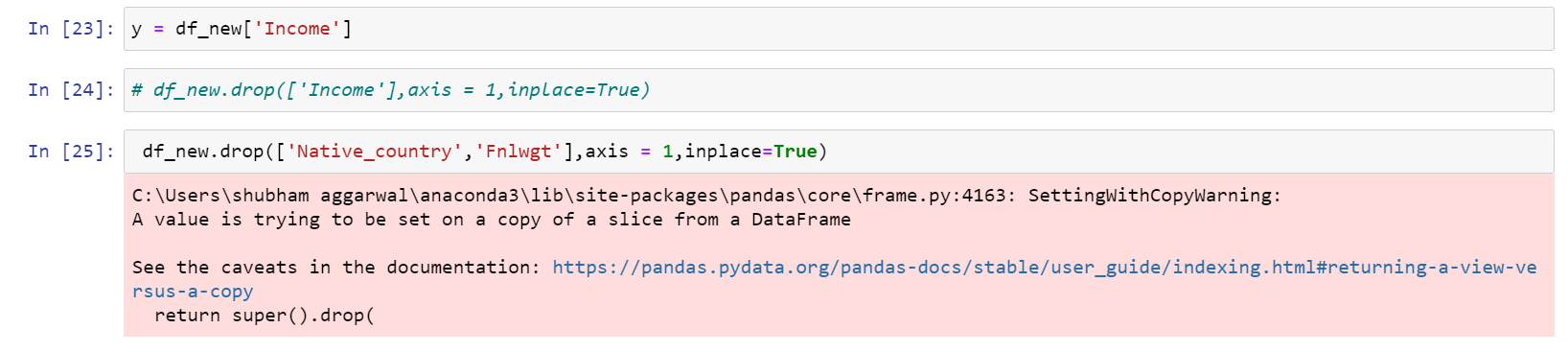
An **outlier** is an object that deviates significantly from the rest of the objects. They can be caused by measurement or execution error. The analysis of outlier data is referred to as outlier analysis or outlier mining.

Identified outliers in the dataset



After removal of the outliers, our dataset turns out this way:



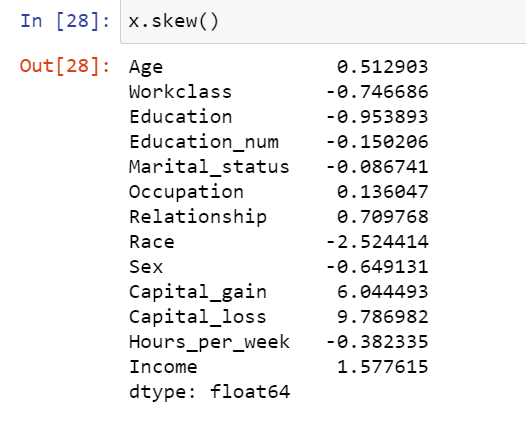


Then we selected our target column and other columns , target column will be represented with y variables

**Skewness**

Skewness is **a measure of the asymmetry of data distribution**. Skewness is an asymmetry in a statistical distribution, in which the curve appears distorted or skewed either to the left or to the right. Skewness can be quantified to define the extent to which a distribution differs from a normal distribution.

Removing the skewness



The skewness is removed from the X named variable not from the Y variable and it was removed using the power transform library which is presented In the sklearn it is important to remove the skewness as it is the part of data cleaning and it also improved the accuracy level of machine learning model

