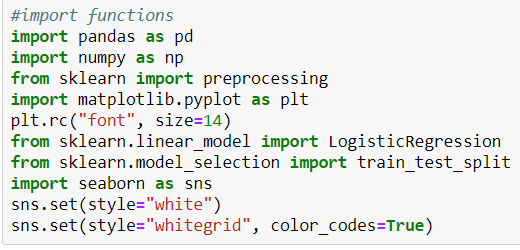
**Write up – On explaining the sourced code**

**Title: Credit card approval**

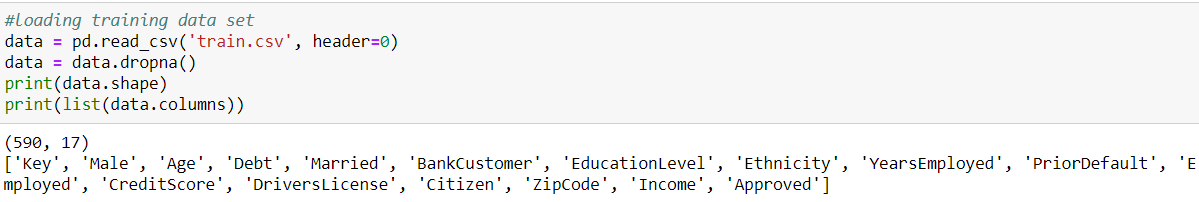
**Author: Jayathilaga**

**Data:**

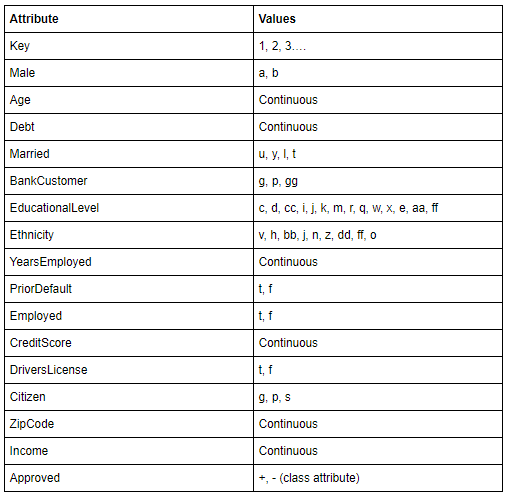
The dataset is related to Credit card approval with the customers information. The classification goal is to predict whether the client will be approved (+/-) to credit card (variable y).



The dataset provides the bank customers’ information. It includes **590 records and 17 fields.**



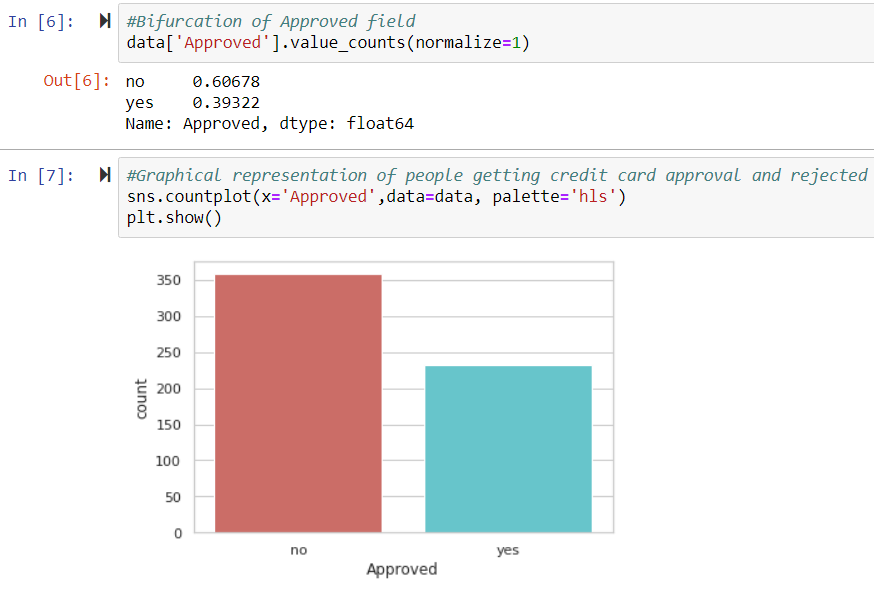
**Input variables:**

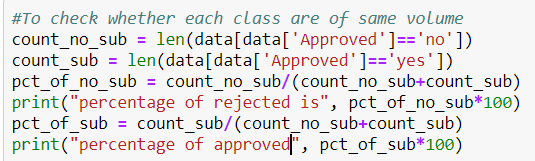


**Predict variable (desired target):**

y — has the client is approved? (binary: “+”, means “Yes”, “-” means “No”)

