

Classification - Credit Score

Agenda

**Importing the Libraries Loading the Data**

**02**

**01**

**Data Cleaning One Hot Encoding**

**04**

**03**

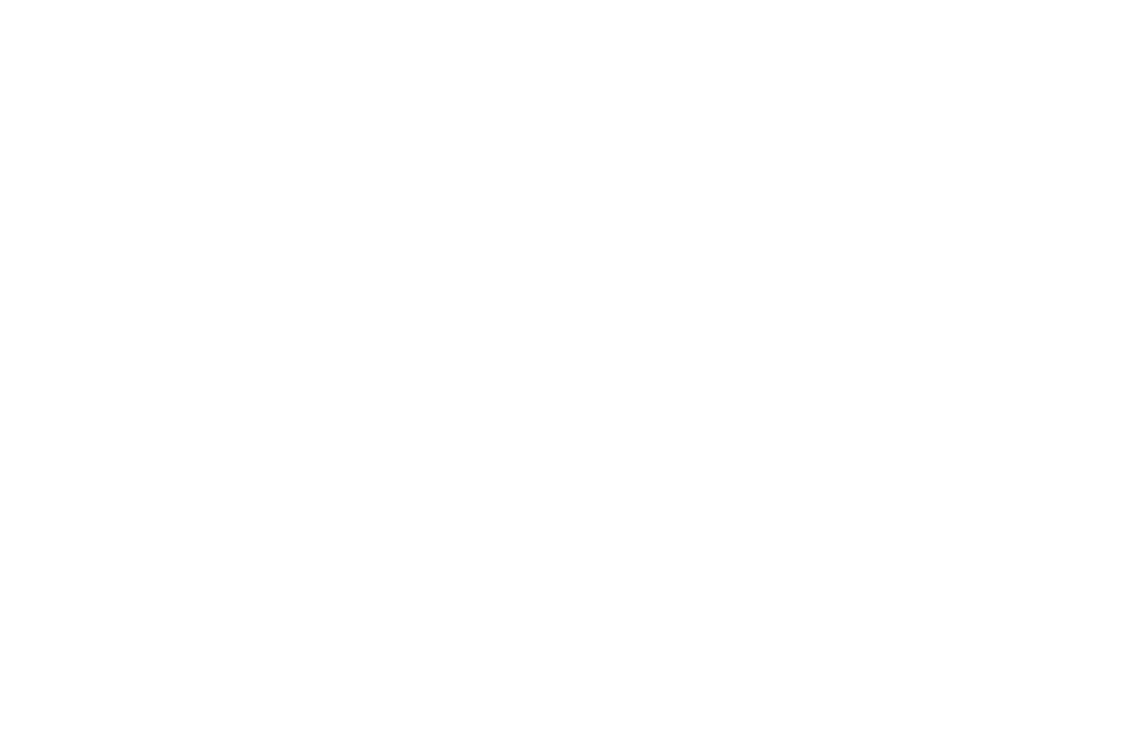
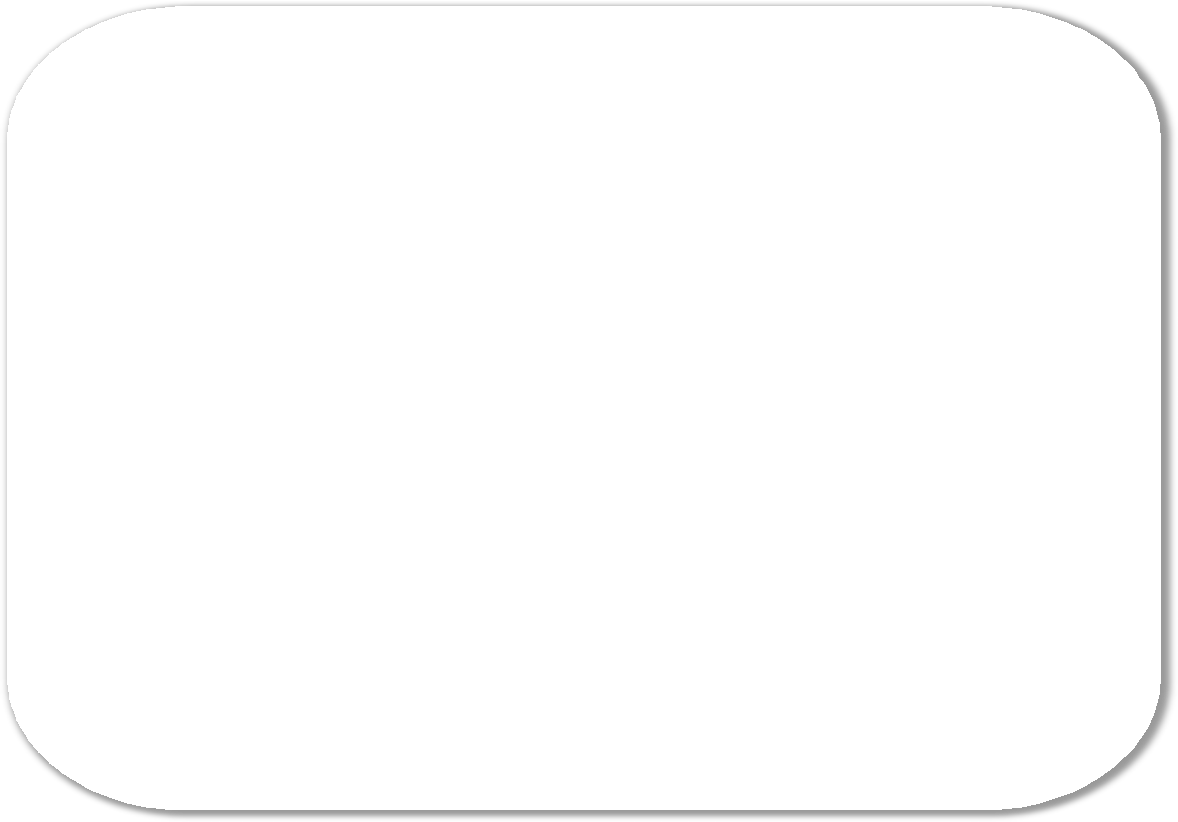
**Feature Selection Implementing ML Algorithms**

