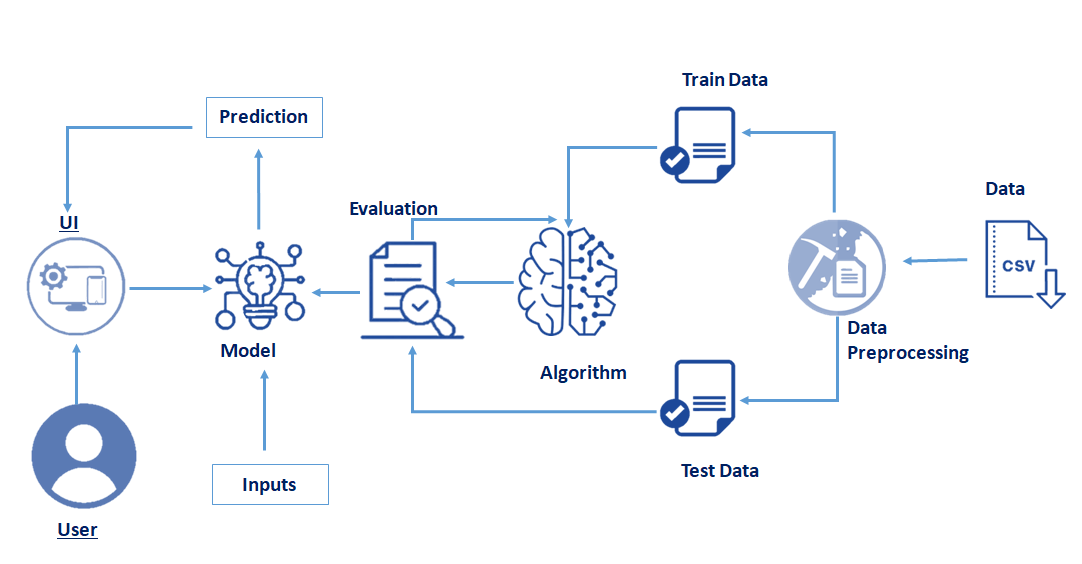
**Credit Card Approval Prediction**

**Project Description:**

## With the increasing number of credit card applications, banks are opting towards the use of prediction-based algorithms as opposed to manual approval methods. Data analysis has exhibited a strong correlation between several financial and personal factors of a client and the likelihood of said client complying with their respective bank's credit policies. In this paper, we propose the use of the Machine Learning algorithm to predict and grant credit cards to applicants based on the customers' activity history. We used some financial and personal factors. We predicted the resulting factors through the use of Machine Learning algorithm with an emphasis on error minimization. Using this Machine Learning model, the machine-learned which of these applicants are most likely to accumulate bad debts and granted or rejected the applications based on the prediction.

.

**Technical Architecture:**

****

**Pre requisites:**

**To complete this project, you must require following software’s , concepts and packages**

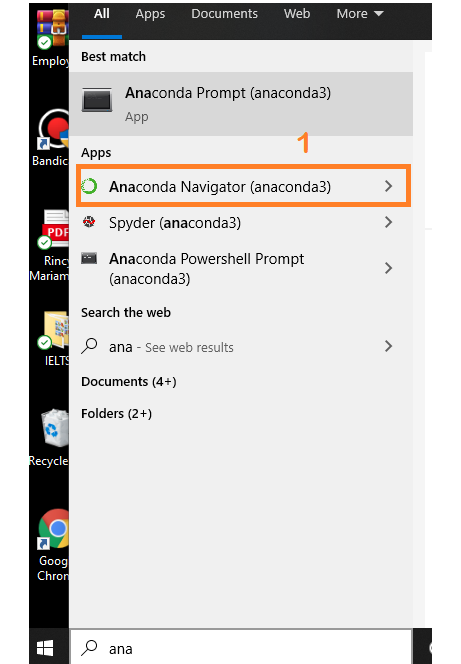
* **Anaconda navigator:**
  + Refer to the link below to download anaconda navigator
  + **Link : https://www.youtube.com/watch?v=5mDYijMfSzs**
* **Python packages:**

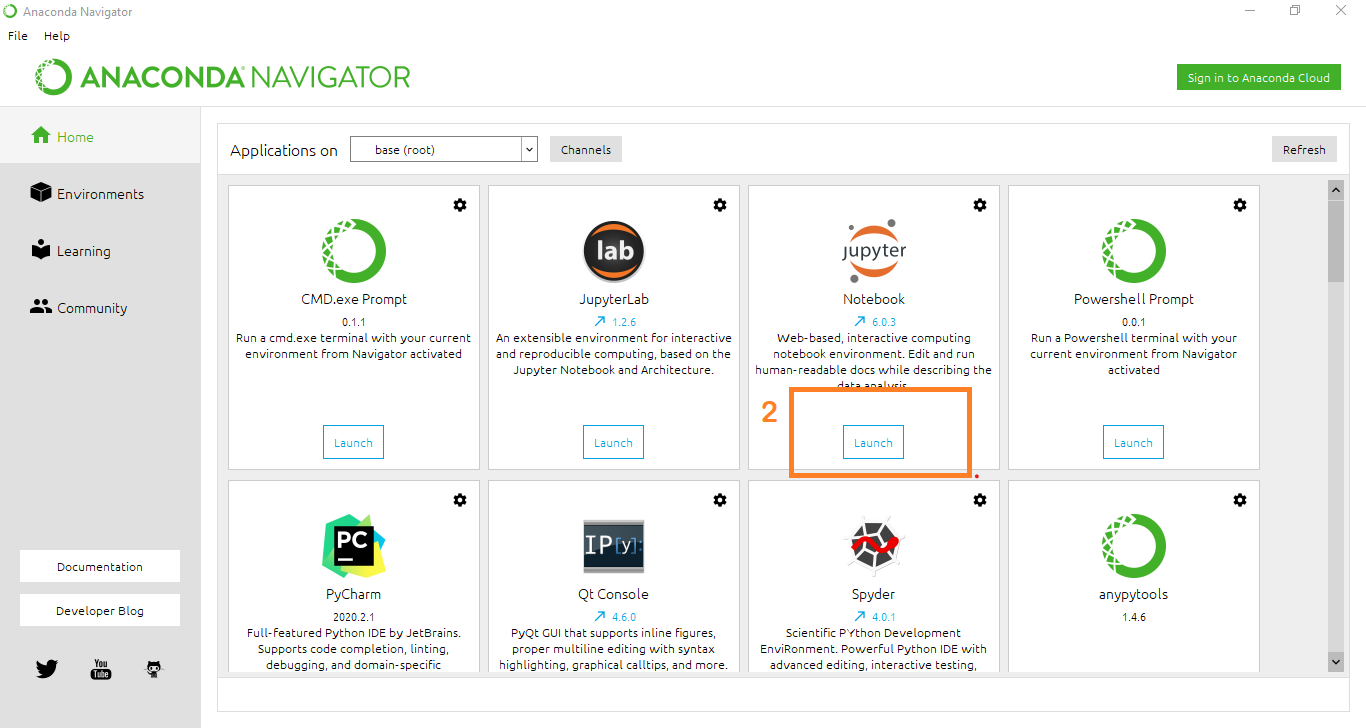
Open anaconda prompt as administrator.

* Type “pip install numpy” and click enter.
* Type “pip install pandas” and click enter.
* Type “pip install matplotlib” and click enter.
* Type “pip install scikit-learn” and click enter.
* Type “pip install Flask” and click enter.

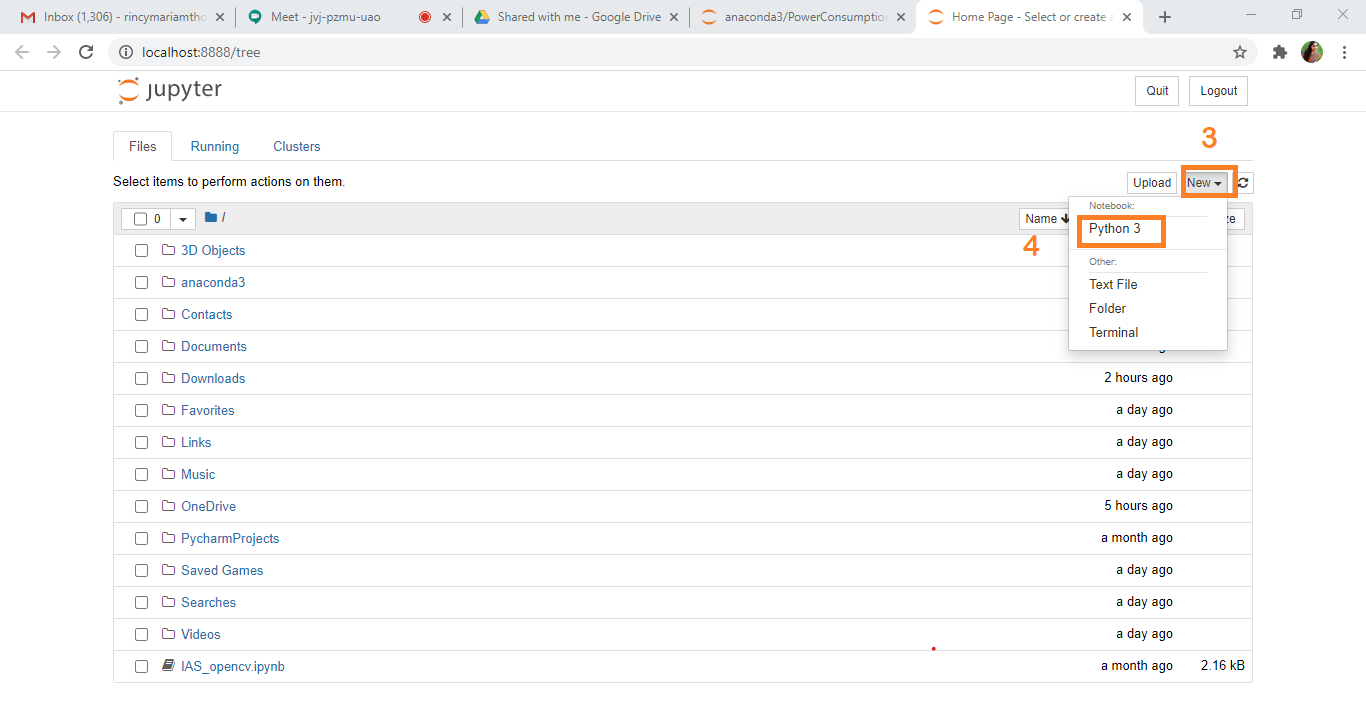
The above steps allow you to install the packages in the anaconda environment

* **Launch Jupyter**
  + Search for Anaconda Navigator and open Launch Jupyter notebook.





* Then you will be able to see that the jupyter notebook runs on local host:8888.
* To Create a new file Go to New 🡪Python3.The file in jupyter notebook is saved with .ipynb extension.



* + Flask Basics : <https://www.youtube.com/watch?v=lj4I_CvBnt0>

**Project Objectives:**

By the end of this project:

* To get a basic introduction of our project & what’s the business problem associated with it?
* We’ll start by loading and viewing the dataset.
* To manipulate data, if there are any missing entries in the dataset.
* To perform exploratory data analysis (EDA) on our dataset.
* To pre-process data before applying machine learning model to the dataset.
* To apply machine learning model that can predict if an individual’s application for a credit card will be accepted or not.

**Project Flow:**

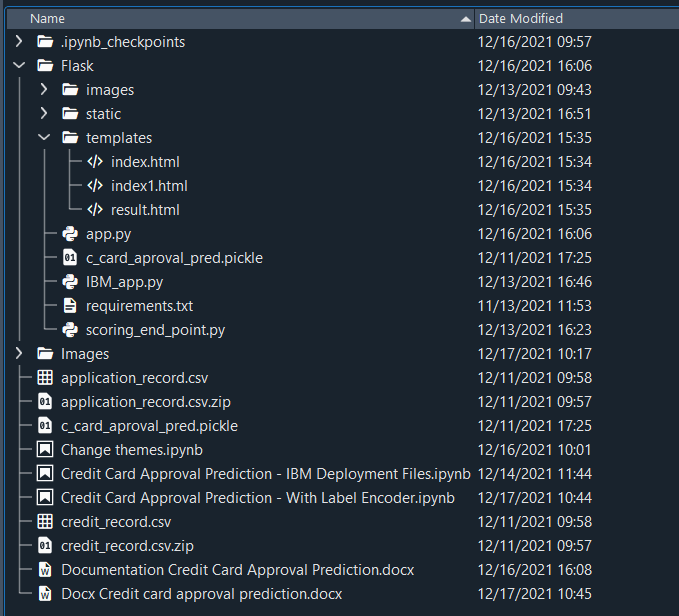
* User interacts with the UI (User Interface) to enter the input values
* Entered input values are analyzed by the model which is integrated
* Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities and tasks listed below

* Data Collection.
* Collect the dataset or Create the dataset
* Data Pre-processing.
  + Import the Libraries.
  + Importing the dataset.
  + Data Visualization.
  + Data Cleaning
  + Feature Engineering
  + Splitting Data into Train and Test.
* Model Building
  + Import the model building Libraries
  + Initializing the model
  + Training and testing the model
  + Evaluation of Model
  + Save the Model
* Application Building
  + Create an HTML file
  + Build a Python Code

**Project Structure:**

Create a Project folder which contains files as shown below

****

* A python file called app.py for server side scipting.
* We need the model which is saved and the saved model in this content is **c\_card\_aproval\_pred.pkl**
* Templates folder which contains index.HTML file, index1.HTML file, result.HTML file.

**Milestone 1: Data Collection:**

ML depends heavily on data, without data, it is impossible for an “AI” to learn. It is the most crucial aspect that makes algorithm training possible. In Machine Learning projects, we need a training **data set.**It is the actual **data set** used to train the model for performing various actions.

**Activity 1: Download The dataset**

You can collect datasets from different open sources like kaggle.com, data.gov, UCI machine learning repository etc.

Please refer to the link given below to download the data set and to know about the dataset

<https://www.kaggle.com/namphuengauawatcharo/credit-card-approval-prediction/data>

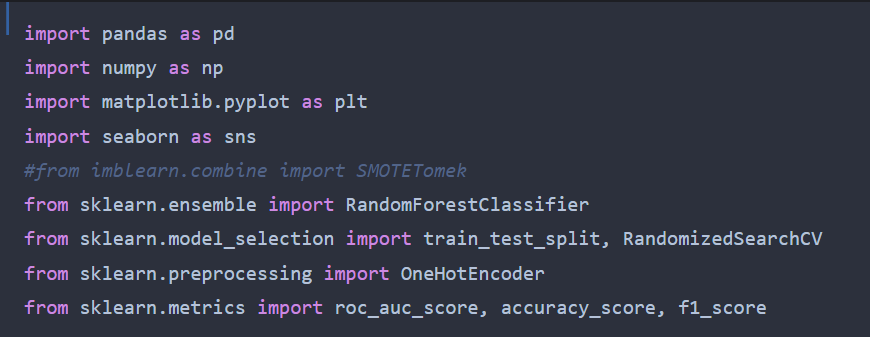
**Milestone 2: Data Preprocessing**

Data Pre-processing includes the following main tasks

* + Import the Libraries.
  + Importing the dataset.
  + Data Visualization.
  + Data Cleaning
  + Feature Engineering
  + Splitting Data into Train and Test.

**Activity 1: Import Necessary Libraries**

* + It is important to import all the necessary libraries such as pandas, numpy, matplotlib.
  + **Numpy**- It is an open-source numerical Python library. It contains a multi-dimensional array and matrix data structures. It can be used to perform mathematical operations on arrays such as trigonometric, statistical, and algebraic routines.
  + **Pandas**- It is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.
  + **Seaborn**- Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
  + **Matplotlib**- Visualisation with python. It is a comprehensive library for creating static,animated, and interactive visualizations in Python
  + **Sklearn** – which contains all the modules required for model building

****

**Activity 2: Importing the Dataset**

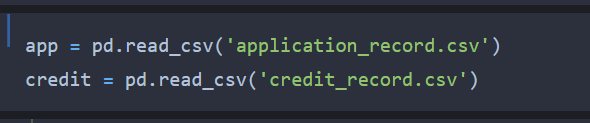
* You might have your data in .csv files, .excel files
* Let’s load a .csv data file into pandas using **read\_csv() function.**We will need to locate the directory of the CSV file at first (it’s more efficient to keep the dataset in the same directory as your program).
* If your dataset is in some other location ,Then

**Data=pd.read\_csv(r”File\_location/datasetname.csv”)**

**Note:**r stands for "raw" and will cause backslashes in the string to be interpreted as actual backslashes rather than special characters.