**Data exploration:**

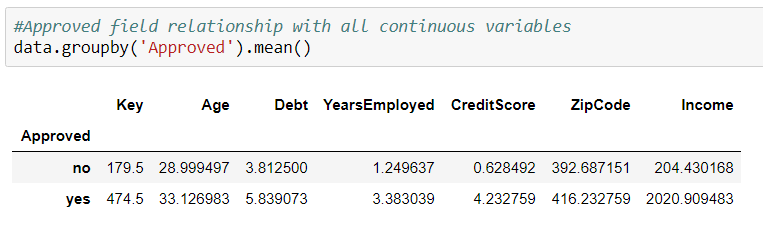




percentage of rejected is 60.67796610169491

percentage of approved 39.32203389830509

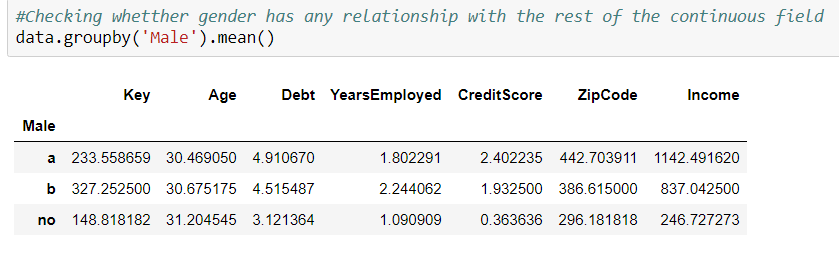
Classes are imbalanced, and the ratio of rejected to approved instances is **61:39.** Before balancing the classes, Here are some more exploration.



**Observations:**

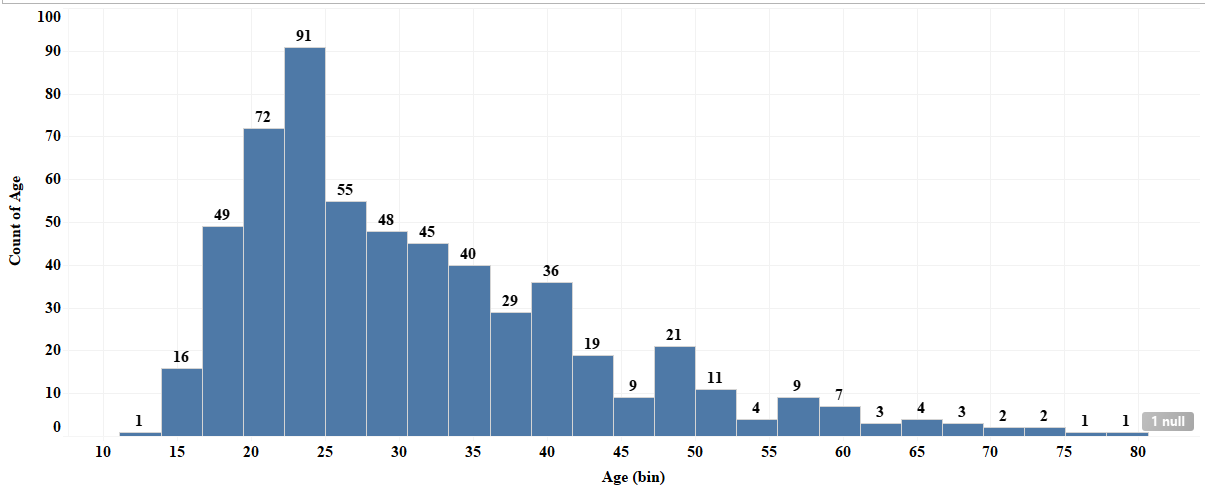
* The average age of customers who is provided credit card is higher than that of the customers who didn’t.
* Years employed for who credit card is not approved is less than that of the customers who got approved.
* Surprisingly, **Debt is higher for the customers** for who the credit card is approved

Categorical means for other categorical variables such as education and marital status to get a more detailed sense of our data.