**Standard scaler**

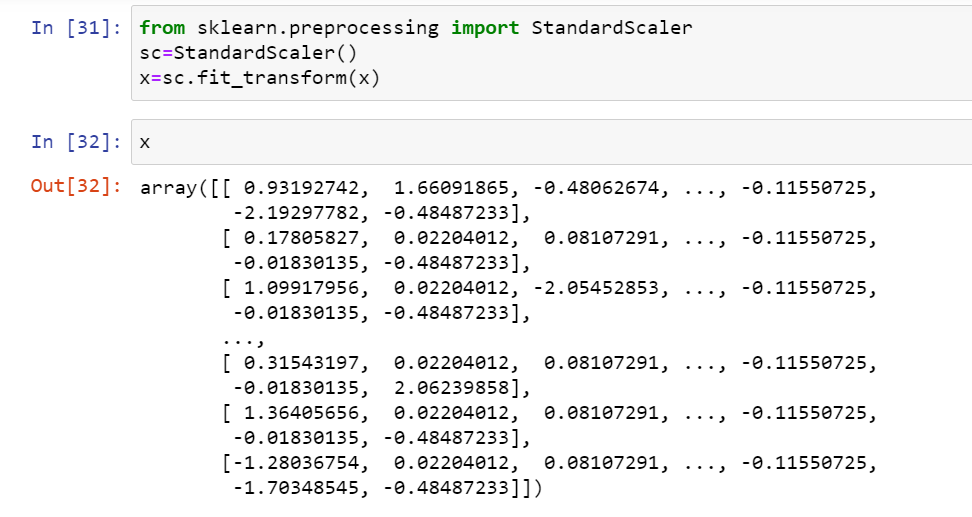
Standard Scaler helps to get standardized distribution, with a zero mean and standard deviation of one (unit variance). It standardizes features by subtracting the mean value from the feature and then dividing the result by feature standard deviation.

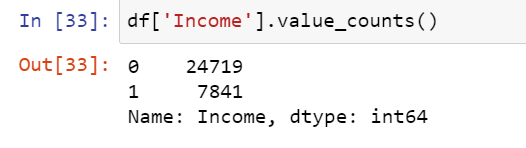
The standard scaling is calculated as:

z = (x - u) / s

Where :

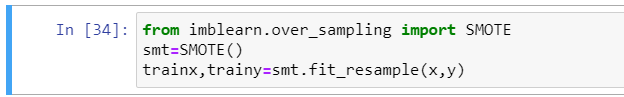
* z is scaled data.
* x is to be scaled data.
* u is the mean of the training samples
* s is the standard deviation of the training samples





Here, 0 is represented as people having salary over 50k and 1 represents salary below 50 k.

