**Project Name:** Census Income Predictions

**Author:** Aniruddha Sawant

* **Problem Definition.**

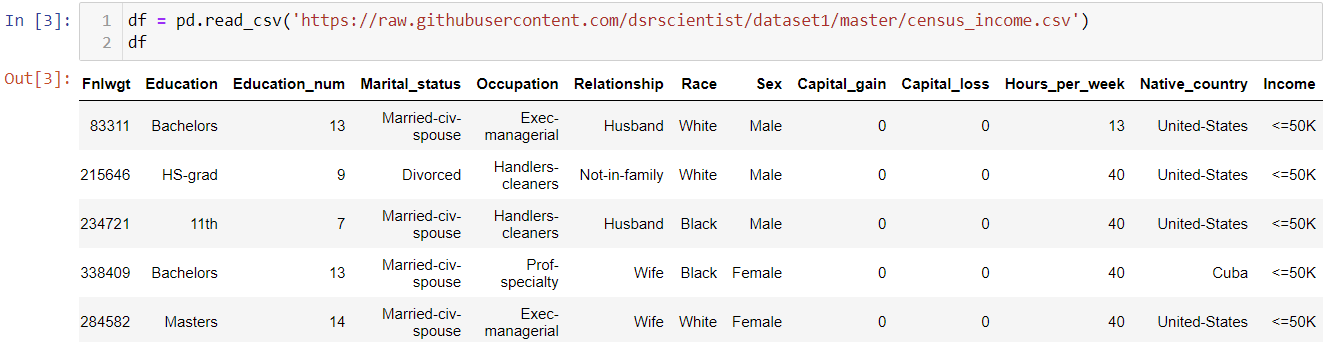
Computing or foreseeing an individual's pay is vital. The pay of individual concludes the development and success of a country and it very well may be helpful in different cases to be specific promoting, research, etc. In this undertaking we will anticipate the individual's pay in view of different highlights and factors like his Age, Education, Occupation, Sex, etc.

To anticipate this information, we want to make an AI model for which we required information and we have the wellspring of such information. This information was separated from the 1994 Census agency data set by Ronny Kohavi and Barry Becker (Data Mining and Visualization, Silicon Graphics). A bunch of sensibly clean records was separated utilizing the accompanying circumstances: ((AGE>16) and (AGI>100) and (AFNLWGT>1) and (HRSWK>0)).

The expectation task is to decide if an individual makes more than $50K a year in view of given factors.

You can find the data set [here](https://raw.githubusercontent.com/dsrscientist/dataset1/master/census_income.csv).

Below is the snapshot of a dataset: -



In this Dataset Income is the Label. We are building a model to Forecast the Income. There are complete 14 features which are: -

1. Age
2. Workclass
3. Fnlwgt
4. Education
5. Education\_num
6. Marital\_status
7. Occupation
8. Relationship
9. Race
10. Sex
11. Capital\_gain
12. Capital\_loss
13. Hours\_per\_week
14. Native\_country

* **Data Analysis.**
* We have total 15 columns including the label i.e Income column.

1. Age - Age of an individual

2. Workclass - Work class of an individual whether he is a private employee or independently employed regardless of pay.

3. Fnlwgt - 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 estimates of the population 16+ for each state.
* Controls for Hispanic Origin by age and sex.
* Controls by Race, age and sex.

4. Education - It addresses the training of an individual concerning his certification.

5. Education\_num - This section shows the number of trainings.

6. Marital\_status - This section shows the marital status of an individual.

7. Occupation - Occupation of an individual.

8. Relationship - This section shows the relationship of an individual in the family.

9. Race - Race of an individual.

10. Sex - Gender of an individual.

11. Capital\_gain - This section shows how much benefit has been made by an individual.

12. Capital\_loss - This section shows how much misfortune has been made by an individual.

13. Hours\_per\_week - Daily long periods of working.

14. Native\_country - Nationality of an individual.

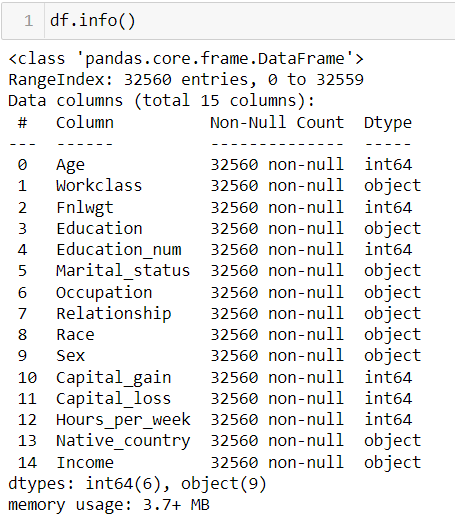
15. Income - Income of an individual. Name in the dataset.