**06**

**05**

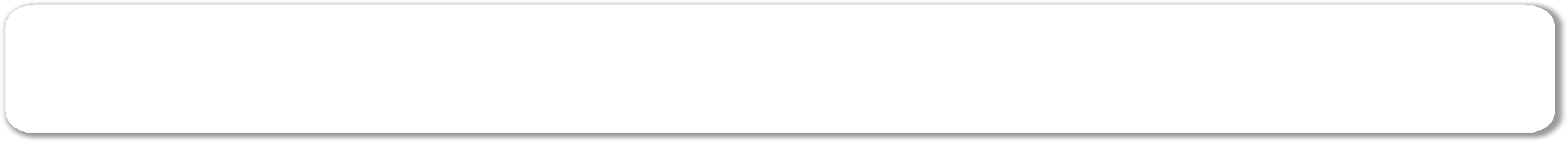
**Problem Statement**

I build an system to segregate the people into credit score brackets to reduce the manual efforts. Given a

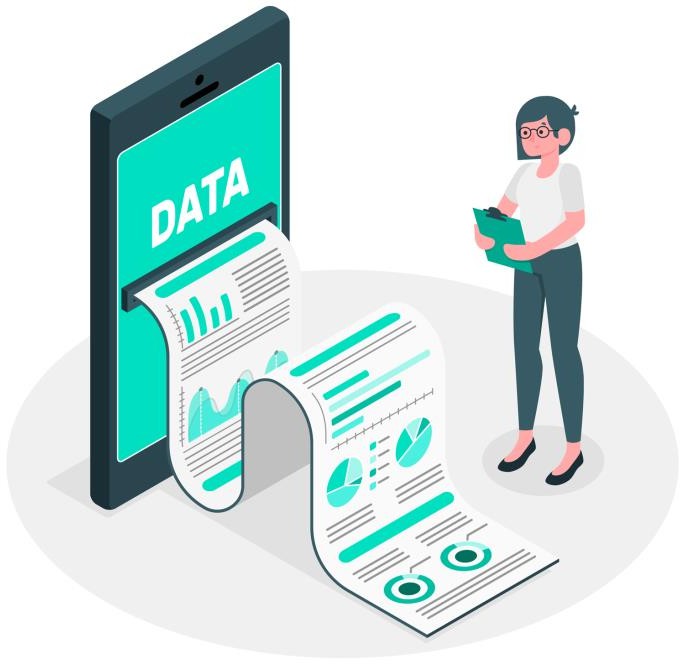


person’s credit-related information, build a machine learning model that can classify the credit score.

**Dataset Information**



Credit score dataset contains 1 lac records with 28 features.



**Dataset Information**

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| ID | unique identification of an entry |
| Customer\_ID | unique identification of a person |
| Month | month of the year |
| Name | name of a person |
| Age | age of the person |
| SSN | social security number of a person |
| Occupation | occupation of the person |
| Annual\_Income | annual income of the person |
| Monthly\_Inhand\_Salary | monthly base salary of a person |
| Num\_Bank\_Accounts | number of bank accounts a person holds |
| Num\_Credit\_Card | number of other credit cards held  by a person |

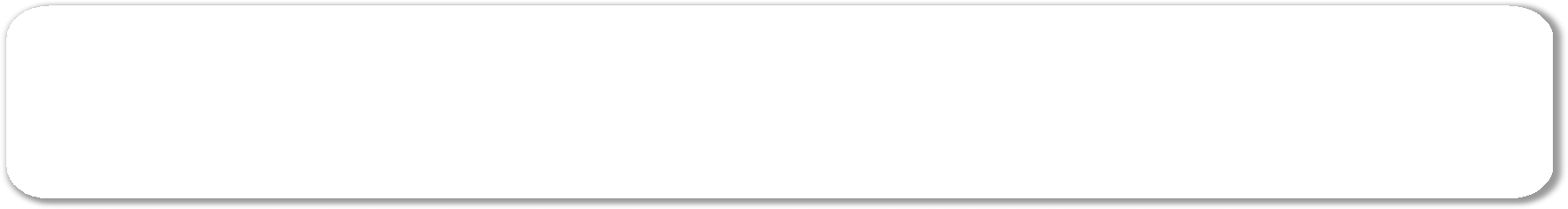
**Dataset Information**

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| Interest\_Rate | interest rate on credit card |
| Num\_of\_Loan | number of loans taken from the bank |
| Type\_of\_Loan | types of loan taken by a person |
| Delay\_from\_due\_date | average number of days delayed from the payment date |
| Num\_of\_Delayed\_Payment | age of the person |
| Changed\_Credit\_Limit | percentage change in credit card limit |
| Num\_Credit\_Inquiries | number of credit card inquiries |
| Credit\_Mix | classification of the mix of credits |
| Outstanding\_Debt | remaining debt to be paid (in USD) |
| Credit\_Utilization\_Ratio | utilization ratio of credit card |
| Credit\_History\_Age | the age of credit history of the person |

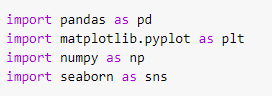
**Dataset Information**

|  |  |
| --- | --- |
| **Attributes** | **Description** |
| Payment\_of\_Min\_Amount | the minimum amount was paid by the person |
| Total\_EMI\_per\_month | monthly EMI payments (in USD) |
| Amount\_Invested\_monthly | monthly amount invested by the  customer (in USD) |
| Payment\_Behaviour | payment behavior of the customer (in USD) |
| Monthly\_Balance | monthly balance amount of the customer  (in USD) |
| Credit\_Score | the bracket of credit score |