**Visualizations:**

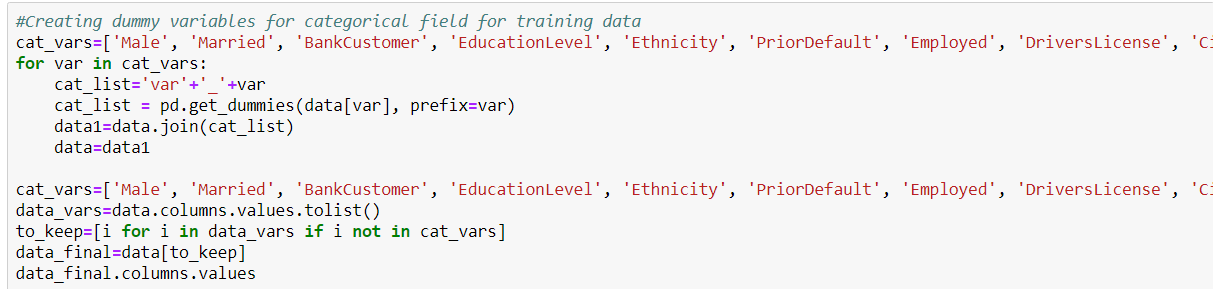
**Historgram of Age**



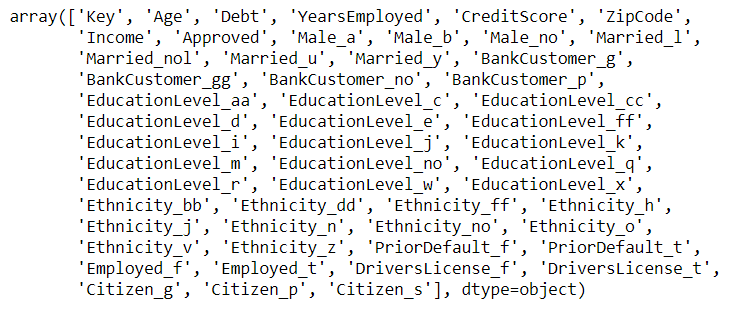
Most of the customers of the bank in this dataset are in the age range of **23-25.**

**Create dummy variables:**

That is variables with only two values, zero and one.



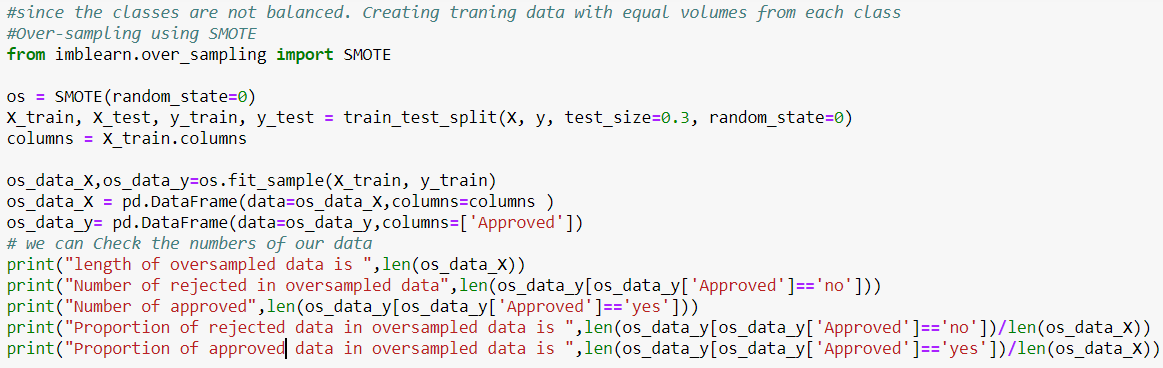
Final data columns will be:



**Over-sampling using SMOTE**

With training data created, I’ll up-sample the rejected using the SMOTE algorithm(Synthetic Minority Oversampling Technique). At a high level, SMOTE:

* Works by creating synthetic samples from the minor class (rejected) instead of creating copies.
* Randomly choosing one of the k-nearest-neighbors and using it to create a similar, but randomly tweaked, new observations.