Each Features have some influence in predicting the income of a person. It is necessary to know which features have greater impact on income and which does not have co-relation with label.

* **Pre-Processing Steps**

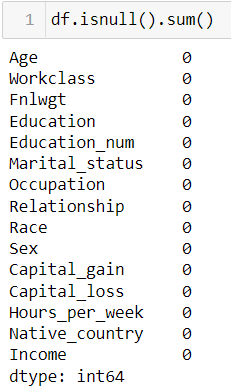
1. Identifying sources of the data
2. Analysing the information
3. Cleaning and handling the information
4. Selecting the most significant elements
5. Writing down findings and observations
6. Using various models to train the data
7. Selecting the best-fitted model for predictions
8. Predicting results for test information

* **Pre-Processing Pipeline.**
* First let’s check the data type of the dataset.

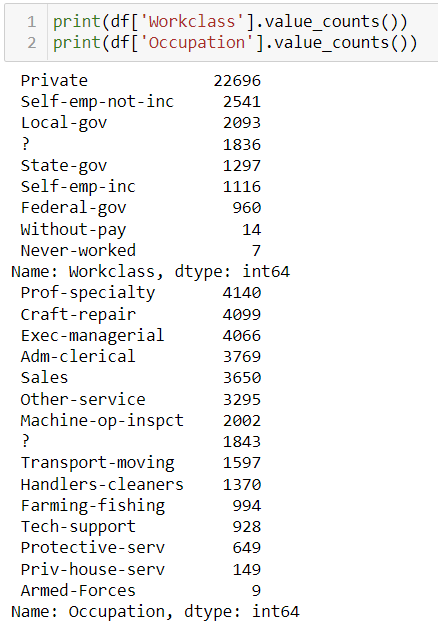


As we can see, Workclass, Education, Marital\_status, Occupation, Relationship, Race, Sex, Native\_country and Income columns are Object type data which needs to change to Integer, since it is important for model building as model does not consider the string value.

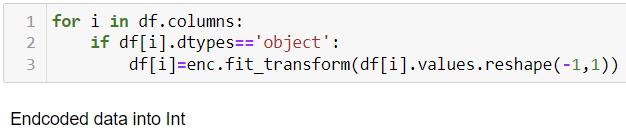
* There are no Null Values present in the dataset so we can move further.



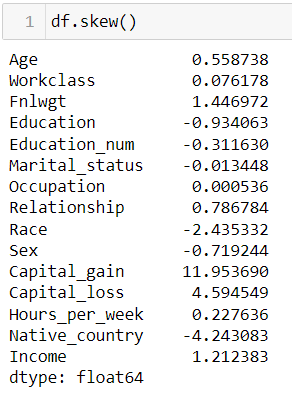
* Sometimes some unwanted things can be found in a dataset which are equivalent to Null values. It is important to take care of such cases.



* Used Mean and Mode method to fill those unwanted characters.
* Encoded data into Integer



* Checked the skewness of the data.