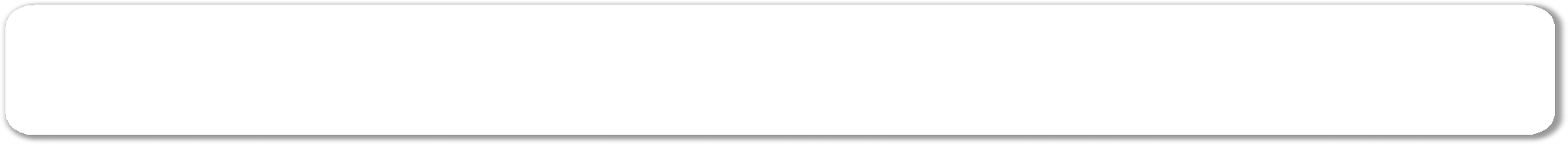
**Importing the Libraries**



We start off this project by importing all the necessary libraries that will be required for the process.

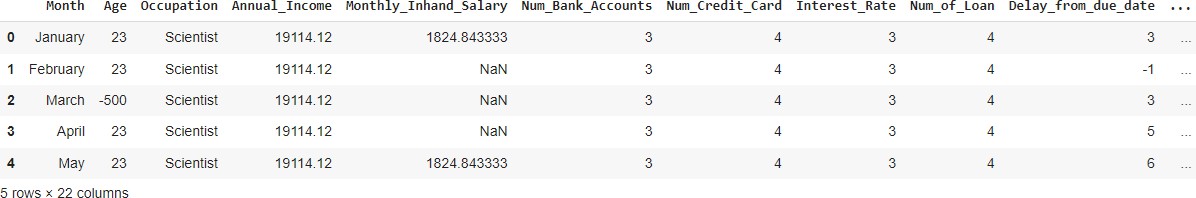


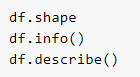
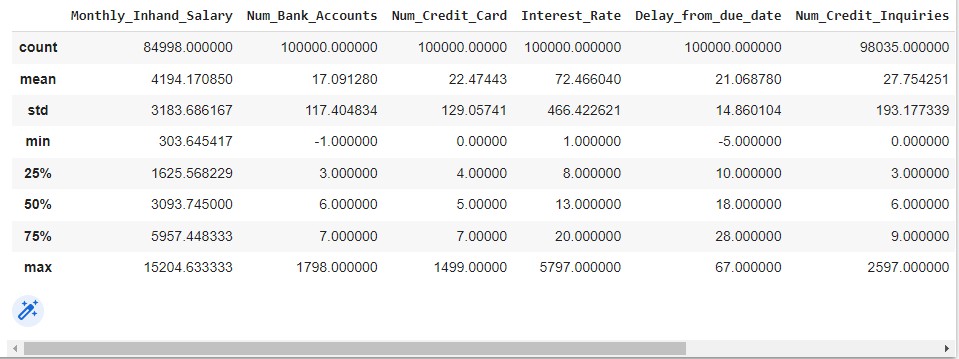
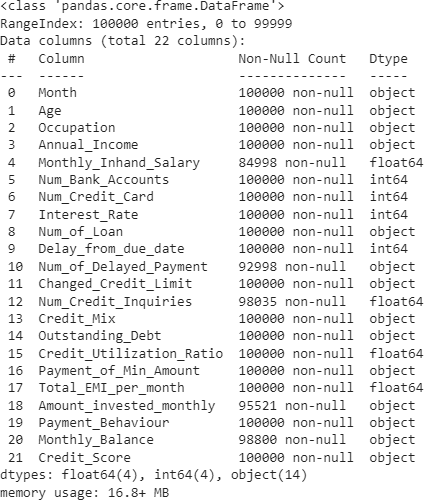
**Loading the Data**

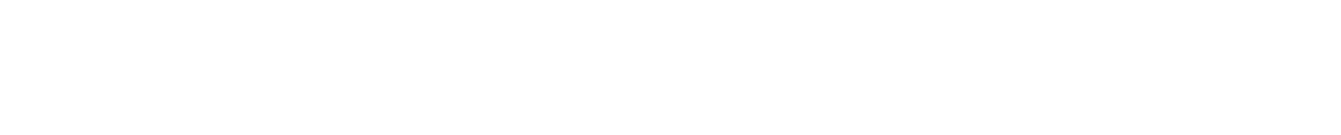
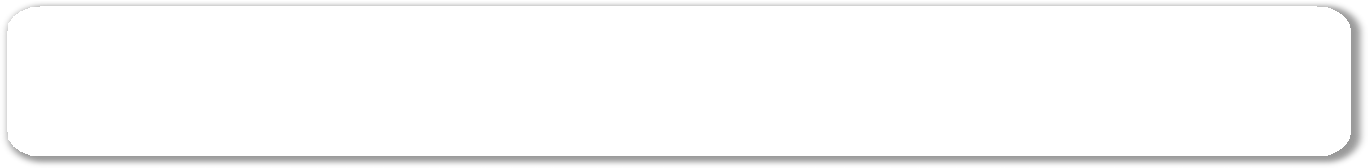


