Machine Learning application — Census Income Prediction

The economic well-being of a Nation is highly driven by the income of the residents.

Countless decisions in private and public sectors are based on Census data. Census data is the backbone of the democratic system of government, highly affecting the economic sectors. Census-related figures are used to distribute the federal funding by the government into different states and localities.

Not only the above, the census data is also used for post census population estimates and projections, economic and social science research, and many other such applications. Hence, the importance of this data and its correct predictions is very clear to us.

Data has always been the backbone of many important decisions. When an assumption is backed up by facts and numbers, the chances of incorrectness and bad decisions decrease.

# Problem Statement

The above introduction had an aim to increase the awareness about how the income factor actually has an impact not only on the personal lives of people, but also an impact on the nation and its betterment.

We will today have a look on the data extracted from the 1994 Census bureau database, and try to find insights about how different features

have an impact on the income of an individual. Though the data is quite old, and the insights drawn cannot be directly used for derivation in the modern world, but it would surely help us to analyze what role different features play in predicting the income of an individual.

# The Dataset

The dataset provided to us contains 32560 rows, and 14 different independent features. We aim to predict if a person earns more than 50k$ per year or not. Since the data predicts 2 values (>50K or

<=50K), this clearly is a classification problem, and we will train the classification models to predict the desired outputs.

Mentioned below are the details of the features provided to us, which we will be feeding to our classification model to train it.

1. Age — The age of an individual, this ranges from 17 to 90.
2. Workclass — The class of work to which an individual belongs.
3. Fnlwgt — The weight assigned to the combination of features (an estimate of how many people belong to this set of combination)
4. Education — Highest level of education
5. Education\_num — Number of years for which education was taken
6. Marital\_Status — Represents the category assigned on the basis of marriage status of a person
7. Occupation — Profession of a person
8. Relationship — Relation of the person in his family
9. Race — Origin background of a person
10. Sex — Gender of a person
11. Capital\_gain — Capital gained by a person
12. Capital\_loss — Loss of capital for a person
13. Hours\_per\_week — Number of hours for which an individual works per week
14. Native\_Country — Country to which a person belongs Output:

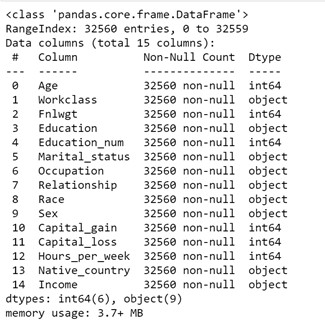
1. Income — The target variable, which predicts if the income is higher or lower than 50K$.

# Contents of the article

The following information/steps will be covered further in the article –

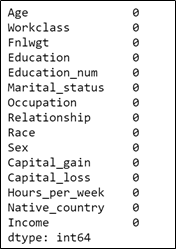
1. Exploratory data analysis
2. Data modeling
3. Outlier detection and skewness treatment
4. Encoding the data — Label Encoder
5. Scaling the data — Standard scaler
6. Fitting the machine learning models
7. Cross-validation of the selected model
8. Model hypertuning
9. AUC-ROC curve
10. Saving the final model and prediction using saved model

# Exploratory Data Analysis

The first step that we do is to check the information about our data. We see the results shown in the image below:

We see that we have a mixture of categorical and numeric columns. We have 6 integer columns and 9 object type columns. We observe that the count of entries is 32560 for all columns, hence no NaN values are present in our dataset.

We confirm this assumption using data.isnull().sum() command –



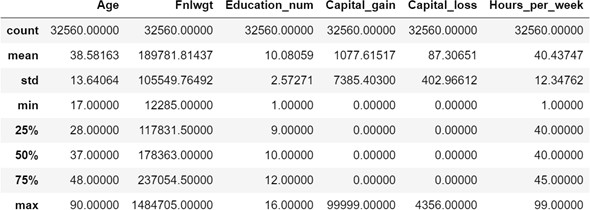
But, while having a close look at our dataset, we observe some of the

values as ‘?’, which represent missing values.



Hence, we deduce that there are some values in our data set which need to be treated.

We also check the numerical statistics of our data using data.describe() command –



Following observations are made in this step –

* The age column has a range of 17 to 90.
* The fnlwgt column has a minimum value of 12285 and maximum value of 1484705
* The education number has a range of 1 to 16
* The capital gain starts from 0 and ends at 99999
* The capital loss starts at 0 and ends at 4356
* Hours per week range between 1–99.
* There are outliers expected in Capital gain column as the values till 75% are 0. Same is the case with capital loss as well.
* The fnlwgt column also has a huge difference between 75% values and the max value. There is a chance of getting outliers here.