length of oversampled data is 516

Number of rejected in oversampled data 258

Number of approved 258

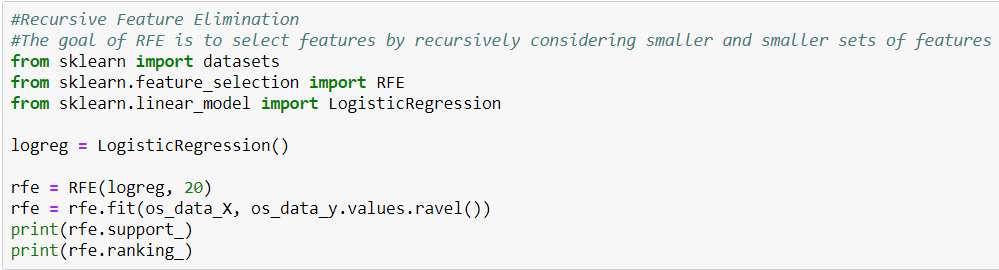
Proportion of rejected data in oversampled data is 0.5

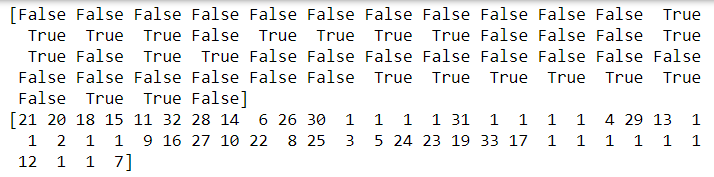
Proportion of approved data in oversampled data is 0.5

Now the data is perfect balanced! I over-sampled only on the training data, because by oversampling only on the training data, none of the information in the test data is being used to create synthetic observations, therefore, no information will bleed from test data into the model training.

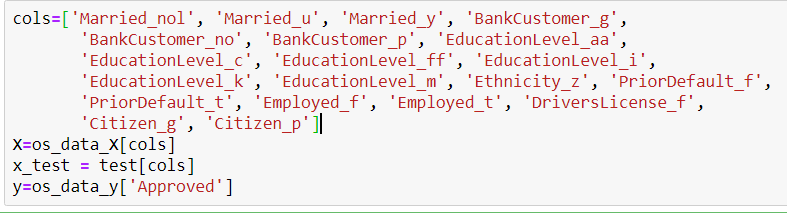
**Recursive Feature Elimination**

Recursive Feature Elimination (RFE) is based on the idea to repeatedly construct a model and choose either the best or worst performing feature, setting the feature aside and then repeating the process with the rest of the features. This process is applied until all features in the dataset are exhausted. The goal of RFE is to select features by recursively considering smaller and smaller sets of features.

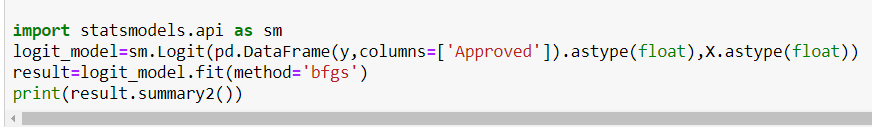


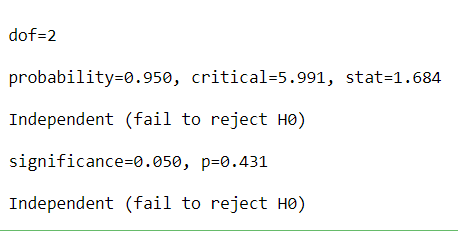


The RFE has helped us select the following features: 'Married\_nol', 'Married\_u', 'Married\_y', 'BankCustomer\_g', 'BankCustomer\_no', 'BankCustomer\_p', 'EducationLevel\_aa', 'EducationLevel\_c', 'EducationLevel\_ff', 'EducationLevel\_i', 'EducationLevel\_k', 'EducationLevel\_m', 'Ethnicity\_z', 'PriorDefault\_f', 'PriorDefault\_t', 'Employed\_f', 'Employed\_t', 'DriversLicense\_f', 'Citizen\_g', 'Citizen\_p'



**Implementing the model:**





The p-values for most of the variables are smaller than 0.05**, except 9 variables,** therefore, we will remove them.