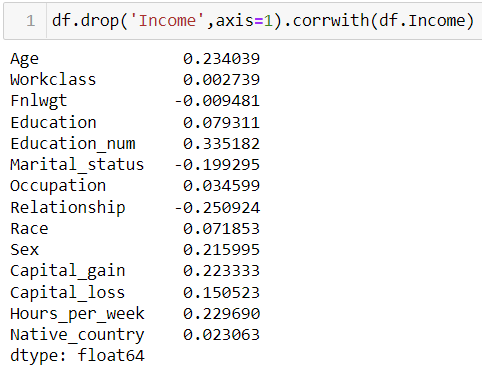


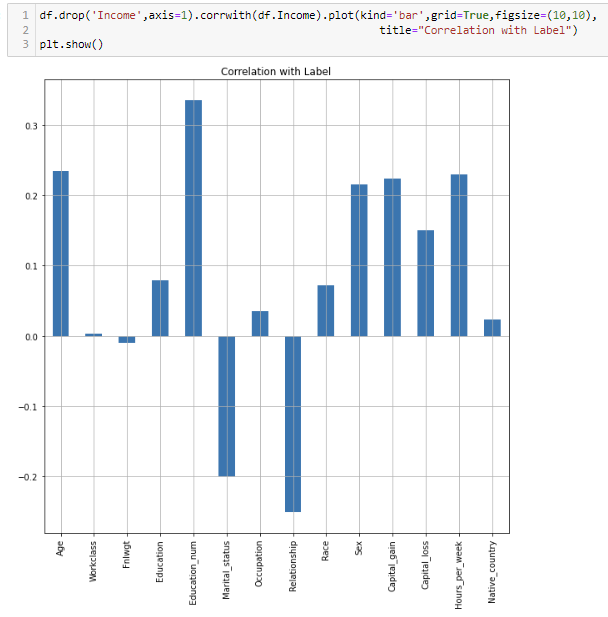


Fnlwgt, Education, Relationship, Race, Sex, Capital\_gain, Capital\_loss and Native\_country has skewness and can have outliers.

Education, Relationship, Race, Sex and Native\_country are the classified columns hence cannot remove the outliers.

* Checked the co-relation with label.





As we can observe in the dataset that 2 columns are inversely proportionate with label. Marital\_status, Fnlwgt and Relationship have inverse relationship with label.



Workclass, Education, Occupation, Race and Native\_country have less or no corelation with label.

Age, Education\_num, Relationship, Sex, Capital\_gain and Hours\_per\_week have the highest co-relation with label.

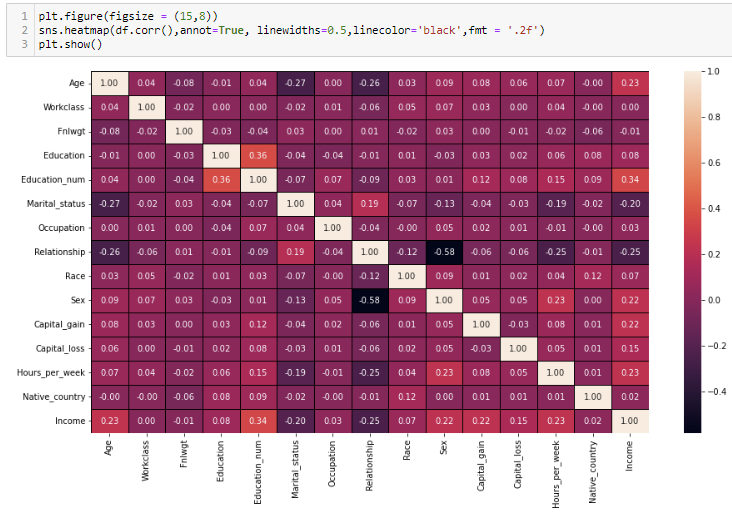
Dropping Workclass, Occupation, and Native\_country columns since it has no co-relation with Label.



**Findings:**

1. It should be visible in the diagram that Education\_num is the significant variable that influences the pay of an individual. Since advanced education can give a job to an individual.
2. Relationship is another significant element that conversely impacts the pay of an individual. In this way, we can presume that an individual who has fewer relationships will procure great.
3. Age likewise influences the pay as higher the age will in general have more experience which brings about higher pay.
4. Hours\_per\_week shows a number of working hours and it has a decent co-connection with pay since a higher quantity of working hours will bring about higher pay.
5. As per the dataset Sex or orientation of the individual likewise has a decent co-connection with the payment of an individual.
6. Capital\_gain shows that the individual has procured any benefit and it affects the pay of an individual as more the capital increase will bring about a big-time salary.

* Checked the Multicollinearity issue in the data.

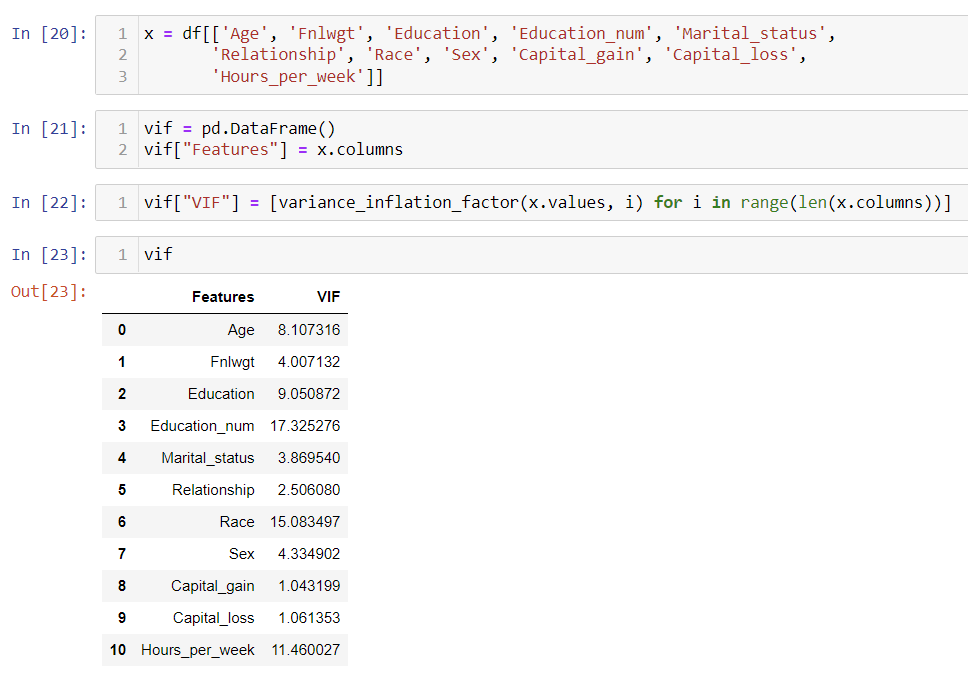


Multicollinearity issue doesn't exist in this data set

Most elevated Multicollinearities exist among Education\_num and Education segments i.e., 36% which is not that great.

* Checked the VIF Score of features

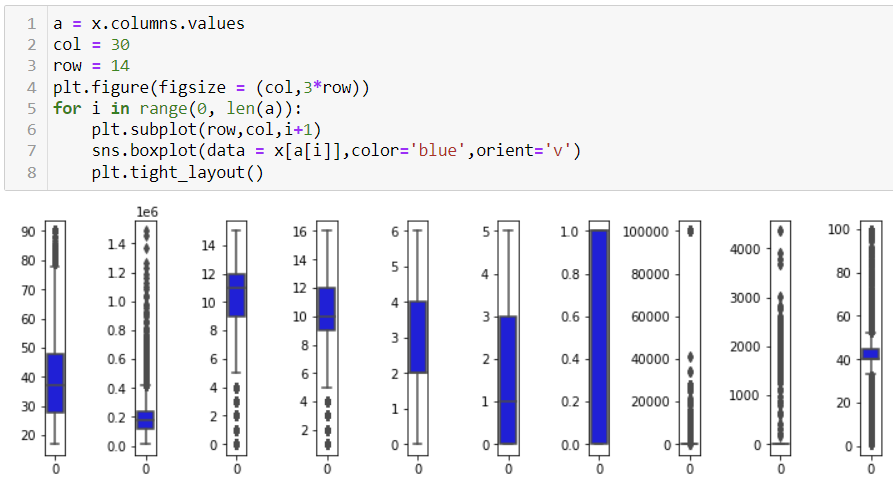
Variance inflation factor (VIF) is a measure of the amount of multicollinearity in a set of multiple features or variables.



Race and Education\_num column has the highest VIF, however Race has low co-relation with Label hence dropping Race column.



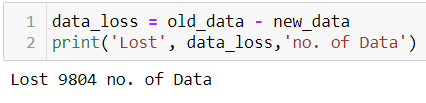
* Checked the outliers in the data



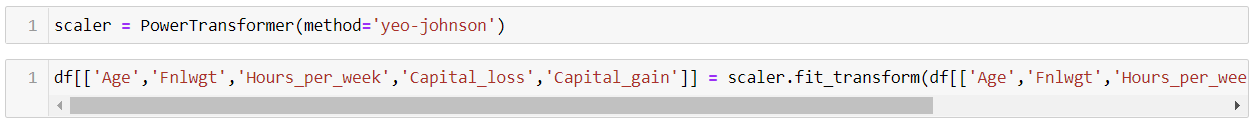
1. Outliers are present in Age, Fnlwgt, Education, Education\_num, Hours\_per\_week, Capital\_loss and Capital\_gain
2. Education and Education\_num are classified columns hence not removing outliers.

* Removed the Outliers

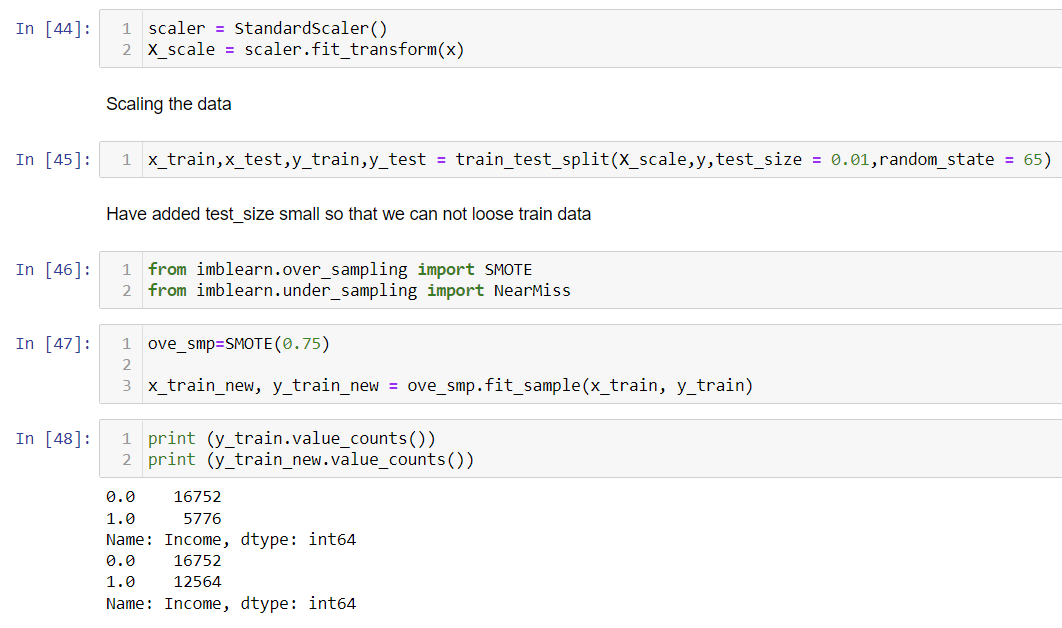
Data Loss due to removing outliers is 9804



* Used power transformation to remove skewness from the data.



* Label data was not balance hence have used SMOTE technique to balance the data after scaling the data



* **EDA Concluding Remark.**

1. The information was not organized and coordinated and subsequently cleaned the information utilizing different information cleaning and pre-handling techniques.
2. There are numerous anomalies present in the information consequently eliminating exceptions
3. There was a skewness in the information thus have eliminated the skewness from the information.
4. There was an irregularity in the information thus have utilized SMOTE strategy to balance the information.
5. Scaled the data utilizing Standard Scalar to make the information normalized to fabricate a model.

* **Hardware and Software Requirements and Tools Used**

1. Libraries and packages used

* import numpy as np - For Numpy work
* import pandas as pd - To work on DataFrame
* import seaborn as sns - Plotting Graphs
* import matplotlib.pyplot as plt - Plotting Graphs
* import pickle – To save the Model
* from sklearn.preprocessing import StandardScaler (To scale the train data), OrdinalEncoder(To encode object data to Integer), PowerTransformer (To remove skewness from dataset)
* from statsmodels.stats.outliers\_influence import variance\_inflation\_factor
* enc = OrdinalEncoder() = Assigned OrdinalEncoder to variable
* from sklearn.model\_selection import train\_test\_split, GridSearchCV, cross\_val\_score,(To split the data into train and test, Search the best parameters, to calculate cross validation score)
* from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix, roc\_curve, roc\_auc\_score - To calculate and analyse model metrics.
* from sklearn import metrics

Models which are used

* from sklearn.ensemble import RandomForestClassifier
* from sklearn.linear\_model import LogisticRegression
* from sklearn.ensemble import GradientBoostingClassifier
* from sklearn.tree import DecisionTreeClassifier
* from sklearn.neighbors import KNeighborsClassifier
* from sklearn.svm import SVC
* import warnings
* warnings.filterwarnings('ignore') - To ignore unwanted Warnings

1. Hardware used – 11th Gen Intel(R) Core (TM) i3-1115G4 @ 3.00GHz 3.00 GHz with 8.00 GB RAM and Windows 11
2. Software used – Anaconda and Jupyter Notebook to build the model.

* **Building Machine Learning Models.**

I have built 6 machine learning models to predict the label. Below are the machine learning models which are been used.

1. LogisticsRegression
2. RandomForestClassifier
3. DecisionTreeClassifier
4. GradientBoostingClassifier
5. Support Vector Classifier
6. KNeighborsClassifier
7. **LogisticsRegression:**

Have used “For Loop” to find out the highest accuracy score with different random state ranging from 0-100 and using that random state to split the data into train and test data.



Have used “Define Function” to define a machine learning model code that automatically provides the train and test accuracy code.

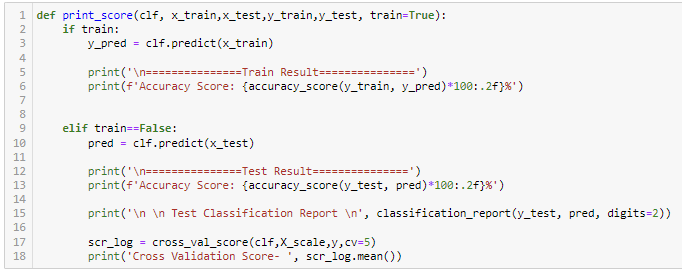
Formulas:

y\_pred = clf.predict(x\_train) = Predicting train data

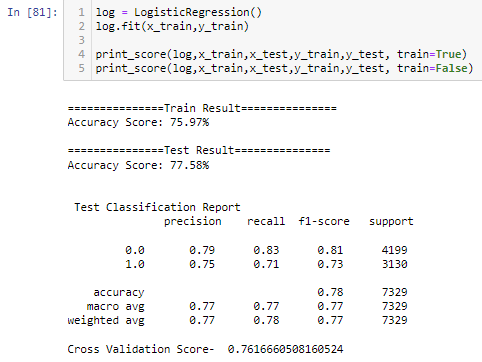
accuracy\_score(y\_train, y\_pred) = Calculating train accuracy score (comparing y\_pred data with y\_train data)

pred = clf.predict(x\_test) = Predicting test data

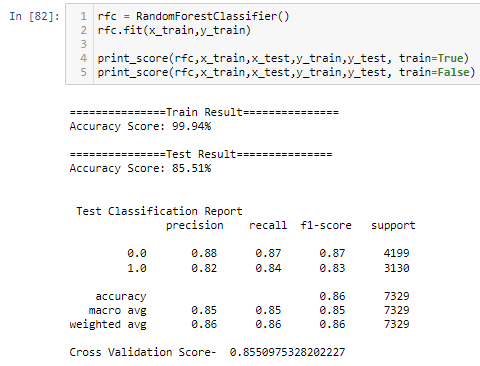
accuracy\_score(y\_test, pred) = Calculating test accuracy score (comparing pred data with y\_test data)



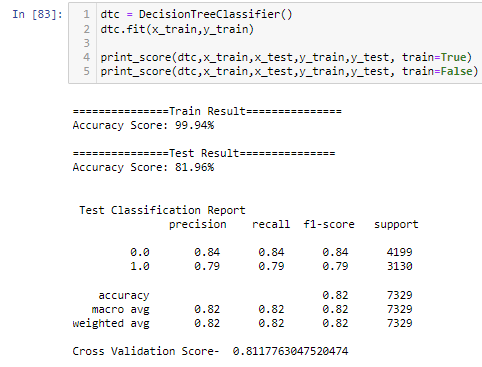
Trained the data and run the “Def” function



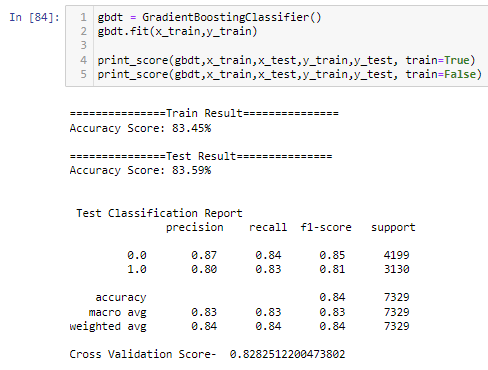
1. **RandomForestClassifier:**

****

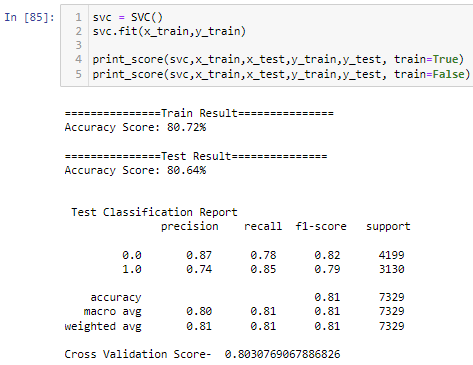
1. **DecisionTreeClassifier:**

****

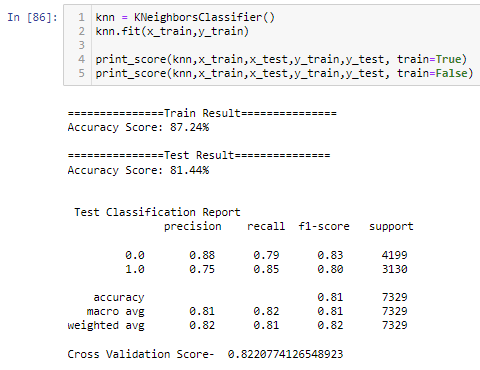
1. **GradientBoostingClassifier:**

****

1. **Support Vector Classifier:**

****

1. **KNeighborsClassifier:**

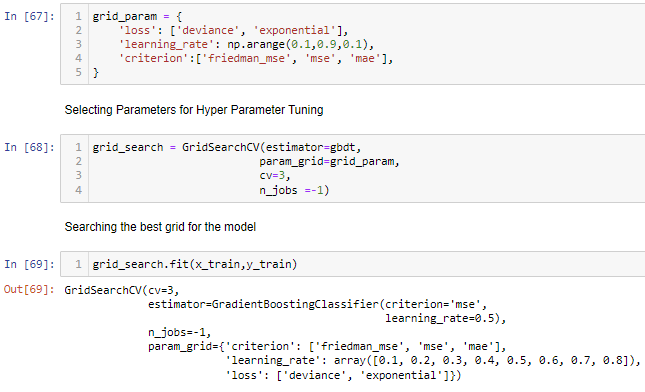
****

* **Findings**
  + **LogisticsRegression** - Cross Validation Score is 76.16%, Accuracy Score of Train Result is 75.97% and Test Result is 77.58%
  + **RandomForestClassifier** - Cross Validation Score is 85.50%, Accuracy Score of Train Result is 99.94% and Test Result is 85.51%
  + **DecisionTreeClassifier** - Cross Validation Score is 81.17%, Accuracy Score of Train Result is 99.94% and Test Result is 81.96%
  + **GradientBoostingClassifier** - Cross Validation Score is 82.82%, Accuracy Score of Train Result is 83.45% and Test Result is 83.59%
  + **Support Vector Classifier** - Cross Validation Score is 80.30%, Accuracy Score of Train Result is 80.72% and Test Result is 80.64%
  + **KNeighborsClassifier** - Cross Validation Score is 82.20%, Accuracy Score of Train Result is 87.24% and Test Result is 81.44%
* **Model Selection:**

Selecting GradientBoostingClassifier as it has low variance between train and test result and has high accuracy i.e., 83.45% and 83.59% respectively.

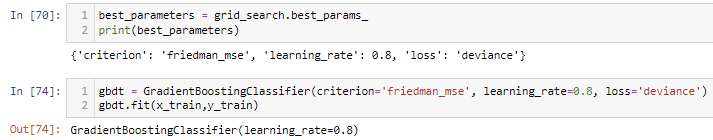
* **Hyper Parameter Tuning:**

In machine learning, hyperparameter tuning is the issue of picking a bunch of ideal hyperparameters for a learning calculation. hyperparameter tuning boundaries rely upon the choice of the model, as the model changes the parameters additionally change. A hyperparameter is a parameter whose worth is utilized to control the learning experience. According to the choice of the GradientBoostingClassifier model, we should tune the model to expand the train and test precision.

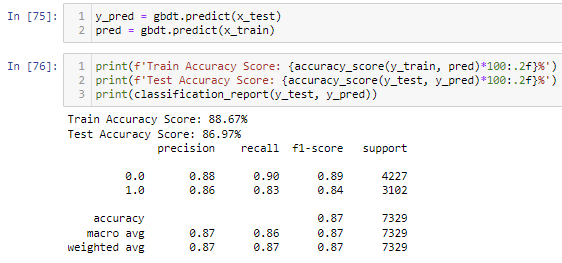
Have referenced the parameters and utilized Grid Search CV to find the best mix of parameters that will build the precision.

Have not used more parameters as they take more time to train and due to slow hardware, it gets impossible to train the model.

Used best fitted parameters to train the model.

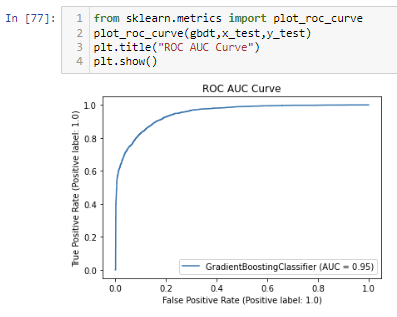


**Model after Hyper Tuning:**



* Previous Train accuracy score was 83.45% and new Train accuracy score is 88.67%
* Previous Test accuracy score was 83.59% and new Test accuracy score is 86.97%
* **ROC AUC Curve:**

AUC – ROC (Area Under Curve - Receiver operating characteristic) curve is a performance measurement for the classification issues at different limit settings. ROC is a probability curve and AUC addresses the degree or proportion of detachability. It tells how much the model is fit for recognizing classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By relationship, the Higher the AUC, the better the model is at recognizing income above $50K and income below $50K.

****

AUC score is 95% which is pretty good.

* **Concluding Remarks.**
* **Key Findings and Conclusions of the Study**

1. Selecting GradientBoostingClassifier model since the Accuracy score i.e., 83.45% and test scores i.e., 83.59% are greater and close to each other.

2. After tuning the model the train and test accuracy score increased to 5% in train data and 3% in test data.

3. AUC score is also high i.e 95%.

* **Saving the Model**

Saving the selected model after hyper parameter using pickle.



You can find the Model [here](https://github.com/AniruddhaSawant/MLModels/blob/main/Census_Income_Project.ipynb).

* **Learning Outcomes of the Study in respect of Data Science**

1. Data Cleaning assists with changing over sloppy and unstructured data into organized data which will be utilized to make discoveries.
2. Data visualization gets it and dissects the information.
3. Model structure assists with anticipating results, for this situation GradientBoostingClassifier model fits ideal for this dataset.
4. This model can be utilized in different use cases like publicizing, marketing, research, lead generation, advertising, promoting, finance, etc.

* **Limitations of this work and Scope for Future Work**
  1. There are 14 features present in the dataset anyway due to pre-handling and representation we have cut down a few elements, consequently this could turn into a disadvantage in the future as we update the dataset and there is the possibility that we might lose some significant data.
  2. It is important to watch out for new and refreshed information to further train the model and settle on choices according to new information.