Loading the data and removing unnecessary column from the dataframe

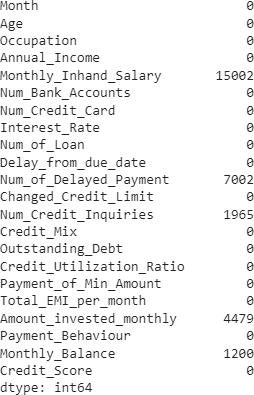


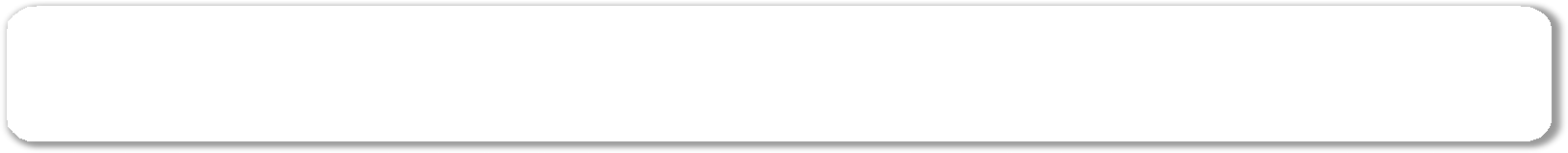


**Loading the Data**



Checking the shape of a dataframe and datatypes of all columns along with calculating the statistical data.

**Missing Values**

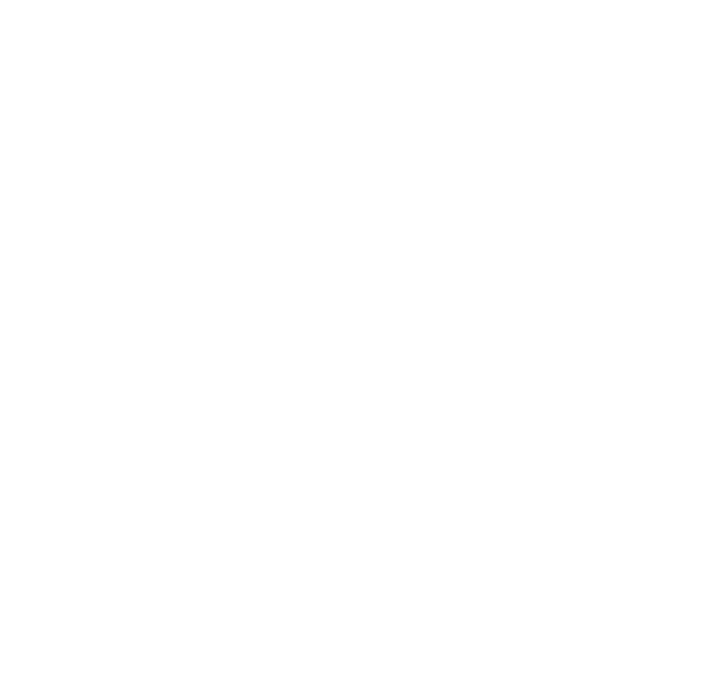
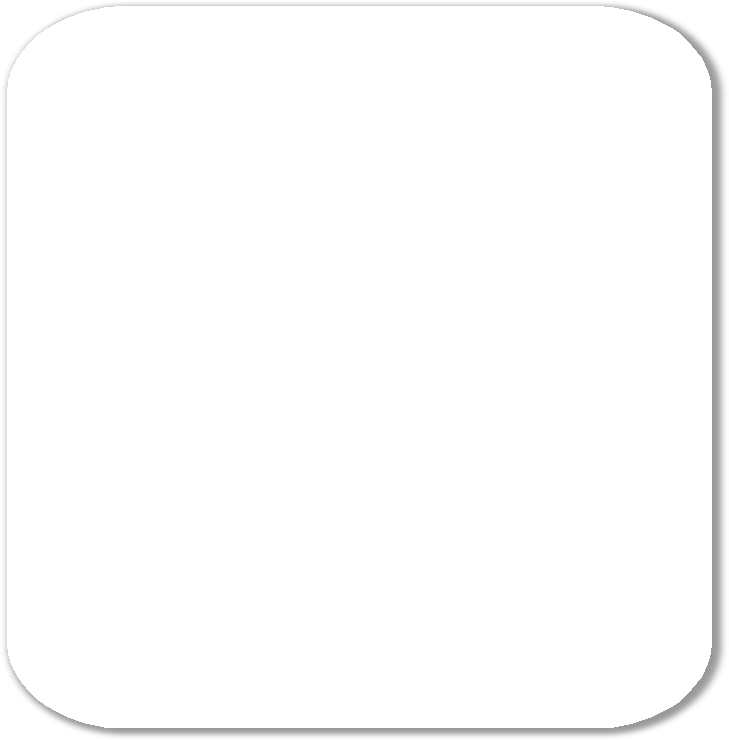