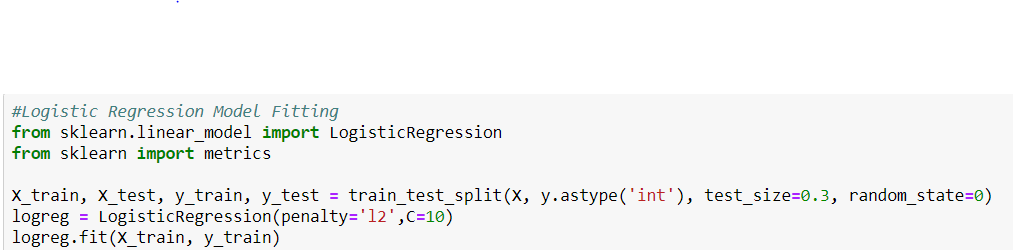
cols=['Married\_u', 'Married\_y', 'BankCustomer\_g',

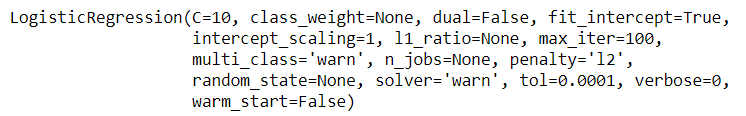
'BankCustomer\_p', 'EducationLevel\_ff', 'EducationLevel\_i',

'EducationLevel\_k', 'PriorDefault\_f',

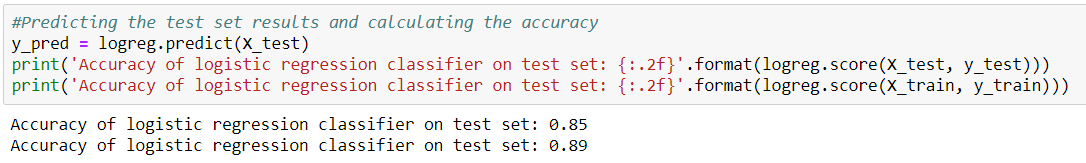
'PriorDefault\_t', 'Employed\_f', 'Employed\_t']

**Logistic Regression Model Fitting:**

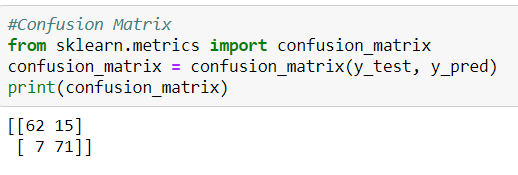




**Predicting the test set results and calculating the accuracy:**



**Confusion Matrix**



The result is refers that,  **62+71** correct predictions and **7+15** incorrect predictions.

**Compute precision, recall, F-measure and support:**

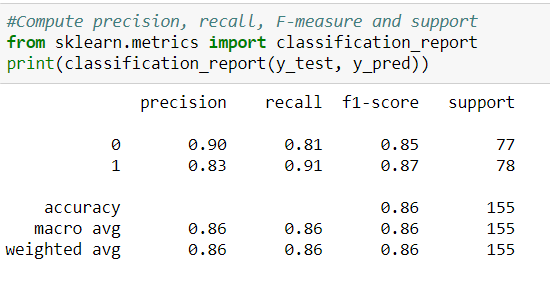
The precision is the ratio tp / (tp + fp) where tp is the number of true positives and fp the number of false positives. The precision is intuitively the ability of the classifier to not label a sample as positive if it is negative.

The recall is the ratio tp / (tp + fn) where tp is the number of true positives and fn the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.

The F-beta score can be interpreted as a weighted harmonic mean of the precision and recall, where an F-beta score reaches its best value at 1 and worst score at 0.

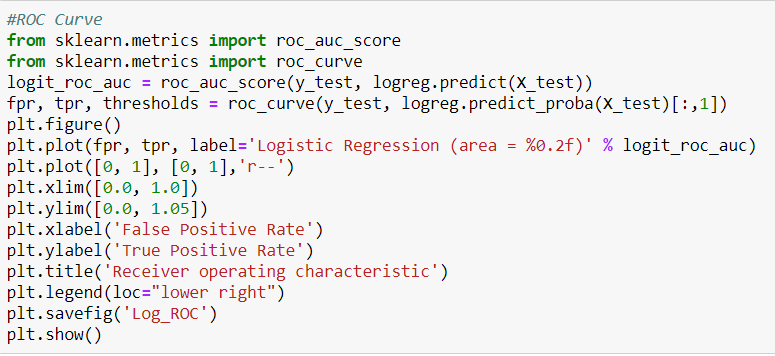
The F-beta score weights the recall more than the precision by a factor of beta. beta = 1.0 means recall and precision are equally important.

The support is the number of occurrences of each class in y\_test.



**Interpretation**: Of the entire test set, **86%** of the approval were the approval that the Bank liked. Of the entire test set, 86% of the Bank preferred Approval that were provided by the model.

**ROC Curve**





The dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner).