|  |
| --- |
| Data Trained Institute  Census Income Project  *Submitted By,*  *Poovarasi Vijayan*  *Batch: 1840* |

# Problem Statement:

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 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:

1. A single cell estimates 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.

**Link:**

<https://raw.githubusercontent.com/dsrscientist/dataset1/master/census_income.csv>

# Context of Project:

This project is to find whether a person earns over $50K or not. To find this, below listed steps are followed.

1. Import the data
2. Clean the data
3. Analyze the data
4. Standardize the data
5. Draw up the potential ML models
6. Shortlist / Finalize ML model
7. Hyper-tune the ML model
8. Predict the output
9. Save the ML model

This project is an attempt to realize a solution for the prediction of Income earned by person is over $50K. Once the optimal ML model has been created / chosen, it can be deployed as a real-time prediction device for Census Income Prediction.

# About the data:

This dataset contains

Rows: 32560

Columns: 15

Here the target variable is ‘Income’

# Insights of data:

From this Data Available, we can bring out some insights or conclusions:

* What are the main factors affecting the Income earned by person?
* What type of job can make income over $50k?
* What education type makes income over $50k?
* Which type of people gets more income?
* What occupation makes over $50k income?
* Which country people earn more income?

# Tools Used:

* Python 3.8
* Numpy
* Pandas
* Matplotlib
* Seaborn
* Data Science
* Machine Learning

# Data Analysis:

For this project pandas, numpy, matplotlib, seaborn libraries are used.



# Importing data:



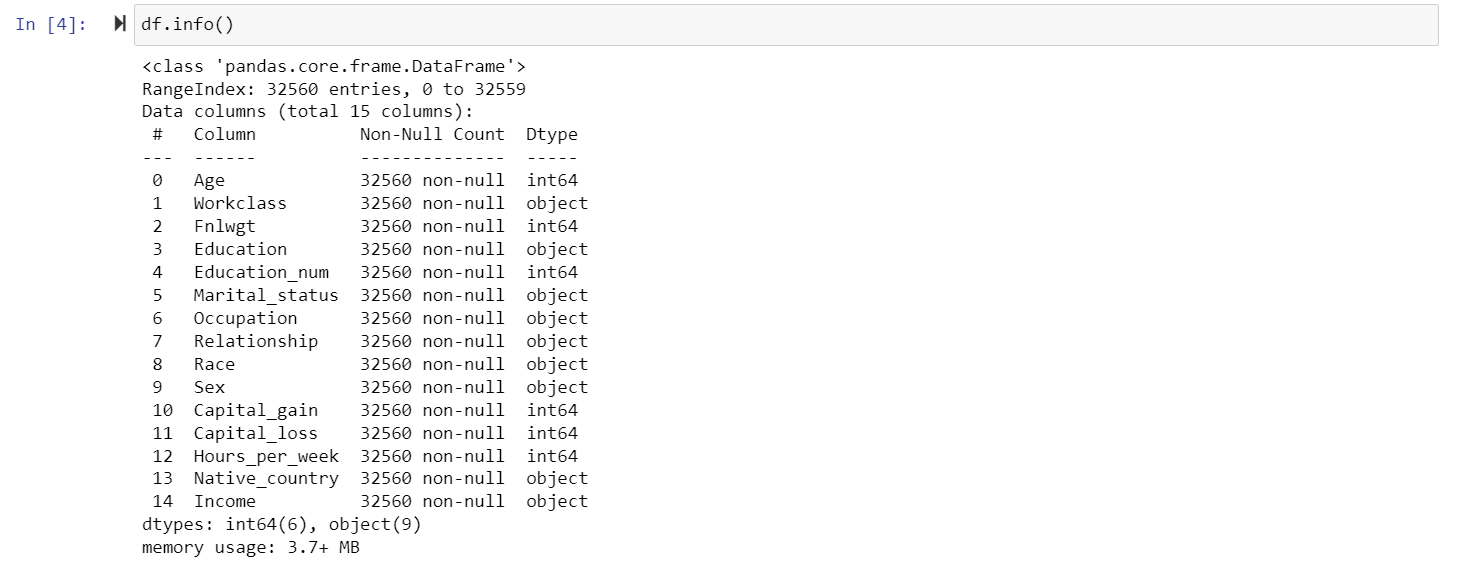
We will start Pre-Processing by analysing basics details from the data. As per our problem statement, we need to predict whether the income is over $50k or not. So, feature variable is converted into binary to build classification model. In this case, ‘Income’ is the feature variable.

# Exploratory Data Analysis (EDA):

EDA is performed to discover trends, patterns and to check assumptions with the help of statistical summary.

First, columns and its information are checked.

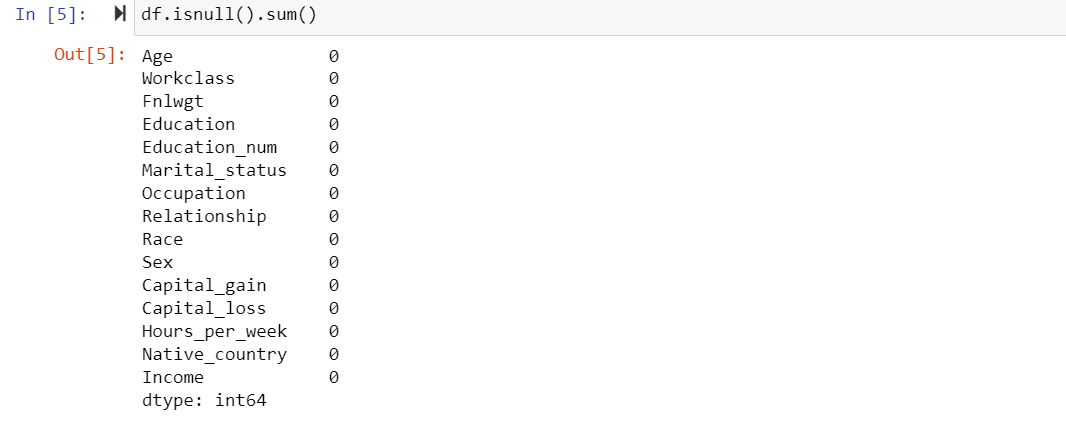




Here the dataset contains int and object datatypes.

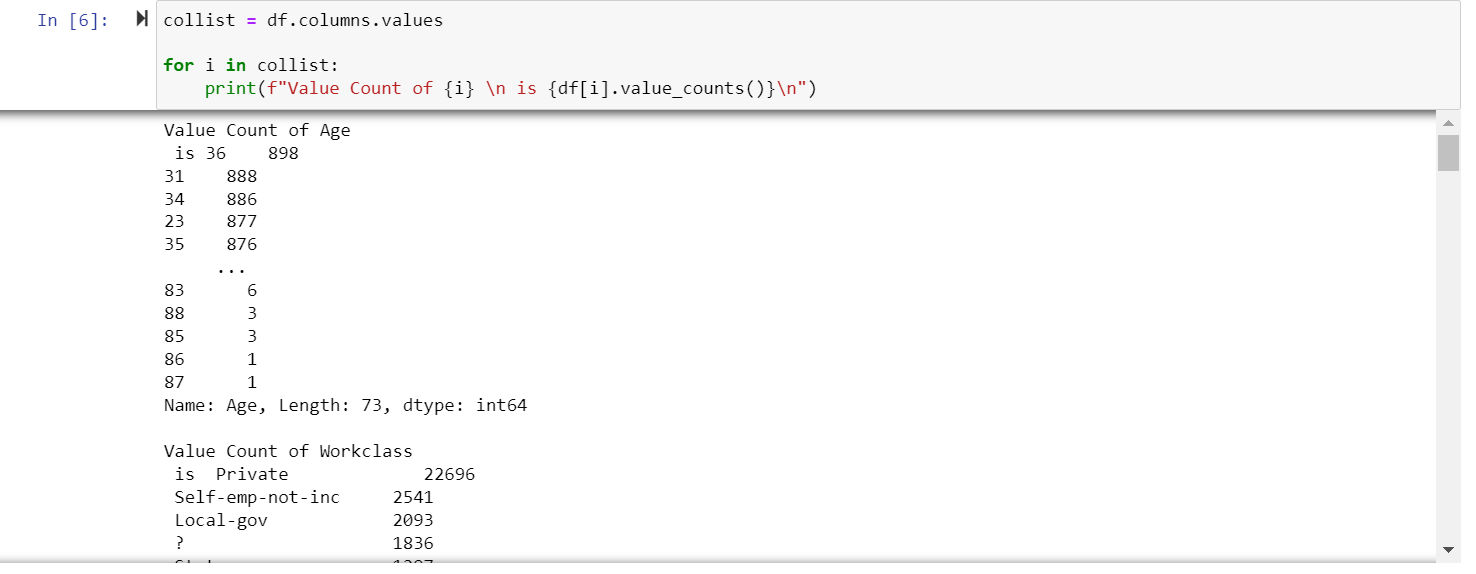
# Data Cleaning and Preparation:

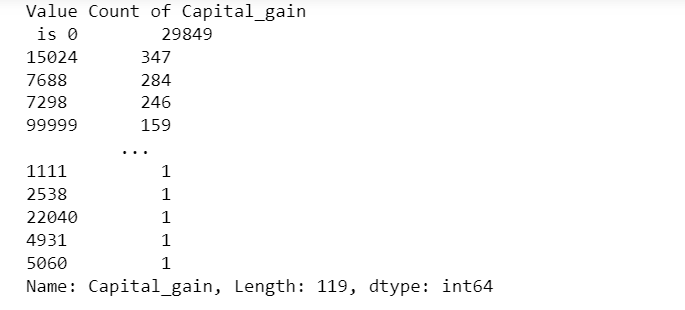
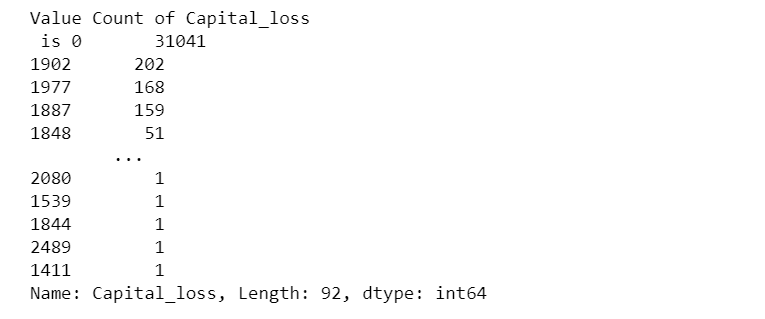
In next step is to check if dataset contains any null values. If there are any null values, it will be dropped or replaced.