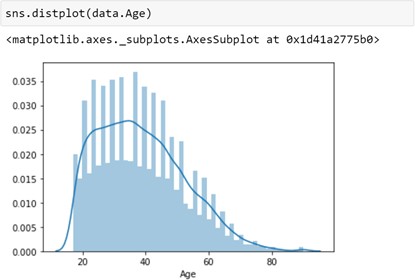
Further, we have a look at our dataset and explore the data in various columns, one by one –

# Income column

The income column is our target variable with 2 values — ‘<=50K’ and ‘>50K’. The count of these values is 24719 and 7841 respectively, suggesting that people with income higher than 50K are significantly less, and our data set is imbalanced considering the target variable.

# Age column

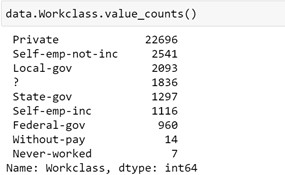
The data in age column has a minimum value 17 and max value 90. We create a distribution plot for the age column –

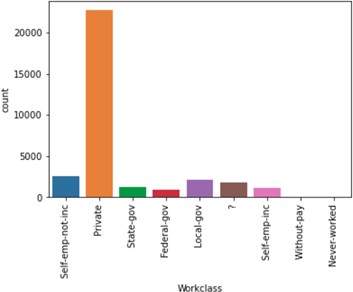


We observe that our data is has right skewness, with majority of the ages falling in the 20–50. The count keeps on decreasing as the age increases.

We also observe that we do not have any null values in the age column.

# Workclass column

While checking the unique values for workclass, we see that we have 7 different types of values, along with some missing values represented by ‘?’. The count of null values is 1836, which is around 5% of the data.



We observe that majority of the people belong to ‘Private’ sector

workclass.

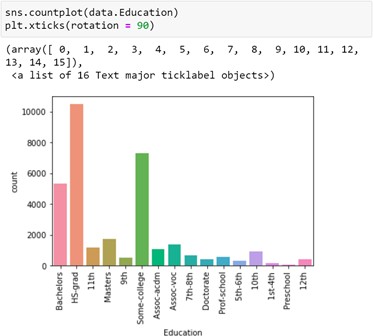


We also make an interesting discovery here — The values where

‘Workclass’ is missing, also has ‘Occupation’ missing!

# Education column

The ‘Education’ column has 16 different categories available. Majority of these categories belong to ‘School’ type (different classes are divided into multiple categories)



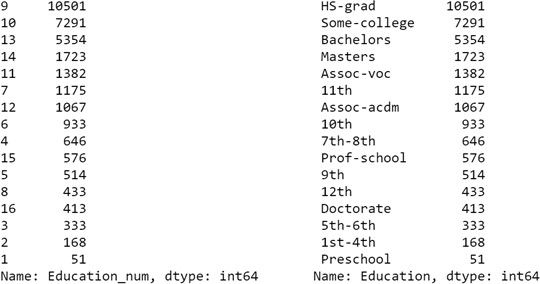
We observe no missing values in this column, and also find out that majority of the people have education level as ‘HS-grad’, followed by ‘Some-college’ and ‘Bachelors’.

# Education-num column

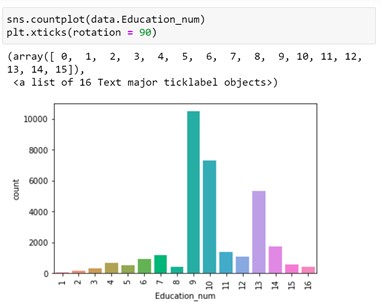
The education number is the number of years for which a person received education. This is an ordinal column, which contains 16 different values. When we check the division of ‘Education\_num’ column, we observe that the count of ‘Education’ column and ‘Education\_num’ is exactly same! Which means, the ‘Education\_num’ column is providing same information as ‘Education’ column, but in a numeric manner!

We can map the ‘Education’ level and ‘Education\_num’ columns by

keeping the information side-by-side as is displayed below:



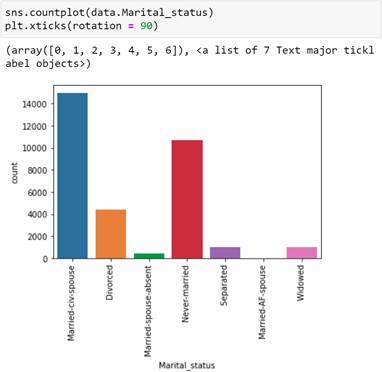
The count chart for ‘Education\_num’ is as follows –



# Marital\_Status column

The ‘Marital\_Status’ column has 7 different categories available, and

has no missing values.