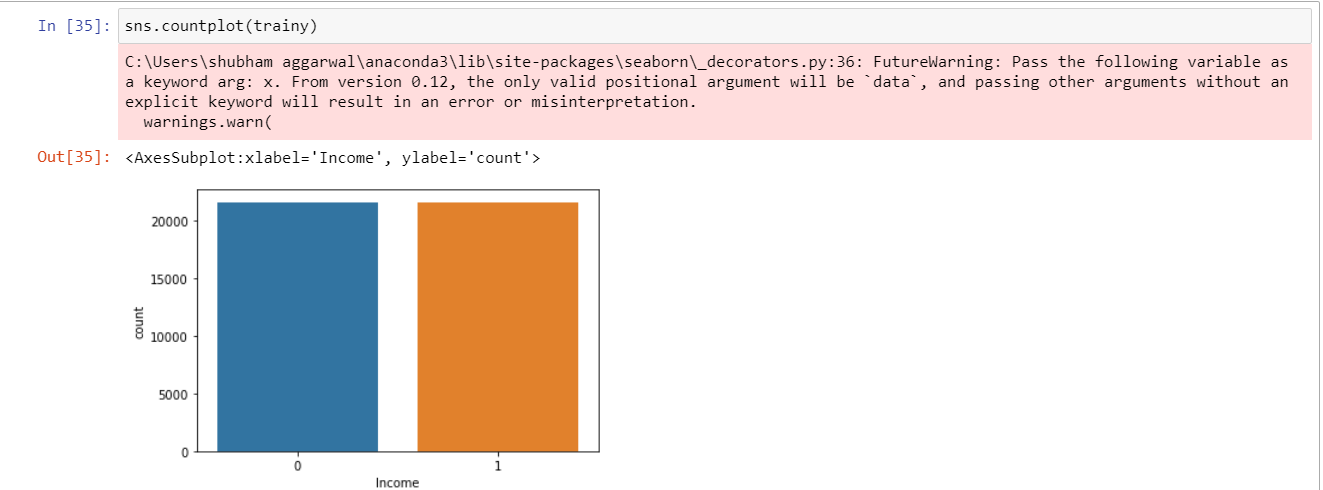
**Balancing the labeled columns:**



**SMOTE (synthetic minority oversampling technique)**

SMOTE (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem.   
It aims to balance class distribution by randomly increasing minority class examples by replicating them.

After smote our data looks like (only to be applied over label or target column )



**Data bifurcation:**



Here, the data is been divided into x and y where y is the target variable. Train to test ratio in the data is 80:20.

The best random state chosen is 997.

Here, in our data we have to predict whether the individuals salary is more than 50k or not so, based on this dataset, we will apply different libraries in order the get the best fit model and the most accurate and precise predictions.

Various algorithms applied here are:

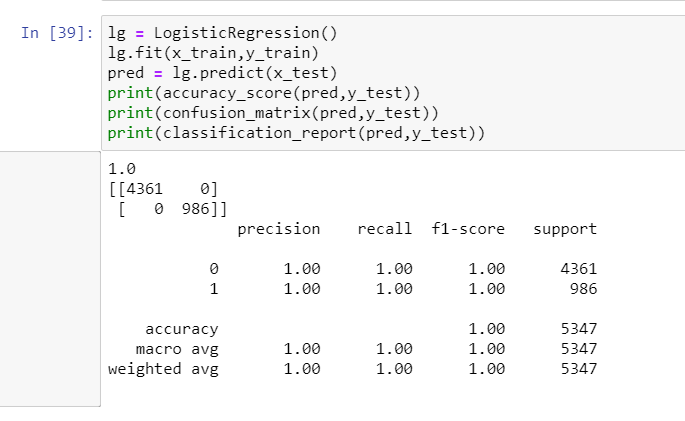
Logistic regression

Support vector machine

Decision tree

K neighbors classifier

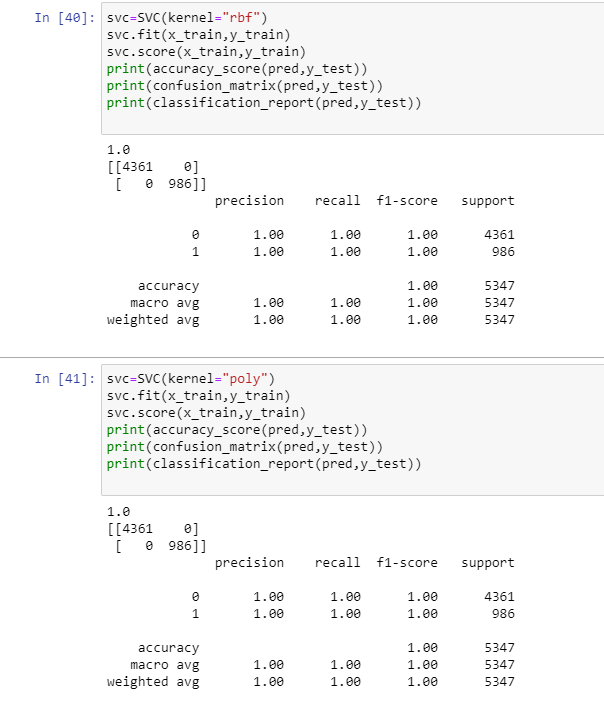
1. **Logistic regression:**



Logistic Regression was used in the biological sciences in early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.

As we can clearly see the precision value we got here is 1 hence, this is the best fit model.