In this dataset, there are no null values. So, we can continue with further analysis.

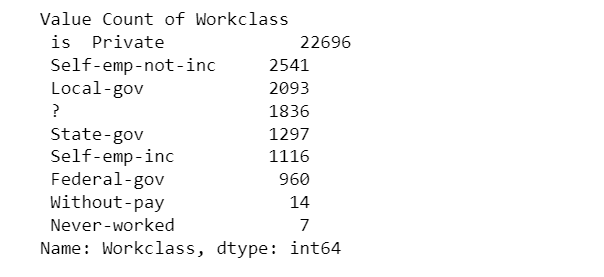
Then the count of each values are checked using value\_counts() to check if there are any invalid values ex., ?, /.

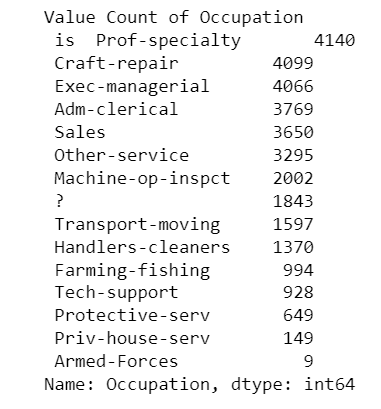




‘Capital\_gain’,’Capital\_loss’ columns are dropped as more than 90 percent of values are 0. It is dropped because it cannot be replaced with other values as it may create bias in data.







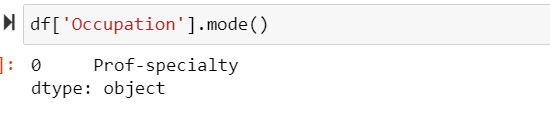
In the above observations ‘Work\_class’ contains 1836 ‘?’ (missing values), ‘Native\_country’ contains 583 ‘?’ (missing values) and ‘Occupation’ contains 1816 ‘?’ (missing values). It may be due to typing errors or technical error. So, let’s replace those unwanted entries. This can be replaced with mode () as it is of object type.





All ‘?’ is replaced with mode values and checked whether all rows are having value.

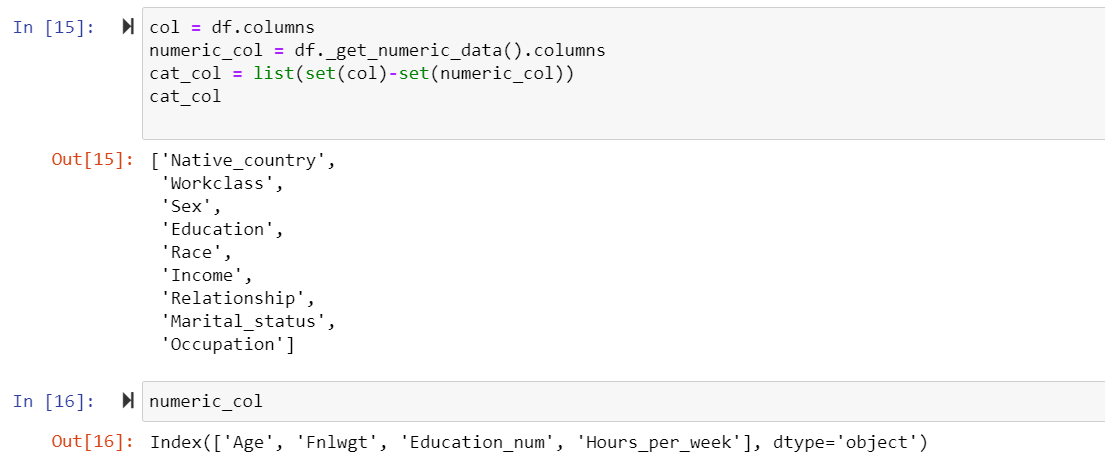
Similarly, other rows containing ‘?’ are replaced with mode().





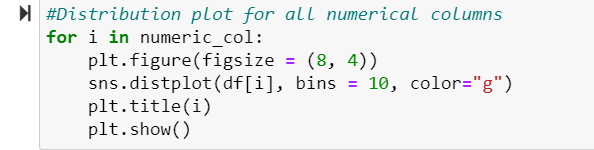


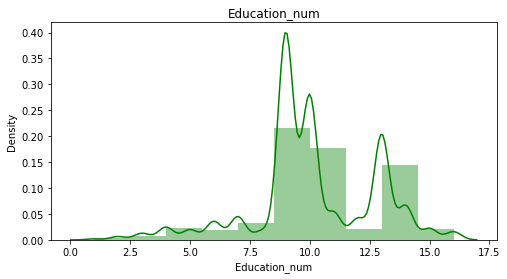
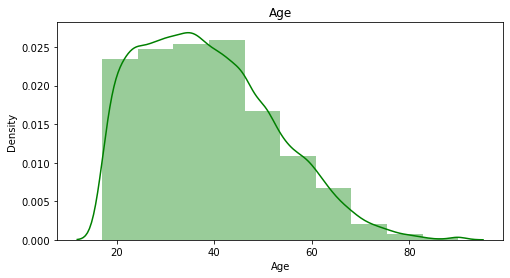
Then numeric and categorical columns are checked, so that further analysis and visualization can be done.



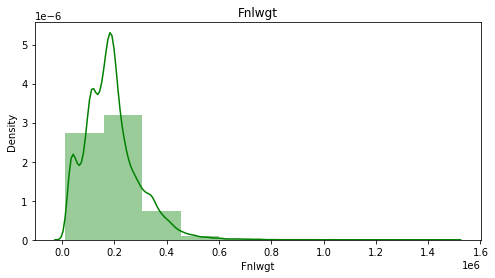
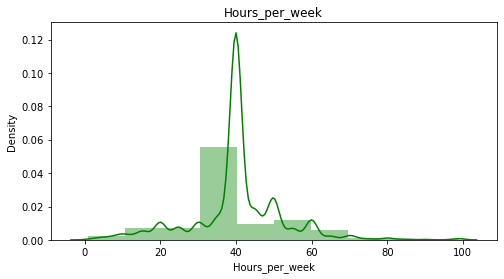
# Data Visualization:

Numeric data is visualized using distplot.



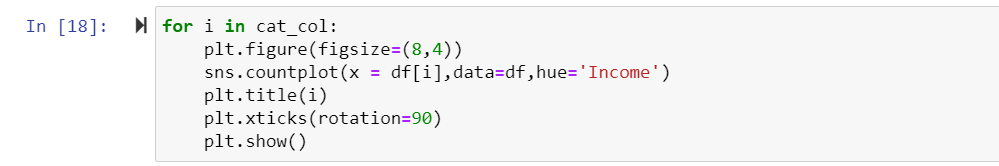


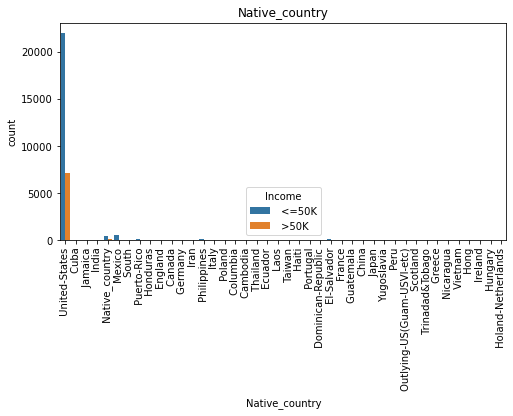
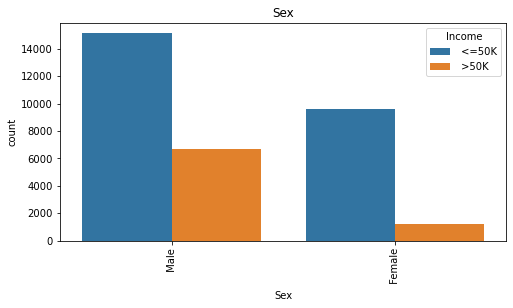
* ‘Age’ column is right skewed
* Education\_num is left skewed