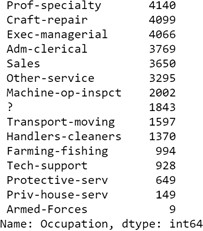


Majority of the people have ‘Marital\_Status’ as ‘Married-civ-spouse’, and least have ‘Married-AF-spouse’.

Count of ‘Never-married’ is also quite high.

# Occupation column

The occupation column contains 14 different categories, and have missing values represented by ‘?’ (which we have already observed, and combined with ‘Workclass’ column).



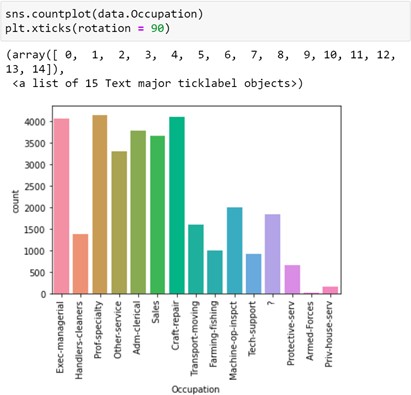
But the count of missing values is slightly higher than ‘Workclass’

column — 1843.

We try to find out the extra rows where the ‘Occupation’ is missing –

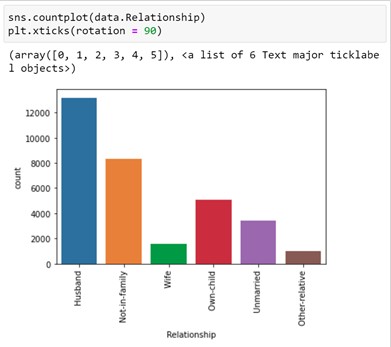


We observe that in case the ‘Workclass’ is ‘Never-worked’, then also the ‘Occupation’ is missing.



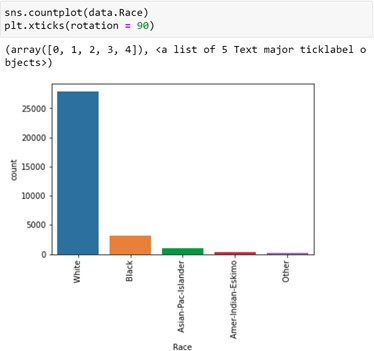
# Relationship column

The relationship column contains 6 different types of values, with highest number set for ‘Husband’ and lowest for ‘Other-relative’. The column does not have any missing value.

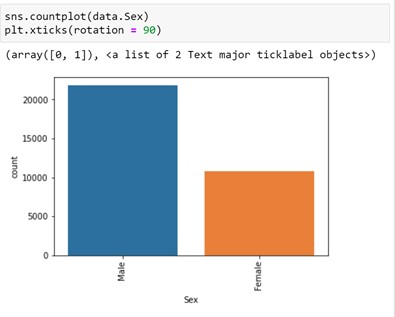


# Race column

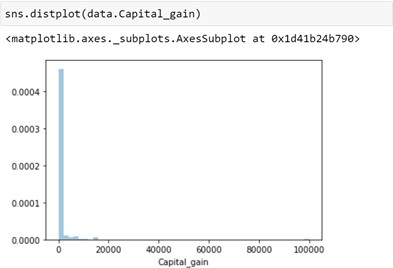
The Race column has 5 different categories, and no missing data. Highest number of people have race as ‘White’ (significantly high numbers).



# Sex column

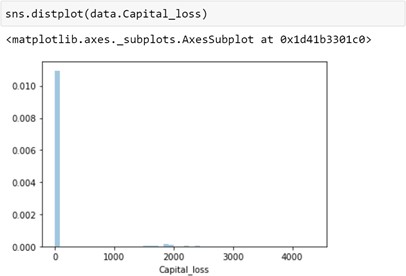
The ‘Sex’ column has 2 categories — Male and Female, where number of males are almost double to number of females. Missing values are not found in this column.

# Capital gain column

‘Capital\_gain’ column is a numeric column, with majority of the values set as 0. The distribution plot for ‘Capital\_gain’ column is highly right skewed.