Checking out the missing values in a dataframe



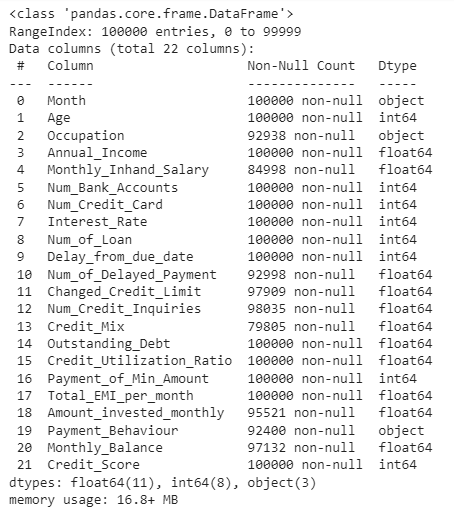
**Data Cleaning**

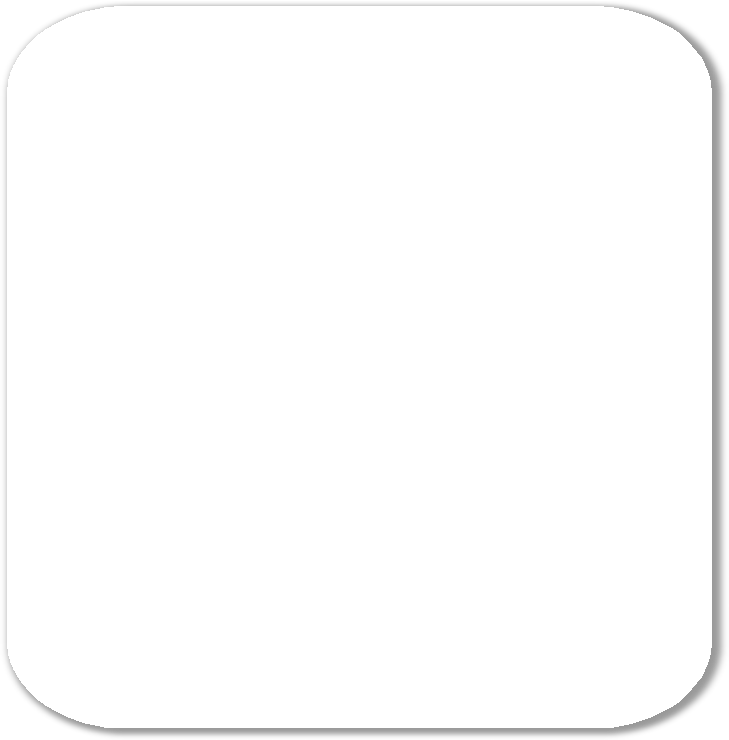




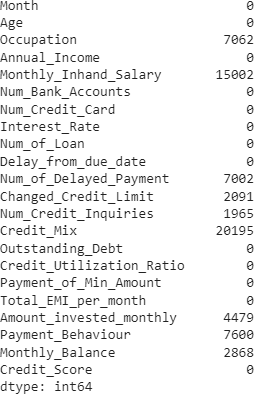
Replacing the special characters with empty string or with null values according to the data and converting it into int or float datatype. Also, Converting the categorical values of some columns into integer values.

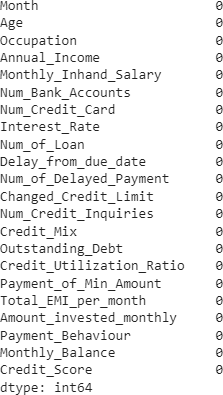
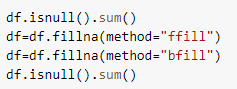
**Data Cleaning**

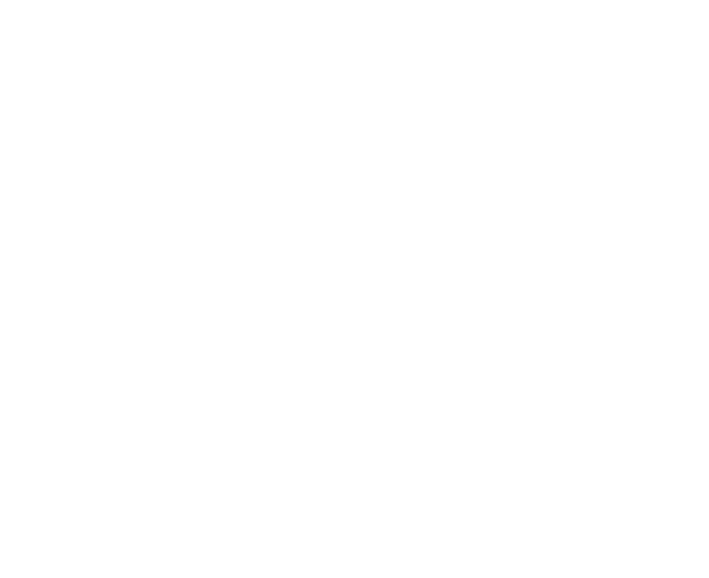
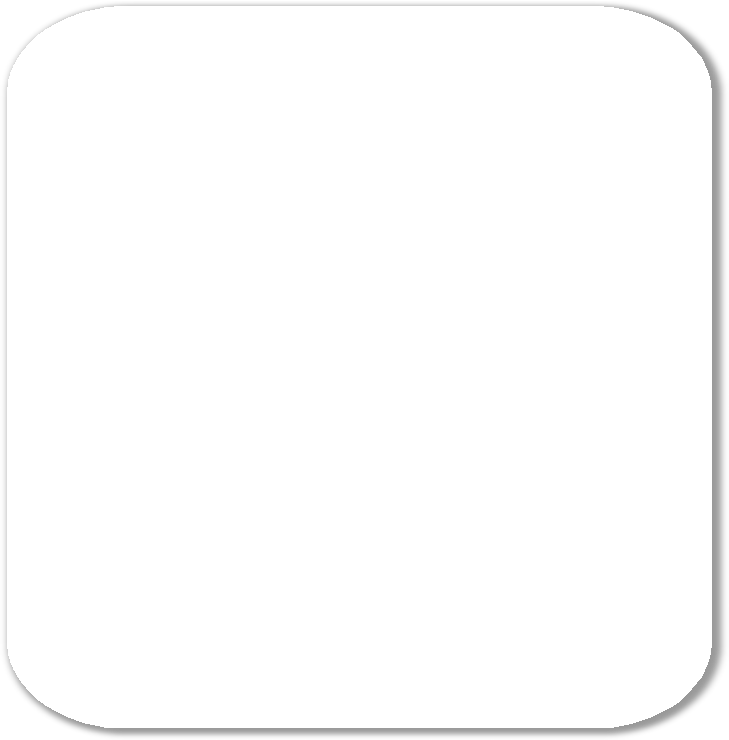




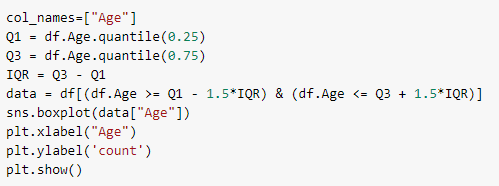
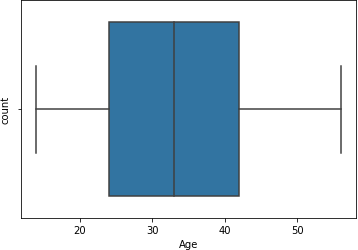
Clearly, The datatype of the columns have been changed after performing the operation