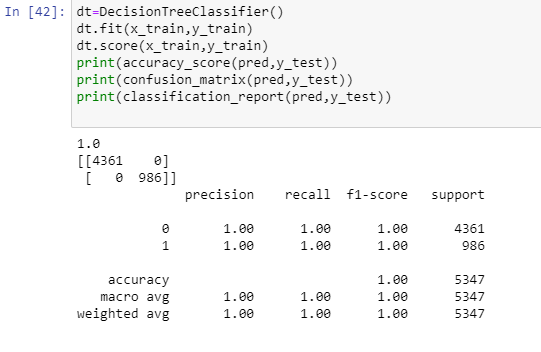
**2)Support vector machine**



The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

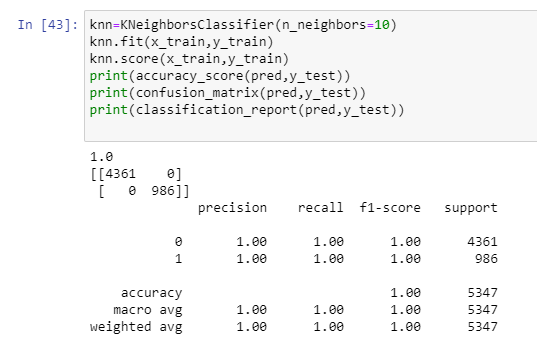
To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

**3)Decision tree**



A decision tree is **a flowchart-like structure in which each internal node represents a "test" on an attribute** (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes).

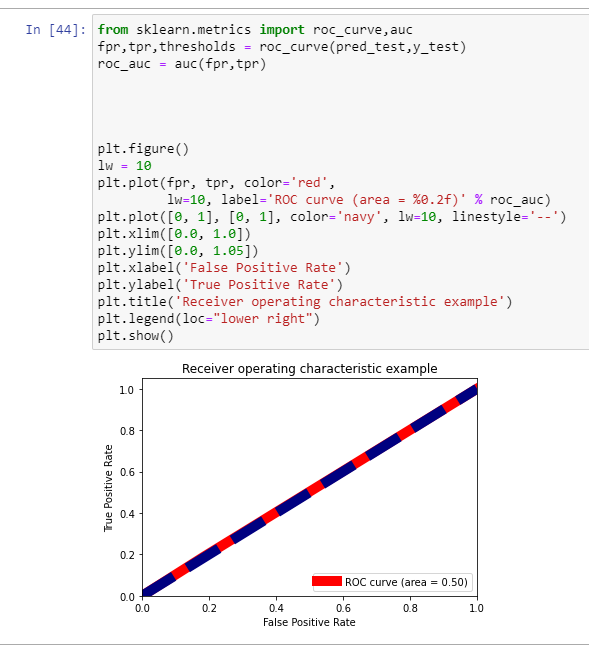
**4)Kneighbours classifier:**



By default, the KNeighborsClassifier **looks for the 5 nearest neighbors**. We must explicitly tell the classifier to use Euclidean distance for determining the proximity between neighboring points. knn = KNeighborsClassifier(n\_neighbors=5, metric='euclidean')

knn.fit(X\_train, y\_train)

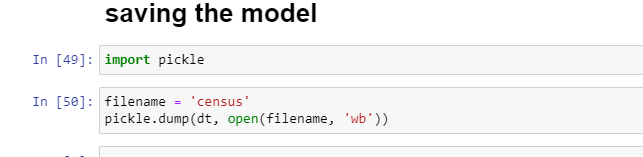
**AUC and ROC curve:**



The **Area Under the Curve** (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

Here, we can see that ROC i.e. receiver operating characteristic benn completly overlapped hence, this is the best fit model.

**Saving the model:**



**Conclusion:**

* The accuracy score or the best score predicted is 100% with logistic regression,
* K-neighbour classifier, support vector machine and decision tree.
* Saving the best predicted model
* Using the pickle library to save the model for further usage.