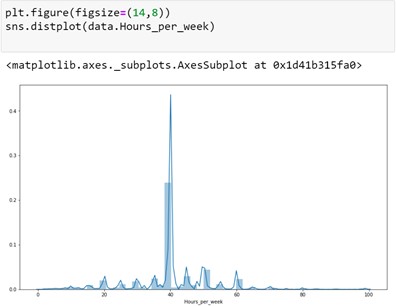
# Capital Loss column

The ‘Capital\_loss’ column also has majority of the values set as 0, similar to ‘Capital\_gains’. The data is highly right skewed in this case as well.



# Hours per week column

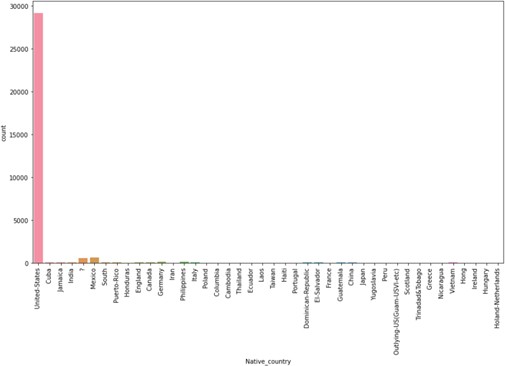
The hours per week column has values scattered over a range of 1–99. The column does not have any missing values. Majority of the values have data near 40 hours and hence a high peak can be observed in the below distribution plot –



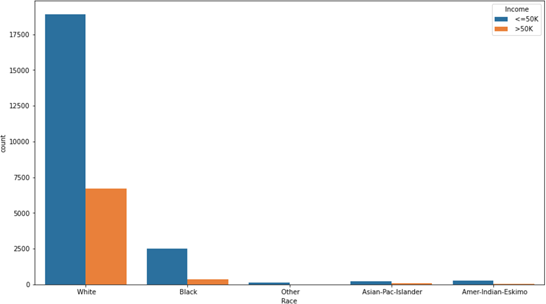
# Native\_country column

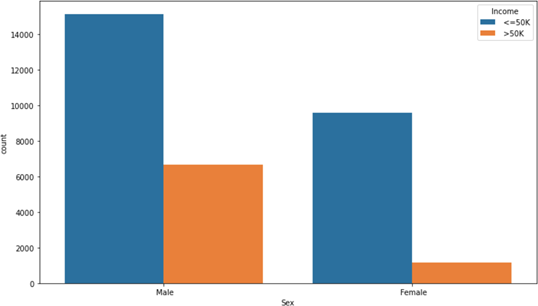
The Native\_country column contains the highest count set to ‘United- States’, and rest of the rows contain quite few numbers (highest count after US is 643).

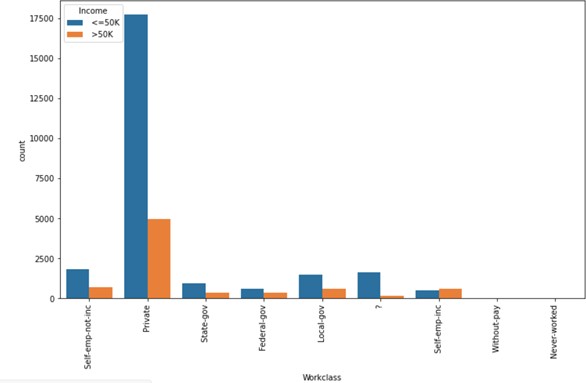
We also have 583 missing values in this column, which we need to treat.

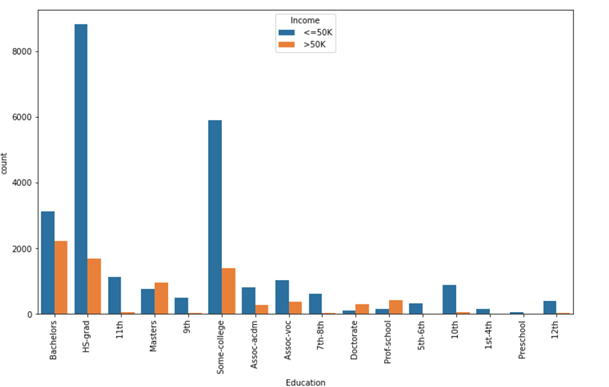


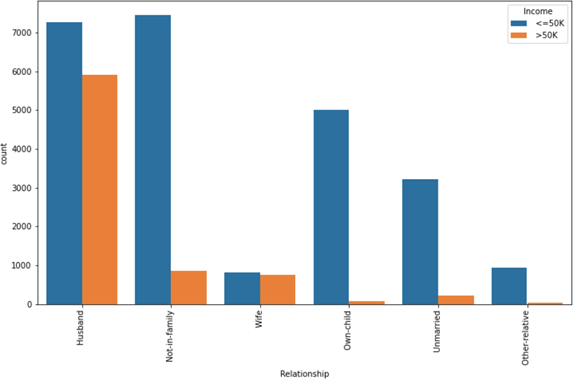
We further check how the income gets impacted due to the features which we just explored.