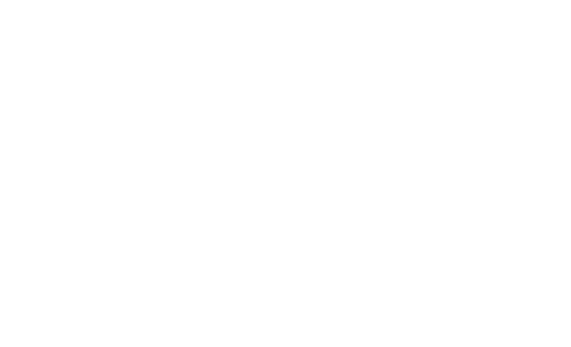
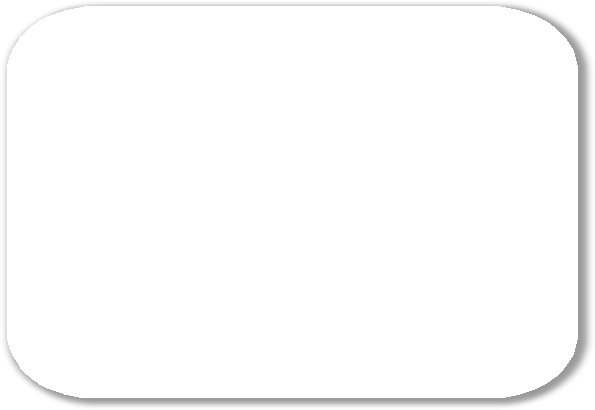
**Data Cleaning**



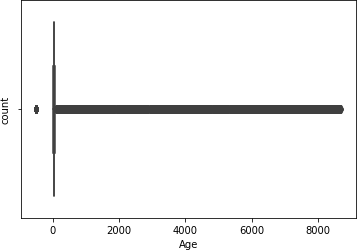
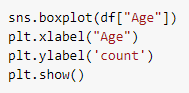


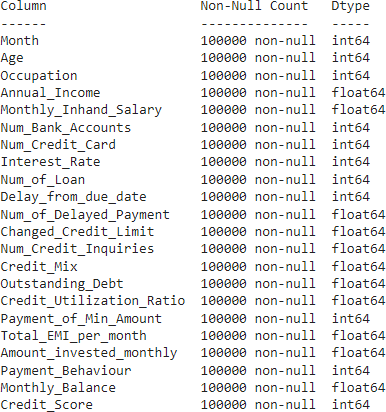
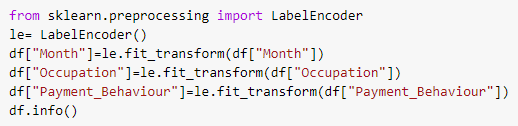
After replacing the special characters with null value. The new missing value is shown in the figure. Here Forward and backward filling method is used to fill the missing values.

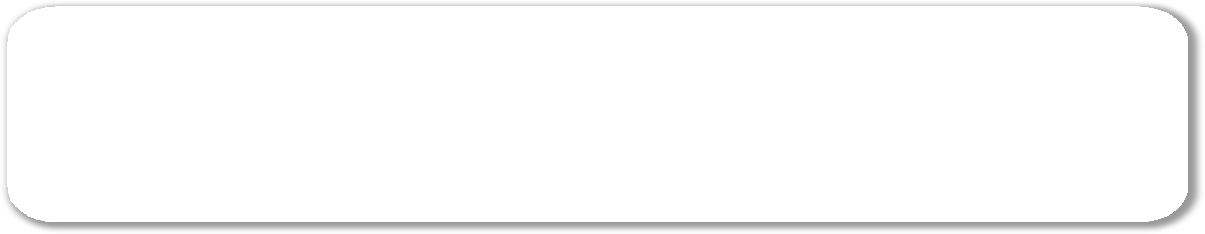
**Data Cleaning**



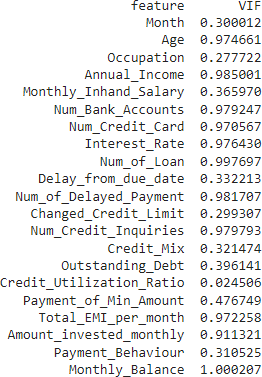
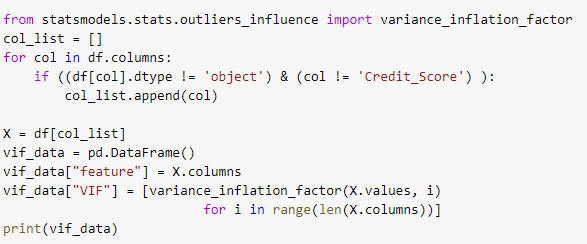
removing outliers from age since all other columns values are relevant

