# **Logistic Regression:**

Here the Problem statement is about to Predict (Y) variable whether the client has subscribed a term deposit or not it is Binomial variable we need to predict 1 or 0.

Data Preprocessing:

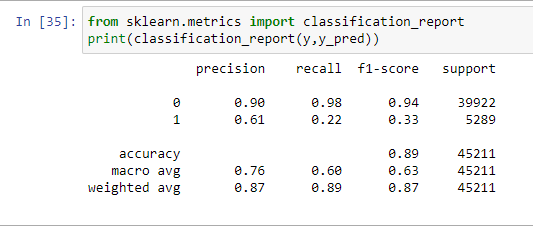
Dimensions of the data are 🡺 (45211, 32)

Checking for Missing values and Nan values:

* In the data set they are No missing values and No Nan Values in the data set
* Here I use Count Plot which shows the count of observations in Each categorical bin using bars
* Dividing the data set into x and y

Model Creation:

* from sklearn.linear\_model import LogisticRegression .
* predicting the values
* checking for the accuracy 🡺89
* Classification report



By considering the Classification report we know that the target variable (“Y”) is Imbalanced data with the values of 0🡺39922, and

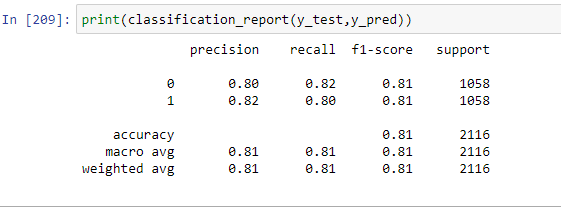
1🡺5289 which is Not valid to Preform model using this type of data

By this type of data even **Accuracy is not Considered**.

* Here we can see that Recall and f1-Score of 0 are high enough but when compared to 1 which is pretty less this is all because of imbalanced Data
* To Encounter the imbalanced data, we use some techniques to solve this kind of problems

**Under Sampling:**

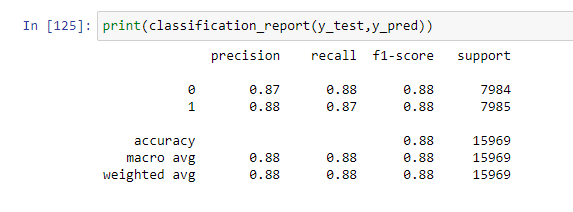
* Under Sampling is a technique which is used in imbalanced datasets which helps in take Random records into consideration from the majority class
* By performing this technique, I got the classification Report as



* You can observe the values of Recall. F1-score, Precision which was improved as compared to previous
* By using the Under Sampling Technique, I Got the Accuracy, Precision, recall and f1-score very effective and efficient to the model

**SMOTE 🡺Synthetic Minority Over-Sampling Technique**

* It Generates Synthetic Examples using the KNN algorithm but I was used using Logistic Regression
* By performing the SMOTE technique, we got the accuracy as 🡺0.88
* After performing the Classification Report



* By using the SMOTE technique, the precision, recall, f1-score, support the values are 0.88
* By comparing with both SMOTE and Under Sampling we got best values for SMOTE