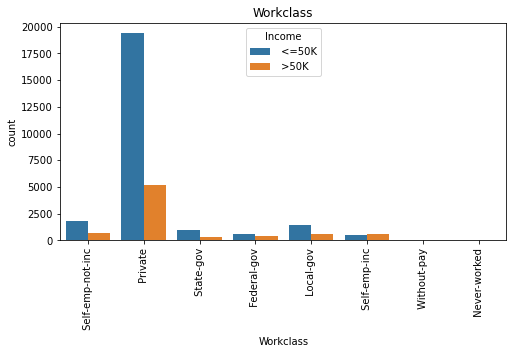


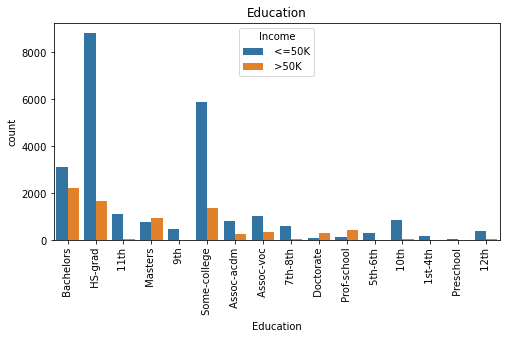
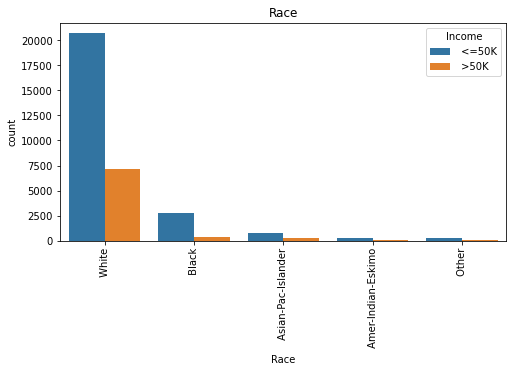
* ‘Fnlwgt’ column is right skewed
* ‘Hours\_per\_week’ is also skewed

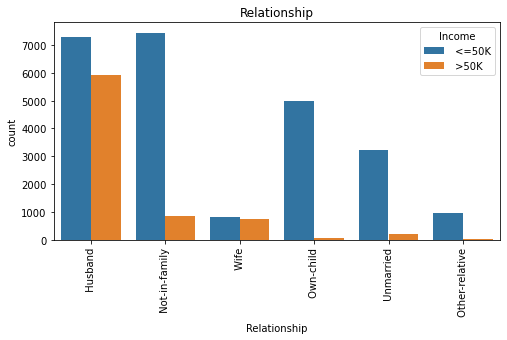
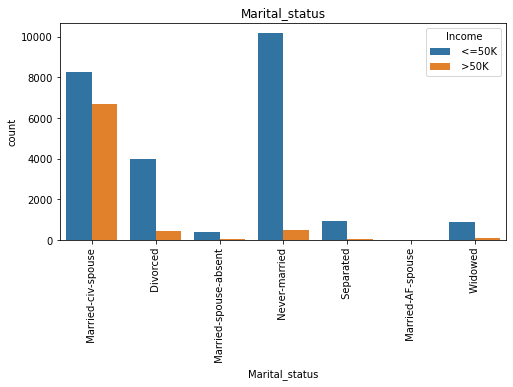
Categorical data is visualized using count plot.

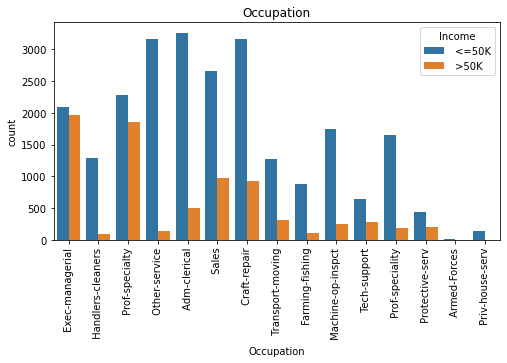
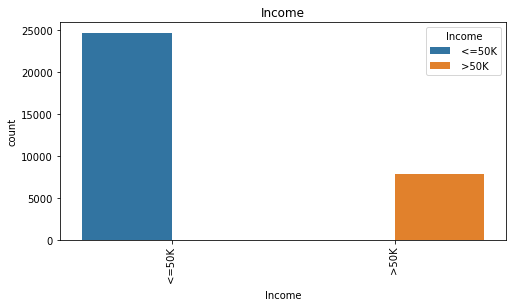










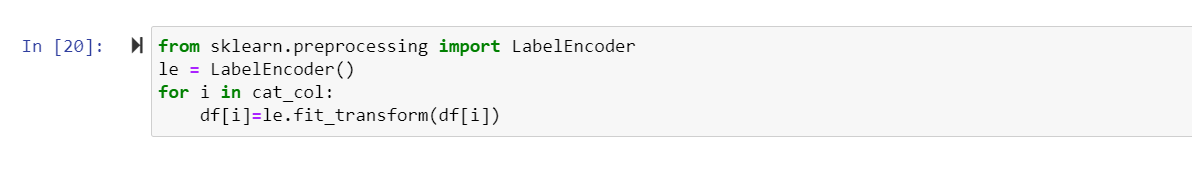


**Key Observations:**

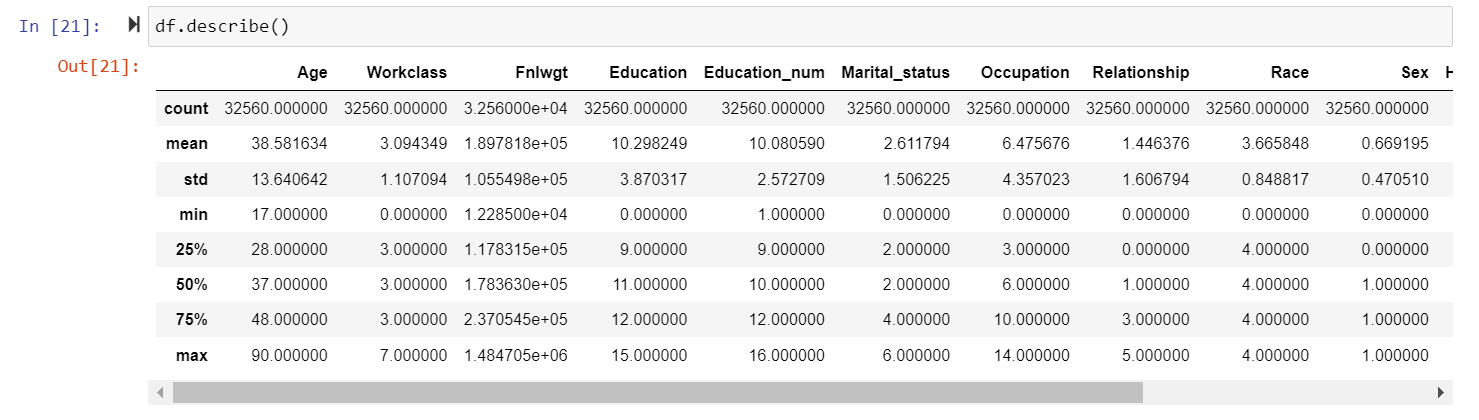
* More number of people earn less than or equal to 50000
* People working in private sector earn more income compare to all other sectors.
* People with Bachelors, HS-grade and Some college education earn more income compare to all other category people.
* Maximum number of Married with spouse earn more than 50000.More number of never married people earn less than or equal to 50000 compare to all others
* In the occupations exec-managerial and prof-specialty almost equal number of people earn less than or equal to 50000 and more than 50000.In rest of the occupations more number people earn less than or equal to 50000
* White people earn far more than any other race people.
* Native people of US earn far more than any other country people.
* Male earn more than females.

# Data Pre-processing and Feature Engineering:

By using Label Encoder all Categorical columns are converted to numeric one so that analysis can be made in better way.

Now, all categorical columns are converted to numeric one.

Now will check the description of DataFrame.



**Key Observations:**

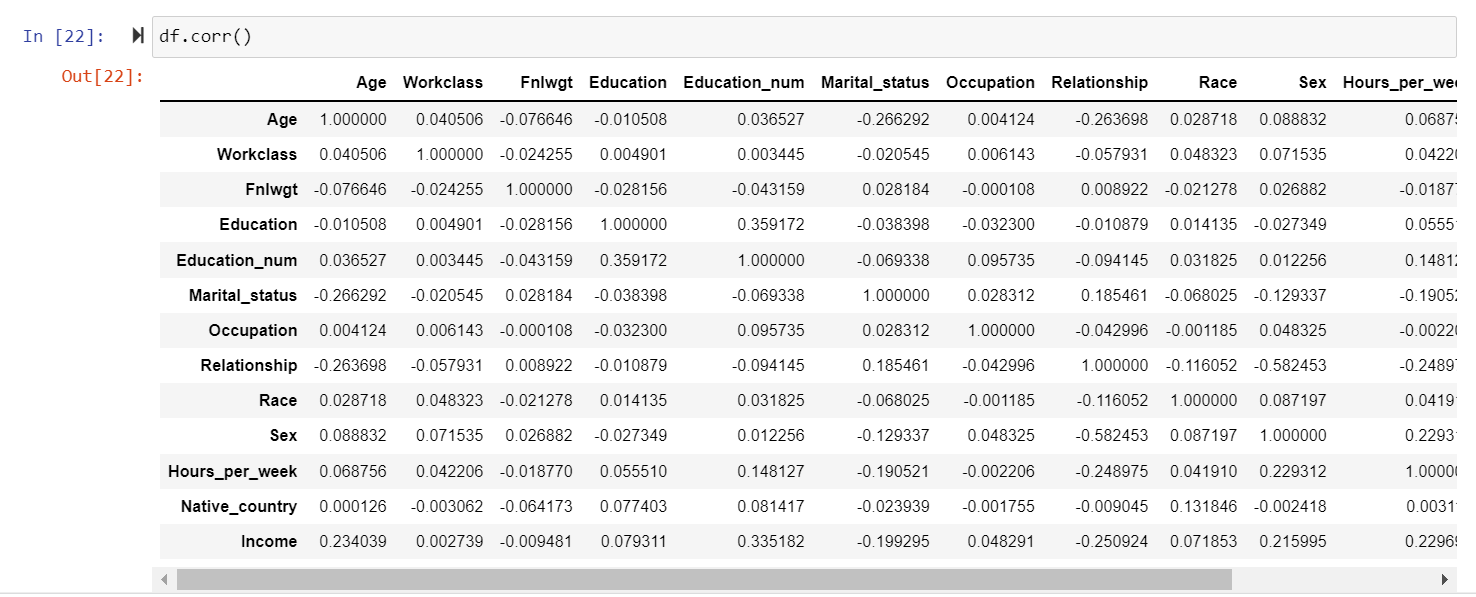
As per above sort description we have seen in ‘count’ there is no null values in the dataset.

* By this we can infer that skewness is present in few columns.
* And Outliers are present in 'Age', 'Hours\_per\_week' column.

we will conform this by visualization method and remove all skewness and outliers present in our dataset.

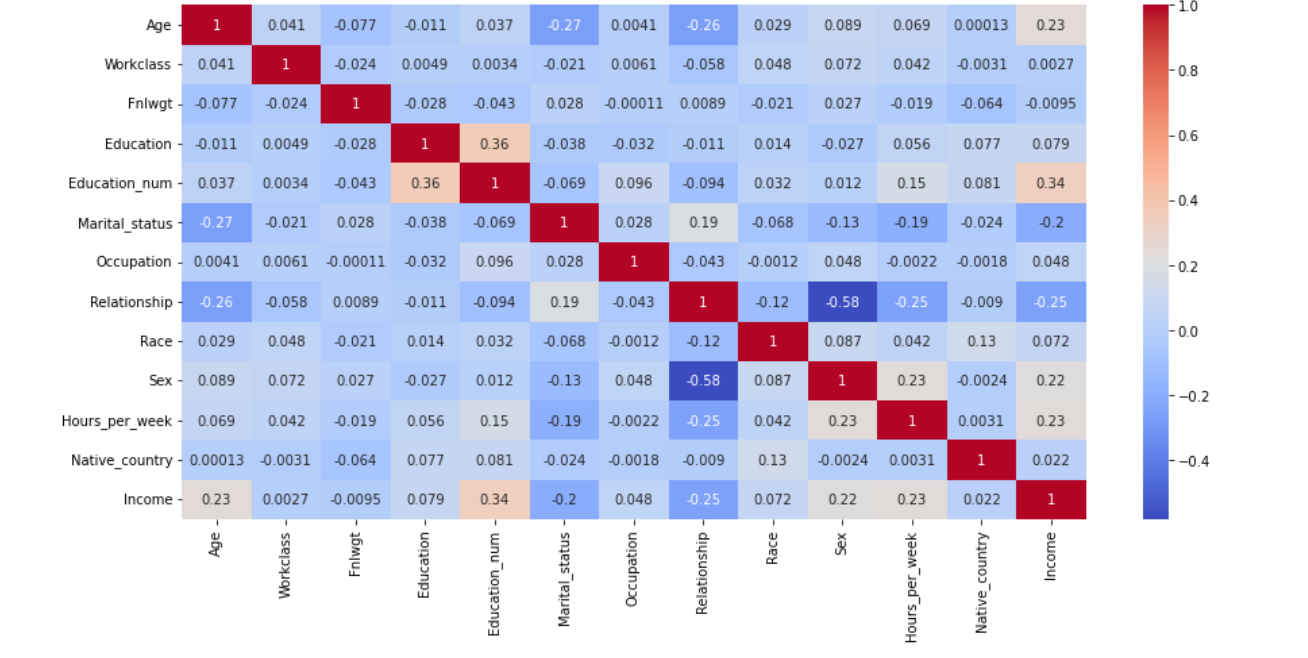
# Correlation:

corr() is used to find the pairwise correlation of all columns in the dataframe.



After checking the correlation, heat map is plotted to get better insight on the corr() values.





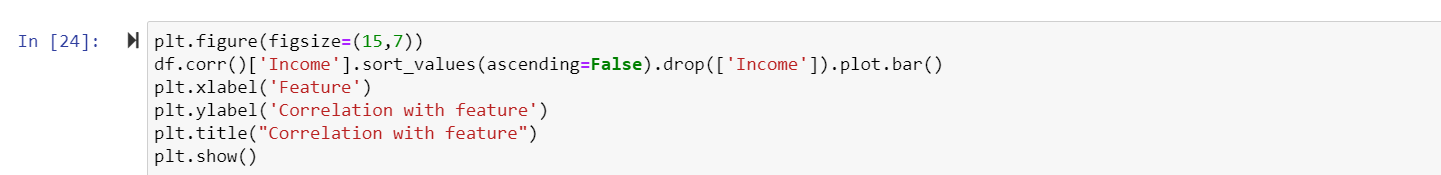
**Key Observation:**

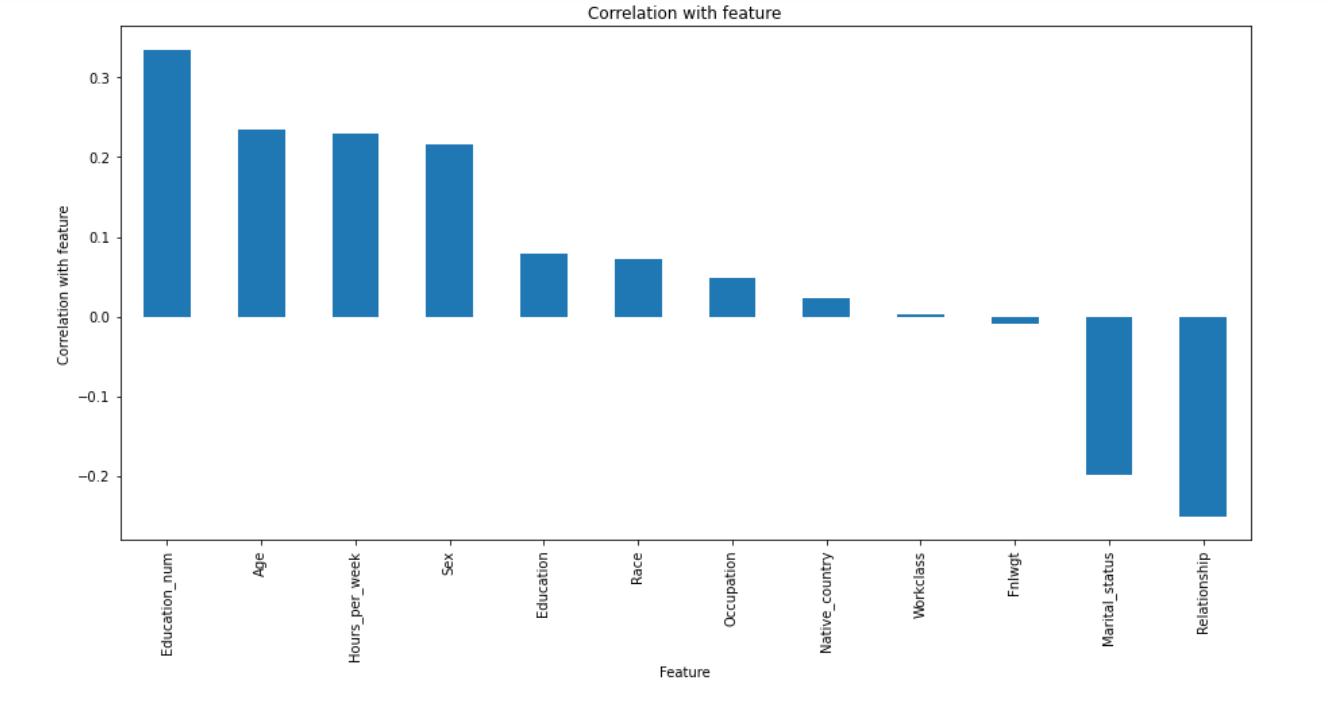
* Positive correlation:

'Education\_num', 'Age', 'Hours\_per\_week', 'Sex', 'Education', 'Work\_class', 'Race', 'Native\_country'

* Negative correlation:

'Relationship', 'MaritalStatus', 'Fnlwgt'

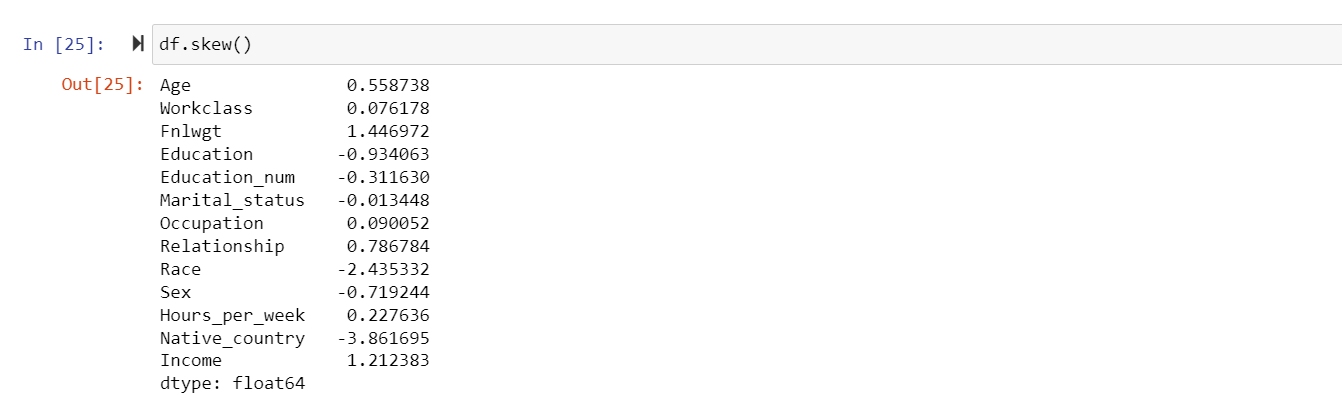


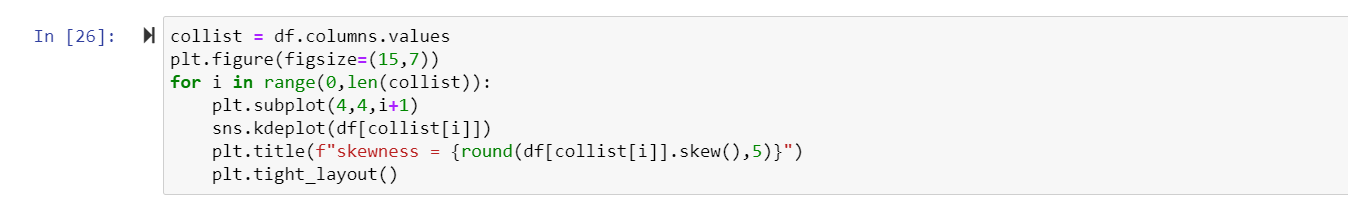


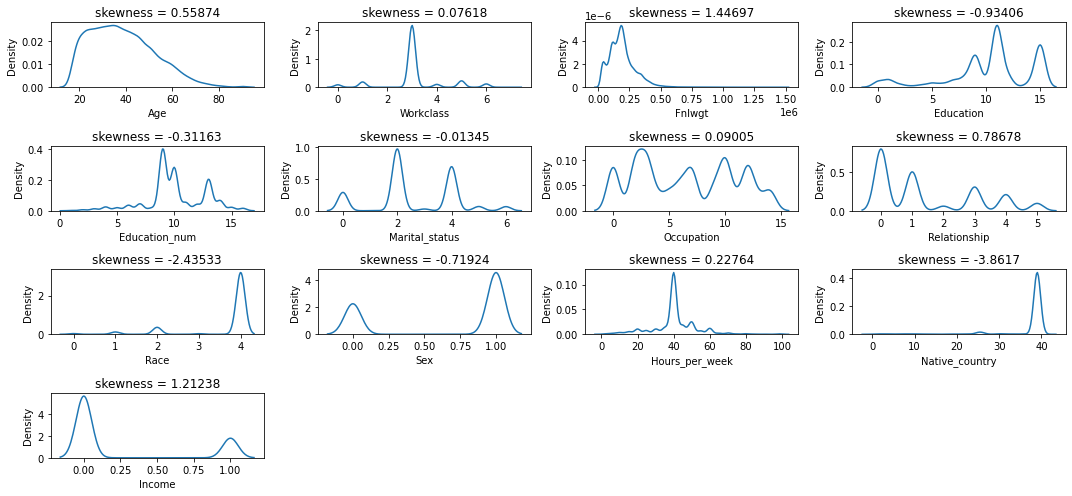
* ‘Education\_num’ is highly correlated with our target(‘Income’) column.
* ‘Relationship’, ‘Marital\_status’ is negatively correlated with target column.
* It is true that if a person is highly educated his Income also increases, whereas 'Relationship' does not have much contribution in Income earned by a person.

# Skewness in data:

Skewness refers to a distortion or asymmetry that deviates from the symmetrical bell curve, or normal distribution, in a set of data.







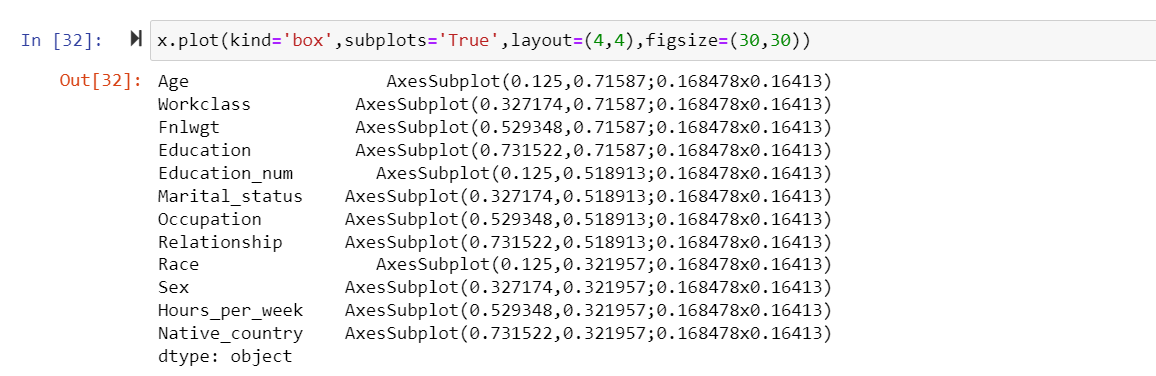
skewness is present in 'Workclass’, ‘Fnlwgt’, ‘Relationship’, ‘Race', 'Sex’, 'Native\_country', ' Income'.

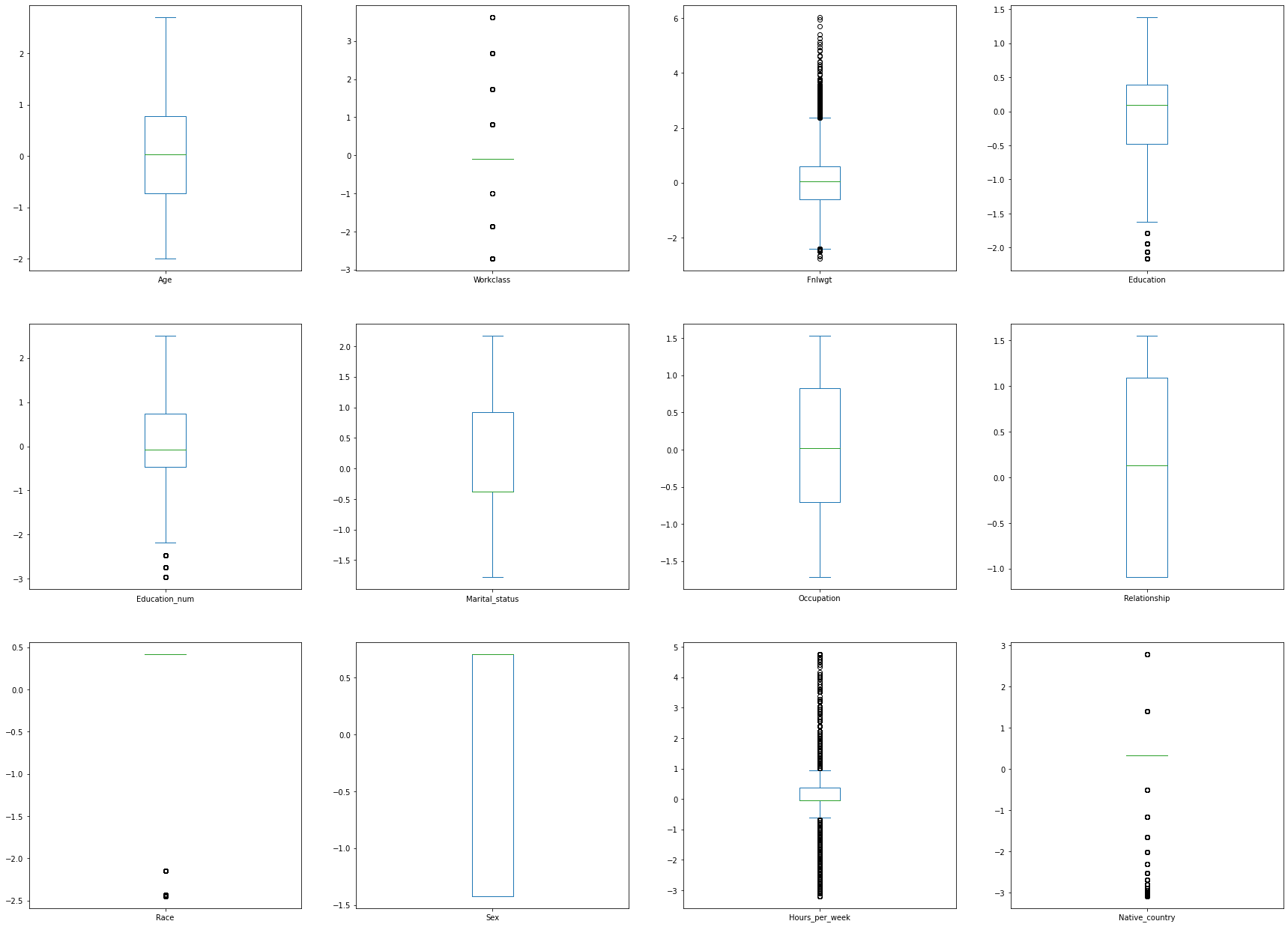
‘yeo-johnson’ method is applied to remove skewness so that the data will be ready for testing.