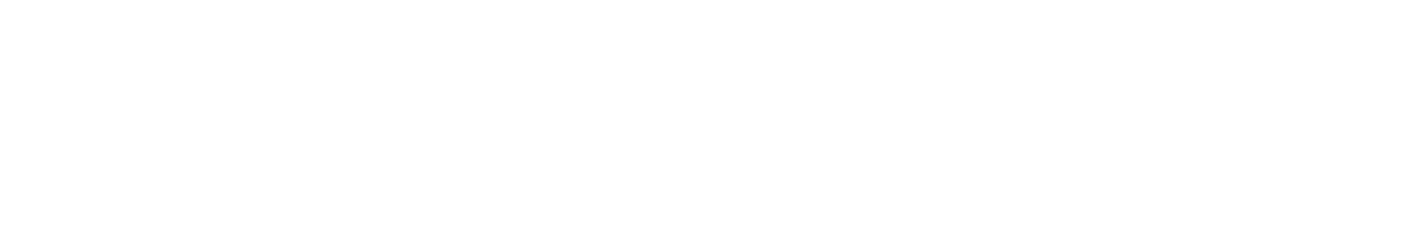
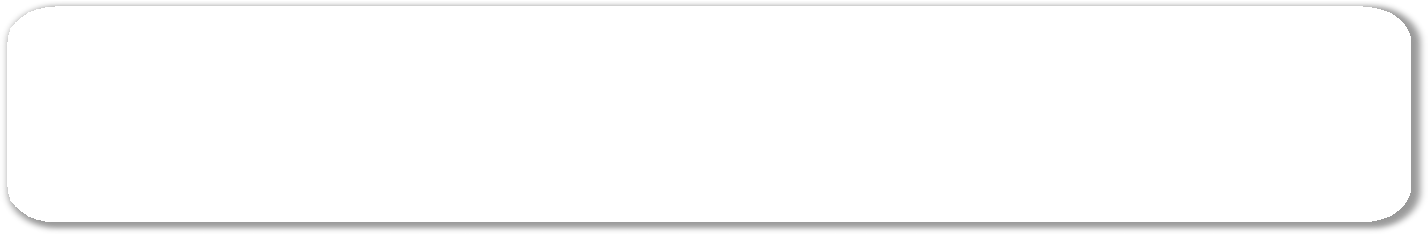


**Data Cleaning**

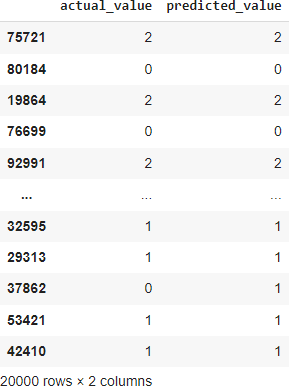
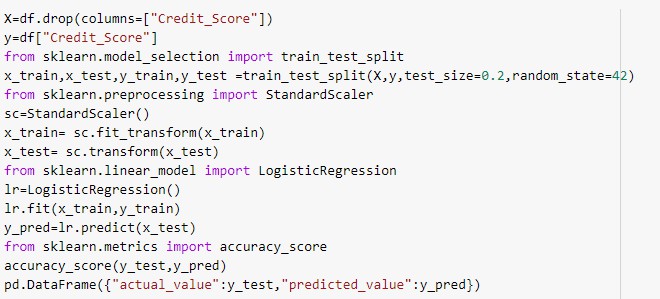


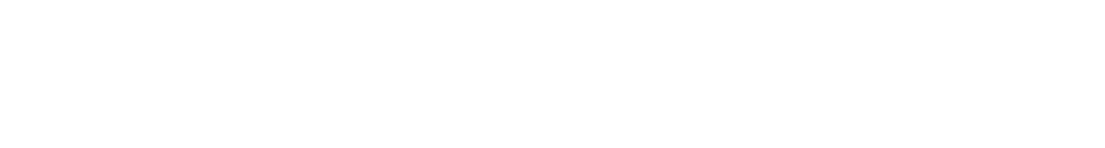
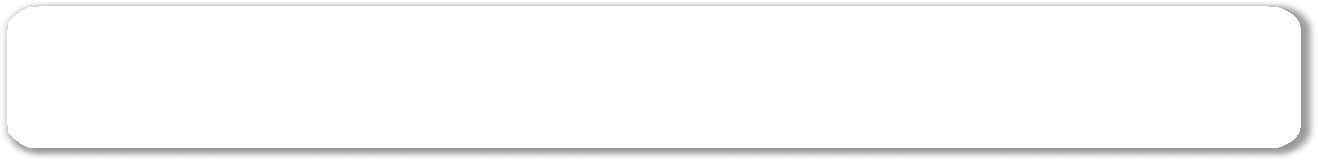
Performing One Hot Encoding for categorical features of a dataframe

**Feature Selection**



Selecting the features using VIF. VIF should be less than 5. Here, all features have VIF value less than 5, So we will select all the features.

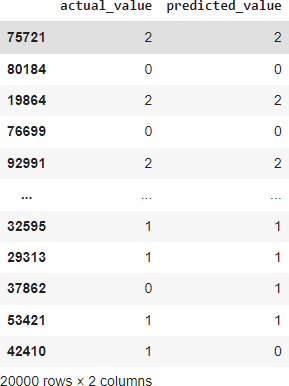
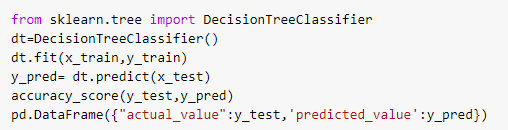
**Logistic Regression**

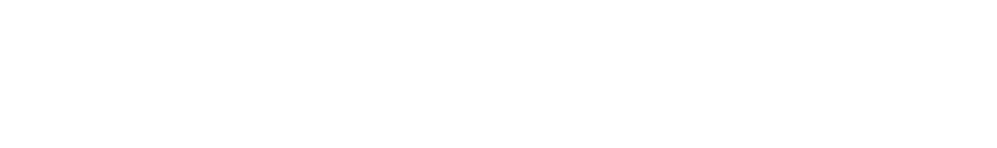
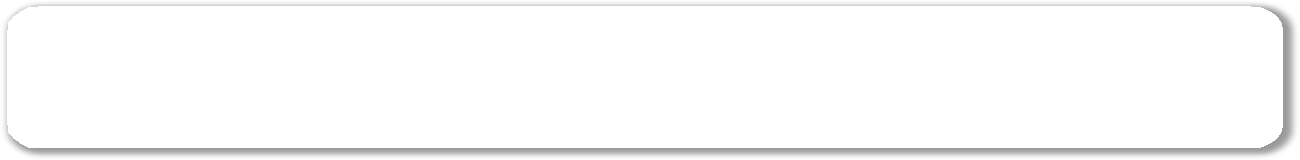