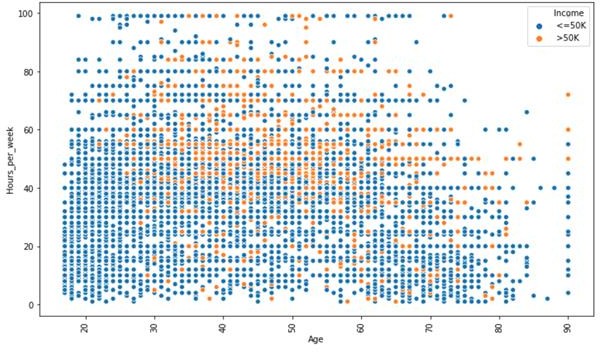












Looking at the graphs above, we make the following conclusions –

1. A person has high chance of earning >50K in case his/her race is

‘White’/’Asian-pac-islander’.

1. Males have a higher chance of earning more than 50K, than females.
2. Ratio of people earning more than 50K is higher in case Workclass

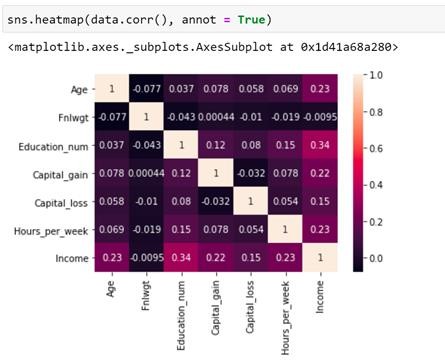
is ‘Self-emp-inc’.

1. People with education level as ‘Masters/Doctorate/Prof-school’ have higher ratios of >50K earning, than <=50K. Bachelors degree also has around 10:7 ratio of <=50K : >50K.
2. If the relationship in family is either ‘Husband/Wife’, the chances of

earning more than 50K is high.

1. From the scatterplot between age, hours\_per\_week and income, we observe that a person needs to be >30 to be earning more than 50K, else needs to work at least 60 hours\_per\_week to earn >50K.

If we check the correlation between the numeric columns, we observe that –



Income has 34% correlation with ‘Education\_num’, 23% correlation with ‘hours\_per\_week’ and ‘age’, and 22% correlation with ‘Capital\_gain’. The correlations are moderate.

# Data modeling

We now proceed to an important part of our process — data modeling. Based on our analysis above, we will fill the missing values in our data, and group certain categories logically, to allow our model to learn better.

# Replacing missing values –

We choose to replace the missing values with the mode of the data, i.e. the most frequently occurring values.

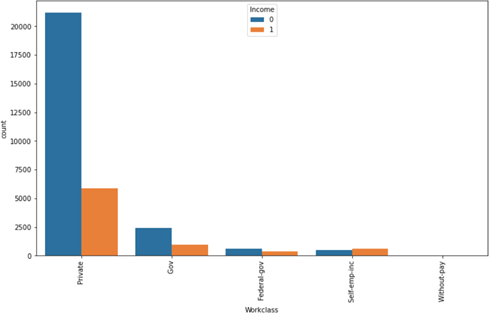
Hence, we replace ‘?’ is ‘Workclass’ column by ‘Private’, ‘Occupation’ column by ‘Prof-speciality’ and ‘Native\_country’ by ‘United\_States’.



# Combining the data logically to reduce categories -

Further, we combine the data in various columns –

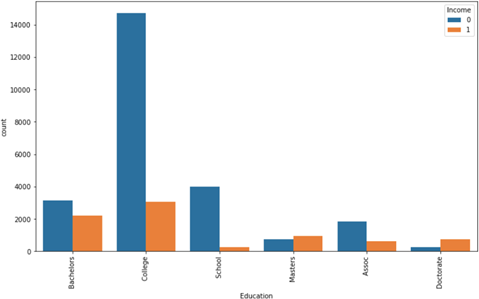
* Workclass –



We put ‘Never-worked’ and ‘Without-pay’ in one category, we classify ‘State-gov’ and ‘Local-gov’ as ‘Gov’, and we add ‘Self-emp-not-inc’ in ‘Private’ category, since the distributions are similar.

We obtain 5 different categories in Workclass column

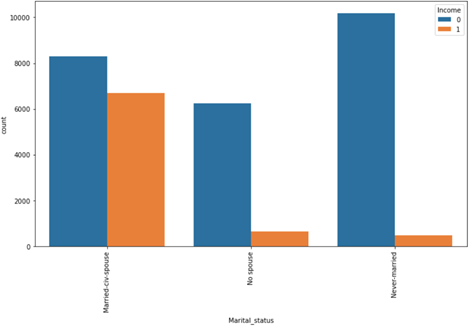
* Education –



We combine all the columns relevant to schools in ‘School’ category, put ‘Doctorate’ and ‘Prof school’ in a single category ‘Doctorate’, ‘Assoc-acdm’ and ‘Assoc-voc’ in one category ‘Assoc’, and ‘HS-Grad’ and ‘Some-college’ in one category ‘College’.

We now obtain 6 categories of education which we feed to our machine learning model.

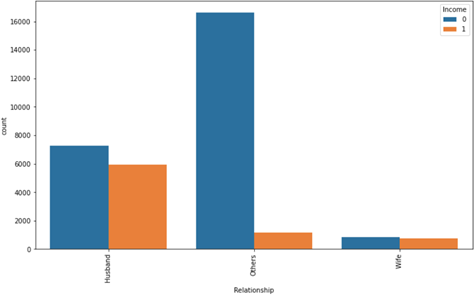
* Marital Status -



We combine ‘Divorced’, ‘Married-spouse-absent’, ‘Separated’, ‘Widowed’ and ‘Married-AF-Spouse’ to one category and name it as ‘No spouse’.

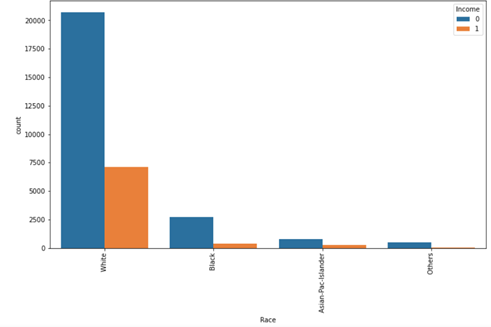
We now obtain 3 categories.

* Relationship column -



We combine ‘Not-in-family’, ‘Own-child’, ‘Unmarried’ and ‘Other- relative’ columns to a single category looking at the distributions, and name is as ‘Other’.

* Race column -

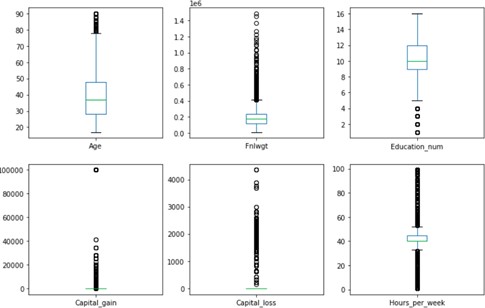


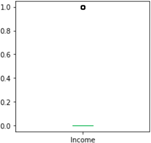
We combine the categories ‘Amer-Indian-Eskimo’ and ‘Other’ to ‘Others’ category, since they have similar distributions.

Our data categorization is now done. The next step is to identify if our data has any outliers, and deal with them.

# Outlier Detection

We further proceed to detect outliers in our data and decide how to deal with them. The best way to interpret outliers using visualizations is boxplot, hence we plot boxplots for our numerical columns, which result in the below visuatization –





We observe outliers in all the numeric columns. We use Zscore values to confirm how much data, out of the complete dataset, falls in an outlier range.

‘***Z-score (also called a*** *standard score****) gives an idea of how far from the mean a data point is.*** *But more technically it’s a measure of how many standard deviations below or above the population mean a raw score is.’*

After calculating the number of rows containing outliers, we find that we would lose 2733 rows taking a threshold value of zscore as 3. This

data has 1546 rows with census income less than 50K$ and 1187 rows with higher income than 50K$. Since our dataset is already imbalanced, losing this number of rows with further increase the imbalance, and would be a significant loss if we consider the rows with income higher than 50K$.

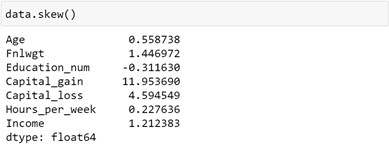
Hence, we keep the rows and proceed with the next steps.

# Skewness treatment

We now proceed with treating skewness in our data, which allows us to fit

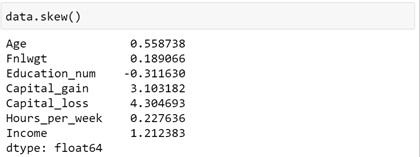
our data in a symmetric distribution, which further allows our model to learn better.

The skewness of the data without any transformation is –



We treat ‘Fnlwgt’, ‘Capital\_gain’ and ‘Capital\_loss’ column for skewness, and use square-root transform and cube-root transform methods (since we cannot apply log and boxcox transform to columns where 0 values are present)

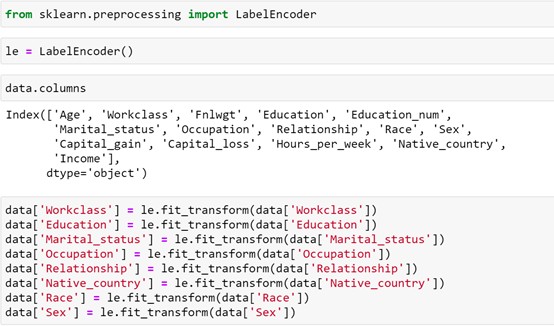
The final skewness that we receive after multiple transformations (sqrt transform for Fnlwgt column, and cbrt transform for Capital\_gain and Capital\_loss columns twice) –



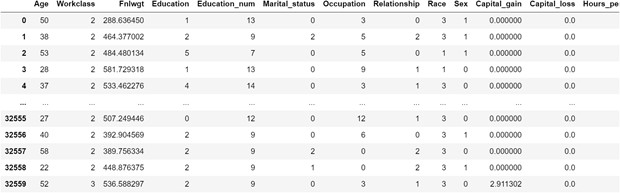
We further proceed to next steps since the skewness does not get further decreased.

# Encoding the data

Since majority of the classification models need input as ‘int/float’, and do not work on ‘string’ data, we encode our categorical columns using ‘Label Encoder’



The final dataset looks something like –



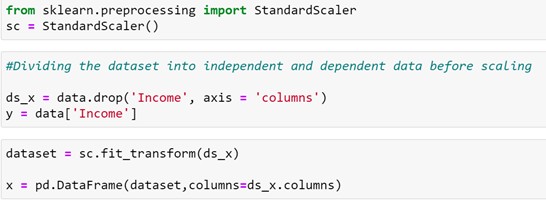
# Scaling the data

The next step is to bring the data to a common scale, since there are certain columns with very small values and some columns with high values. This process is important as values on a similar scale allow the model to learn better.

We use standard scaler for this process –

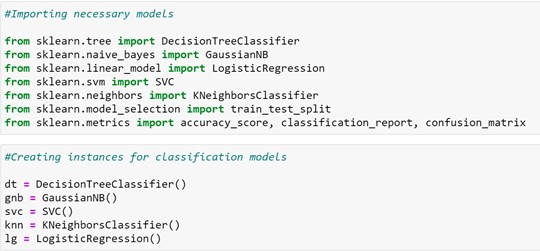
***‘****StandardScaler follows Standard Normal Distribution (SND).*

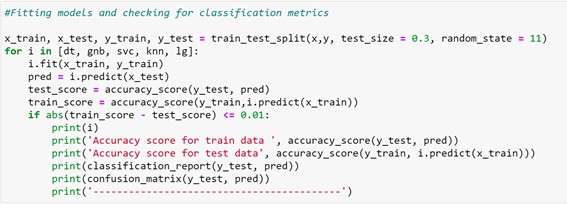
*Therefore, it makes mean = 0 and scales the data to unit variance’*



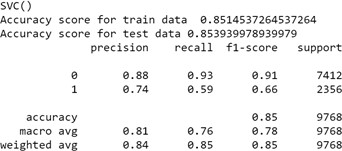
# Fitting data into classification models

We now proceed to the main step of our machine learning, fitting the model and predicting the outputs. We fit the data into multiple

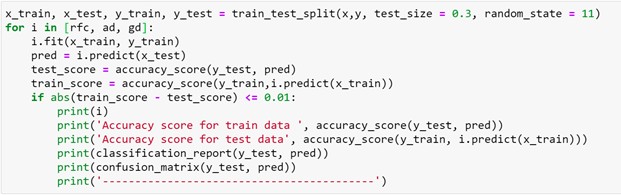
classification models to compare the performance of all models and select the best model –

We use the below mentioned code snipped to fit the data into ML models and predict the output –

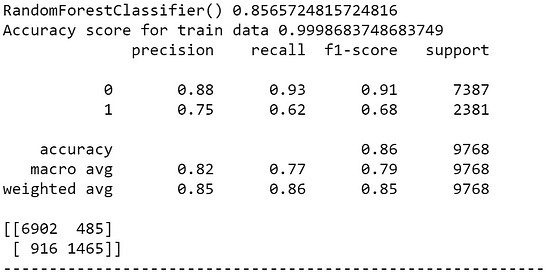
We achieve the best results using — ‘Support Vector Classifier’, which provides an accuracy of 85% on test data. Below is the classification report for the SVC model –

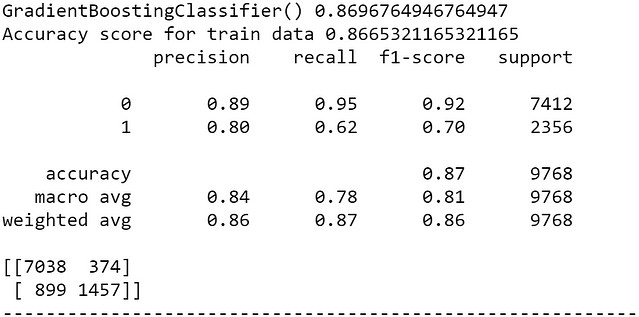


We further try fitting the data to classification models to check how our ensemble models perform on the given dataset.



As a result, we observe that Random Forest Classifier over fits the train set, and gives a 99% accuracy on train data.





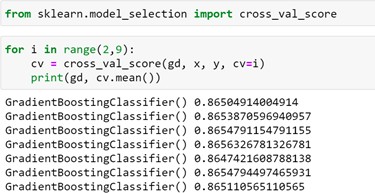
We see that the Gradient Boosting Classifier gives us an accuracy of

~87% (higher than SVC), and the f1-score, recall and precision scores also improve. ***Hence we choose ‘Gradient boosting classifier’ as our final model***, and proceed with hypertuning the model. But before this, we perform k-folds cross validation on our dataset.

# Cross validation

The **goal of 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 (i.e., an unknown dataset, for instance from a real problem).

We obtain the following results using cross-validation –



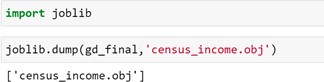
This helps us interpret that the model is not overfitting and will perform well for new data that we feed to our model. We now proceed with hypertuning the model, using GridSearch CV.

# Hypertuning the model

We receive the best params for our model, which result in a best score of 87.7%. We increased our model accuracy by 1% using hypertuning.

We now save the model with best parameters that we identified using

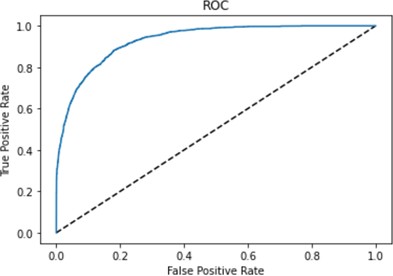
GridSearch, and create an object for our model using ‘Joblib’.



# AUC ROC curve

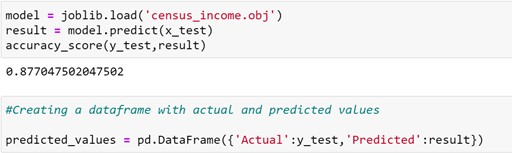
AUC — ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0’s as 0’s and 1’s as 1’s.

We draw the AUC-ROC curve to obtain the following output –



The area under the ROC curve represents the ability of our model to predict correct values, and the curve that we got is quite a good score.

# Conclusion

We further proceed to test the object that we saved using joblib, and create a dataframe of predicted values –

Following are the results that we achieve, with an accuracy of 87.7%.



This marks the end of our process; we have successfully trained our model to predict the income of a person, with an accuracy of ~88%.

We moved step by step, analyzing, cleaning and modeling the data, and applied various machine learning models to achieve the desired predictions. We also tuned the model to improve the accuracy, and were able to achieve a model with quite a good accuracy.