# Outliers Detection:

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population.

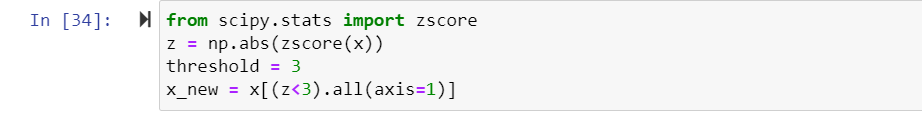


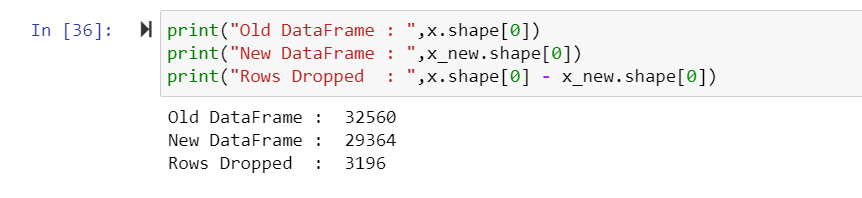


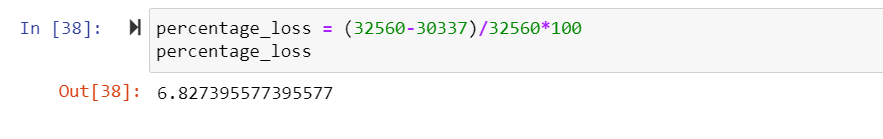
Here outliers are present in few columns.

'Workclass', 'Fnlwgt', 'Education', 'Education\_num', 'Race', 'Hours\_per\_week', 'Native\_country'

Outliers are removed by using z-score.





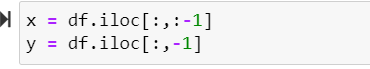


By removing outliers we have 6.8% data loss, which is less than 10% so we can remove those outliers for better accuracy of model while training the dataset.

# Scaling the data:

Standard Scaler removes the mean and scales each feature/variable to unit variance.

* As a first step to make model, separate the dependent and independent features.
* Taken x as all independent features and y as dependent/target feature.
* Scaled independent features to get the same range in all the columns. If independent columns are not scaled then there is a chance that, model may get baised.



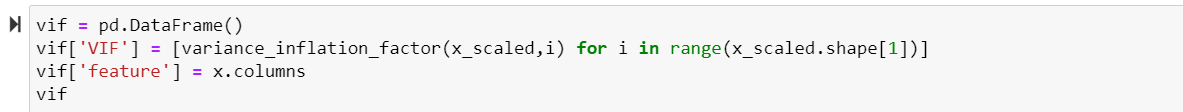


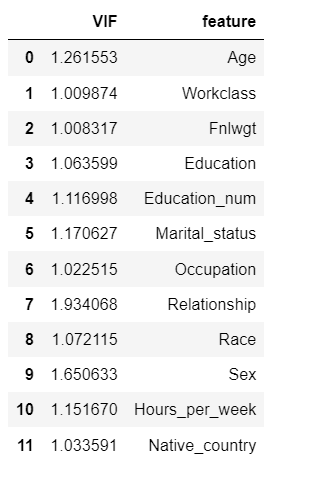
Now scaling is done. Then Multicollinearity is checked using VIF (variance inflation factor) in order to find if any column is correlated with each other.

# Multicolinearity:

Multicollinearity (or collinearity) occurs when one independent variable in a model is linearly correlated with another independent variable.





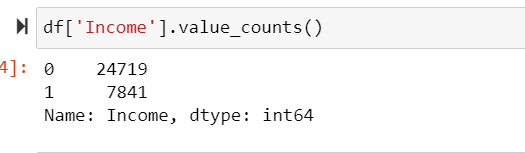


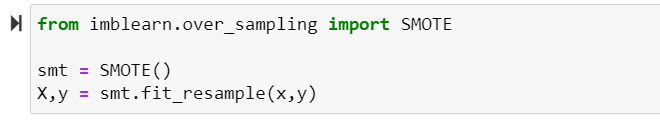
Here we have all VIF values less than 10 so no need to drop any columns. If any VIF value greater than 10 then that column need to be dropped.

# Balancing the data:

Then balancing of the target variable is done using oversampling method. It can be done with under sampling also but due to data loss, it is not used.

Then value\_counts of target column is checked, as there is huge difference in values, SMOTE (oversampling) method is used to balance the variable.







Now the target column is balanced.

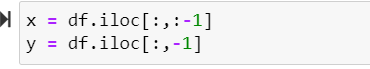
Data is all set for model building. Let’s go ahead with classification algorithms as we are dealing with Classification Problem.

# EDA Remarks:

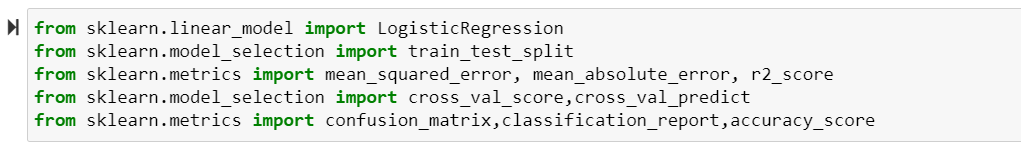
* Checked for NaN values and there are no missing values in the dataset.
* Dropped the unnecessary columns and replaced the ‘?’ entries with their suitable values.
* Used both matplotlib and seaborn to visualize the data.
* Used distplot, countplot to get better insight on the features. Since most of my columns were categorical, I have used all categorical plots. For numerical columns I have used numerical plotting but I did not get any good pattern with numerical columns.
* Outliers and Skewness is removed then Scaling of data is done.
* Checked for Multicolinearity to find if any columns is correlated with each other, in this project there is no multicollinearity present.
* Balancing of Target variable is done by using over sampling(SMOTE) technique.

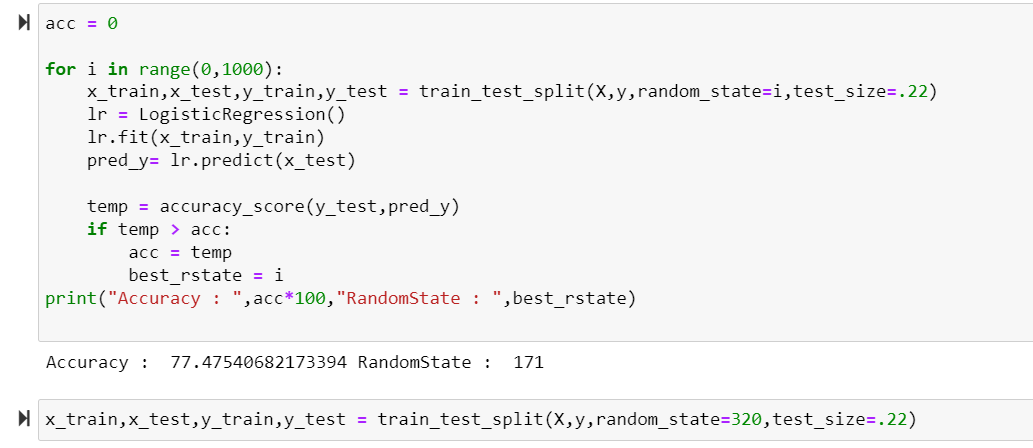
# Building Machine Learning Models:

Now we will train data using several Machine Learning models and compare their results. First, we need to use the sklearn train test split and divide the dataset into test and train. Before that we have to divide the data into dependent and independent variables.



# Finding best random state and accuracy:





Got Accuracy as 77.4 at Random state 171.

Train and test data is splitted as x\_train, x\_test, y\_train, y\_test.

Then Cross validation is used as model evaluation metrics for all the algorithms and accuracy\_score, Confusion metrics, classification report as metrics in model building.

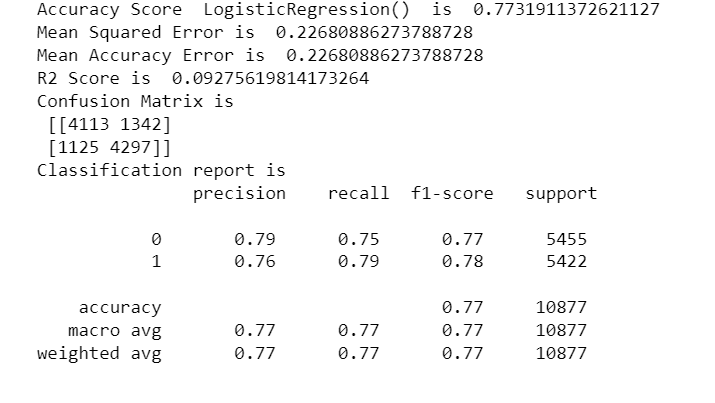
# Classification algorithm:



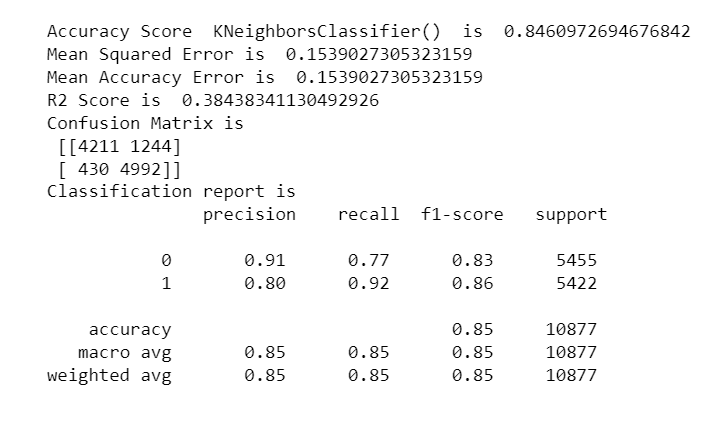
We are using 7 models. Let’s train these models now



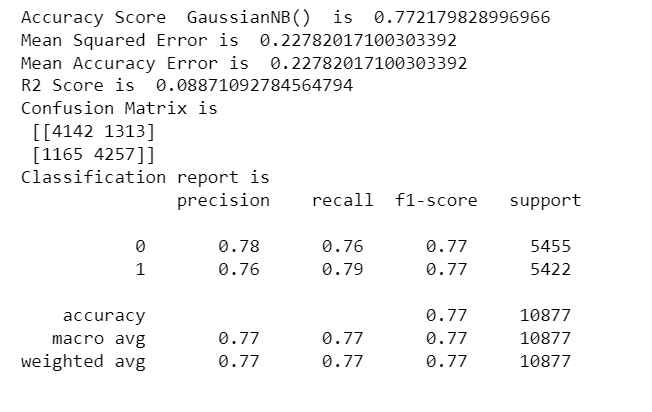
Model accuracy is checked by accuracy\_score, confusion\_matrix , classification\_report and r2\_score



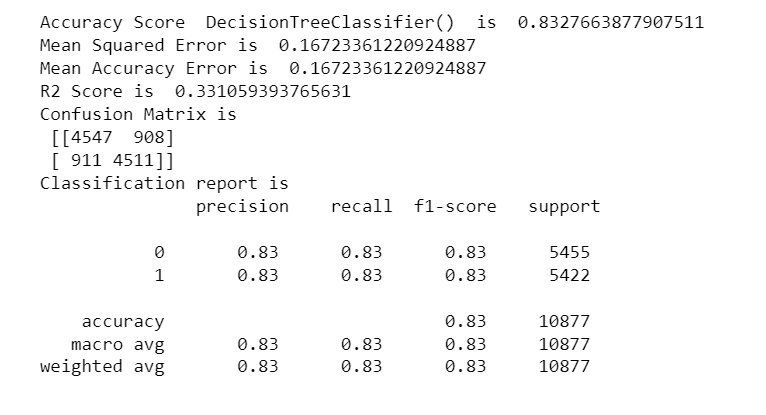
Accuracy Score of Linear Regression is 77%



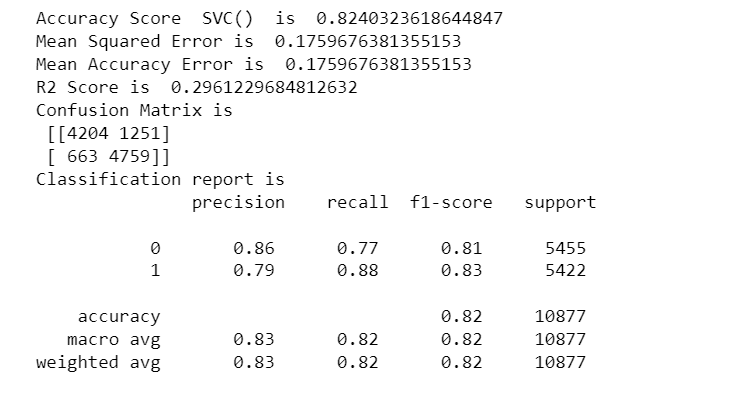
Accuracy Score of KNeighborsClassifier is 84%



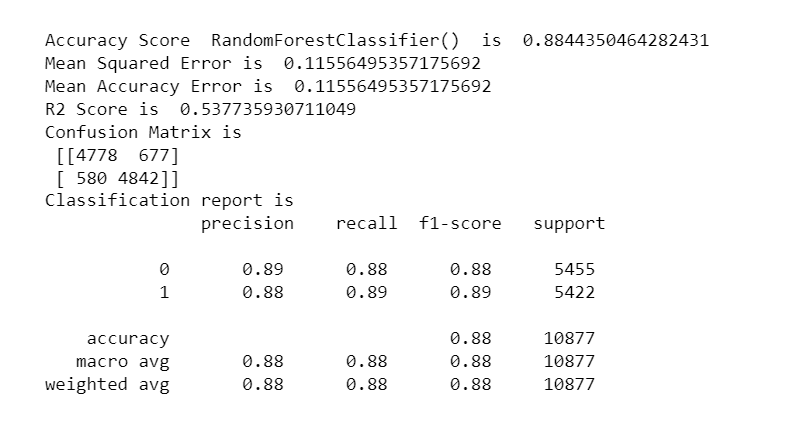
Accuracy score of GaussianNB is 77%



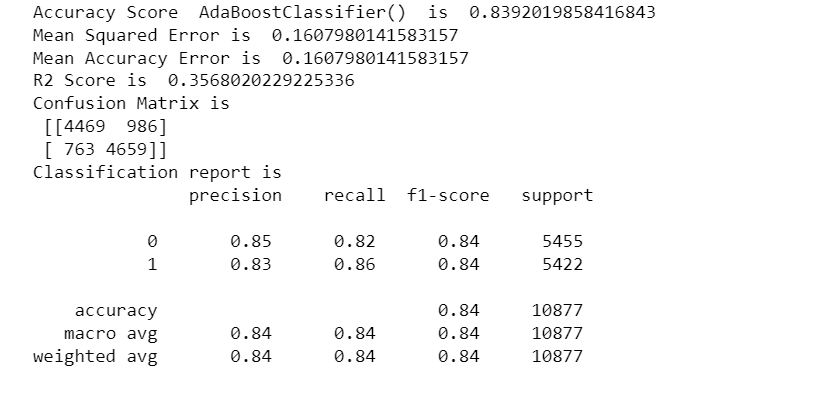
Accuracy Score of DecisionTreeClassifier is 83%



Accuracy score of SVC IS 82%



Accuracy score of RandomForestClassifier is 88%

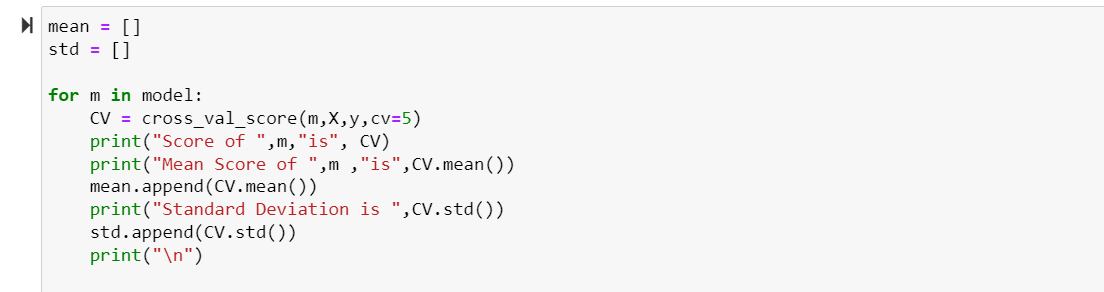


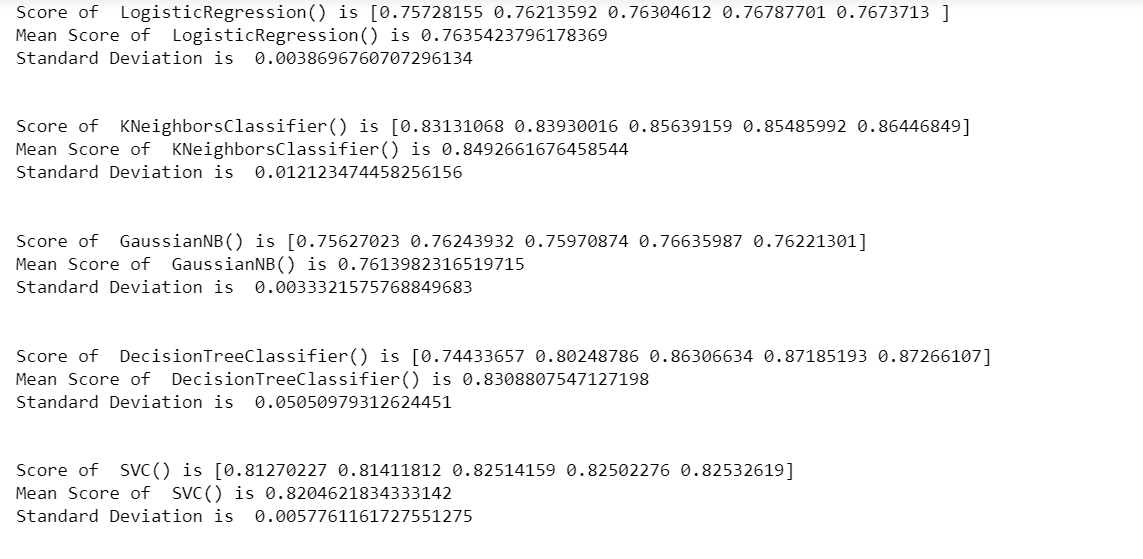
Accuracy score of AdaBoostClassifier is 84%

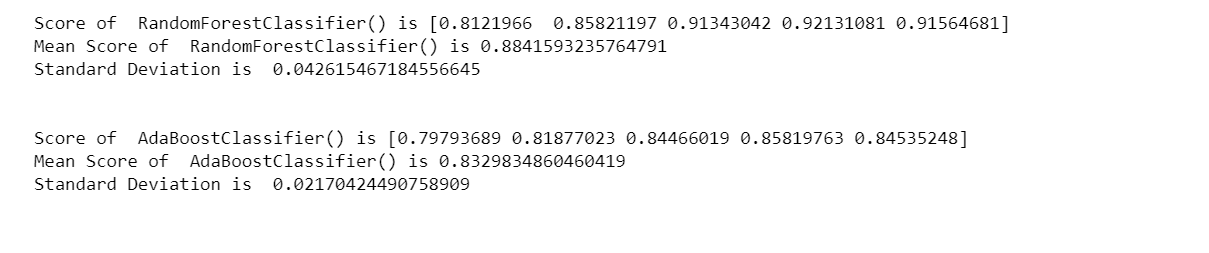
Then Cross Validation, for all the models for evaluation.

# Cross Validation:

Cross-Validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias and to give an insight on how the model will generalize to an independent dataset.





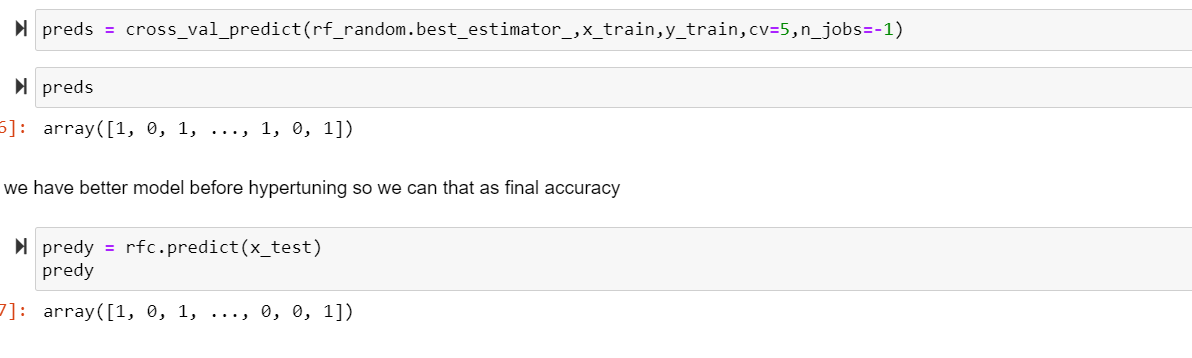


We have RandomForestClassifier with 88% score which can be improved with Hypertuning.

# Hypertuning:

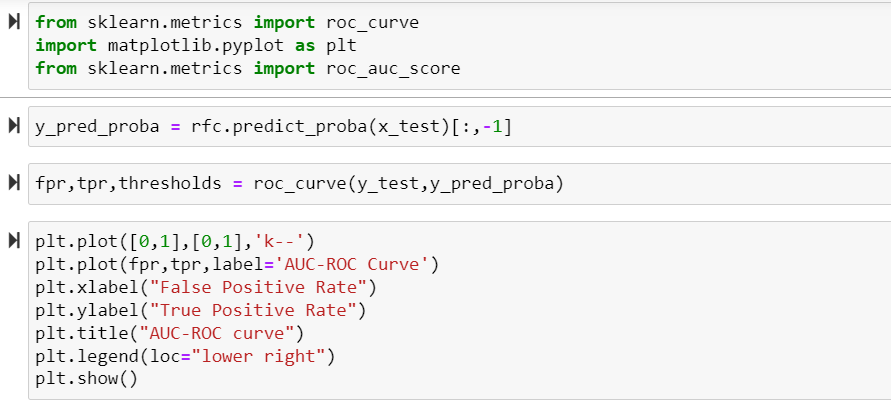


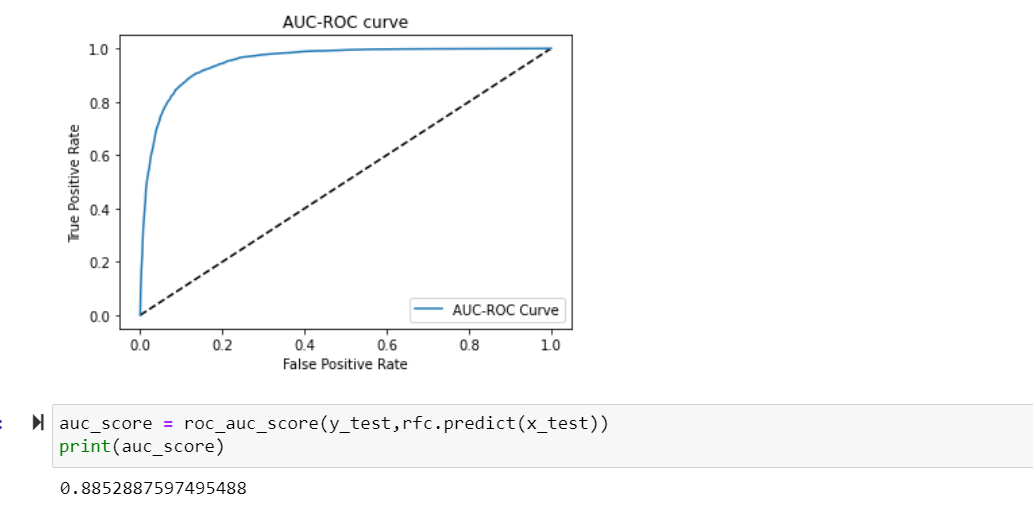
We have assigned “rf\_model = grid\_search. best\_estimator\_” we can see that our model has reduced scored from 88.4% to 87.7%(almost similar) because it has corrected the overfitting.



# AUC-ROC Curve:

Another way to evaluate and compare your binary classifier is provided by the ROC AUC Curve. This curve plots the true positive rate (also called recall) against the false positive rate (ratio of incorrectly classified negative instances), instead of plotting the precision versus the recall.





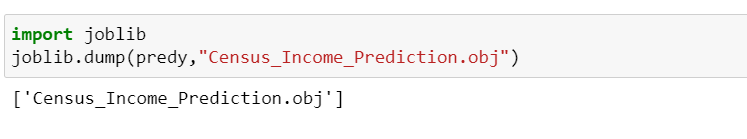
Above is the ROC curve for best model. And the AUC value also remained almost same.

After getting best model I have saved the model.

As we observed in all methods, we are getting almost same score.

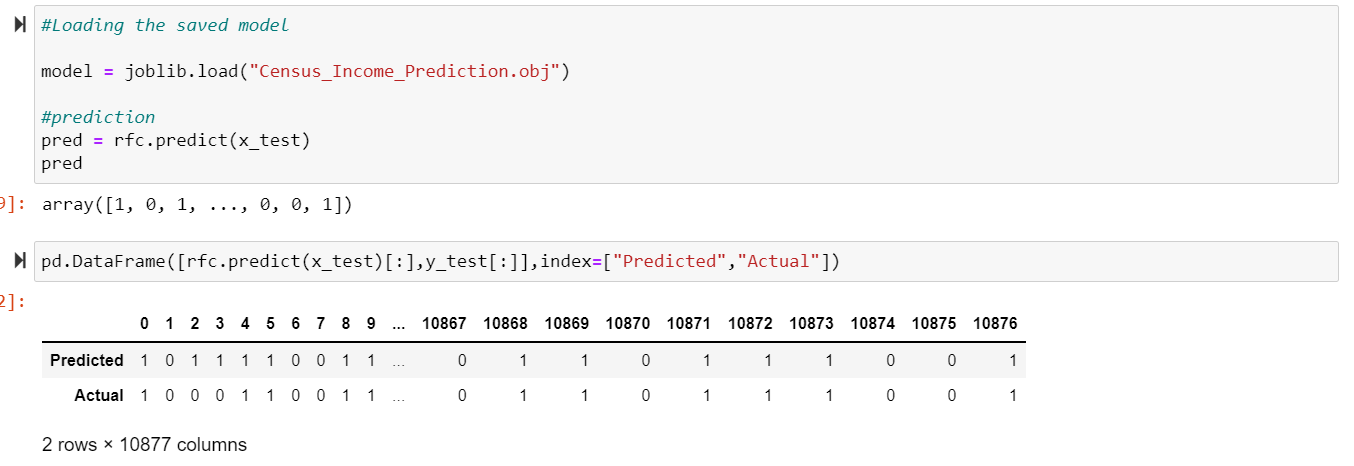
# Saving the Best Model:

We are saving our model with joblib library as ‘Census\_Income\_Prediction’.



# Prediction:

Now using the saved model, we can predict whether the Income earned is over $50k or not.



**Conclusion:**

* We have started our project by importing the dataset, Data cleaning, data pre-processing, visualizations are made. With clean data, skewness and outliers are checked and removed. Standard Scaling, Multicolinearity and balancing of target variable is done.
* Then Cleaned data is prepared for training on splitting it as Target and Feature variable. We have trained our data with 7 models, then we have finalized Random Forest Regressor with model score of 87% and with CV-Score of 88%. Further to increase the performance of our model we have hyper tuned our model with GridSearchCV and we have reduced the overfitting our final model score is 0.878. And using this Machine Learning Model one can predict whether a person earns over $50k or not.
* By this ‘Census\_Income\_Prediction’ project we have found what are all the main factors affecting the Income earned by the person, which type of person gets more Income, Which Country people earns more, what occupation gets more Income and what are all the factors that does not affect the Income of a person.