The accuracy of the logistic regression model is

61.8 percentage



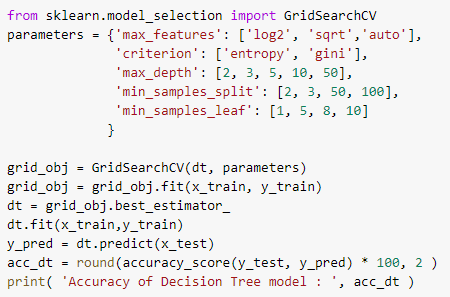
**Decision Tree**

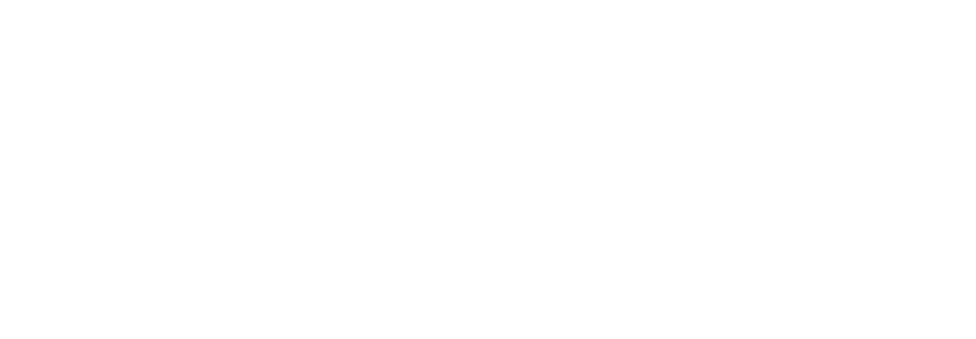
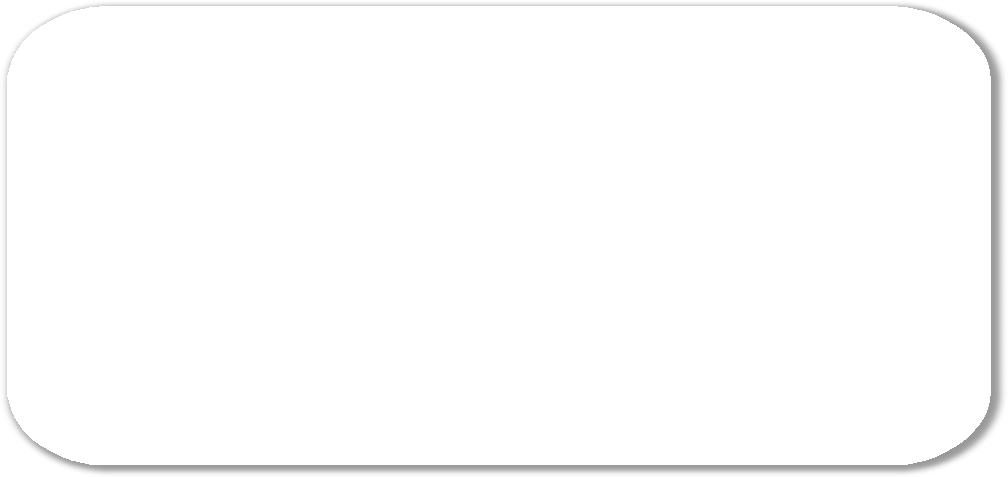


The accuracy of the decision tree model is

69.7 percentage

**Hyperparameter Tuning on Decision Tree**



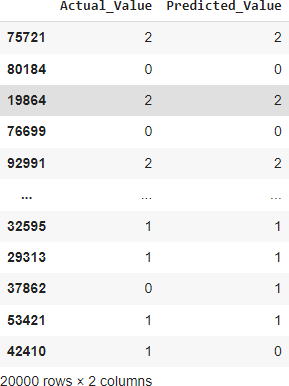
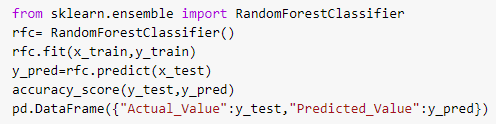


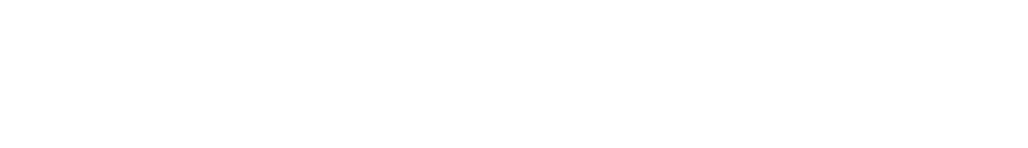
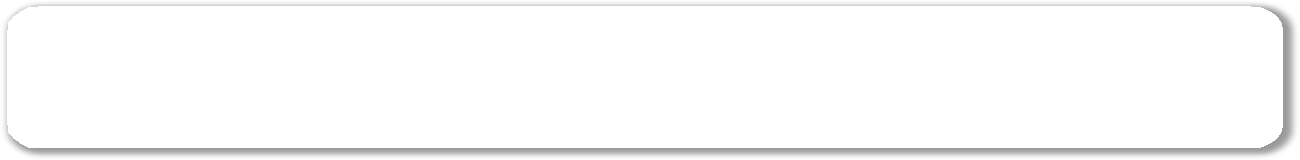
Here, We are using

GridSearch CV technique which is used to identify the optimal hyperparameters for a model and the

accuracy obtained from Decision Tree is 70.93



**Random Forest**



The accuracy of the random forest model is

79.7 percentage