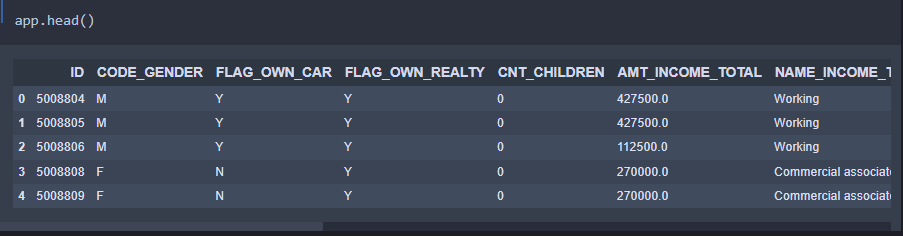
* If the dataset in same directory of your program, you can directly read it, without giving raw as r.
* We have two data sets 1) application\_record.csv and 2) credit\_record.csv contains following Columns
* Data set 1) ID, CODE\_GENDER, FLAG\_OWN\_CAR, FLAG\_OWN\_REALTY, CNT\_CHILDREN,
* AMT\_INCOME\_TOTAL, NAME\_INCOME\_TYPE,NAME\_EDUCATION\_TYPE,
* NAME\_FAMILY\_STATUS, NAME\_HOUSING\_TYPE, DAYS\_BIRTH,
* DAYS\_EMPLOYED, FLAG\_MOBIL, FLAG\_WORK\_PHONE, FLAG\_PHONE,
* FLAG\_EMAIL, OCCUPATION\_TYPE, CNT\_FAM\_MEMBERS
* Dataset 2) ID, MONTHS\_BALANCE, STATUS – Target Column

The target column to be predicted is **Weather a person is eligible for credit card or not**. Based on the input variables. The predicted output gives them a fair idea about he/she is eligible for credit card or not.

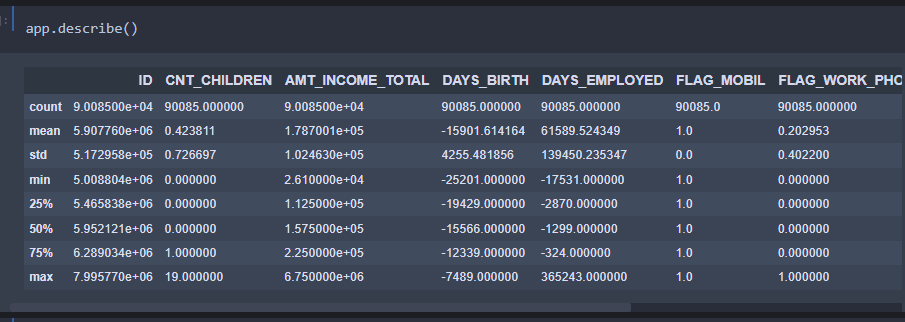


**Activity 3: Analyse the data**

* head() method is used to return top n (5 by default) rows of a DataFrame or series.

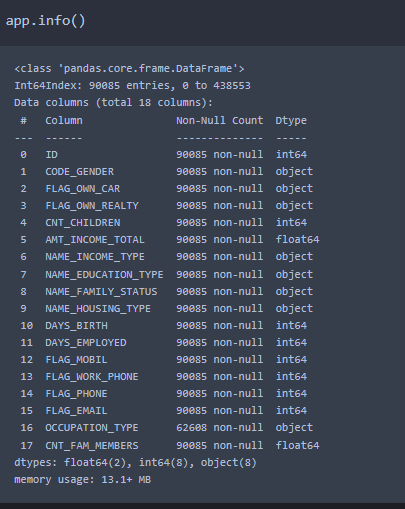


* describe() method computes a summary of statistics like count, mean, standard deviation, min, max and quartile values.



From the data we infer that there are only decimal values and no categorical values

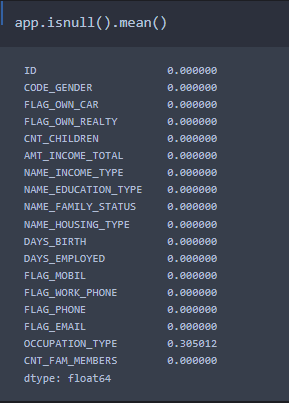
* info() gives information about the data



**Activity 4: Handling Missing Values**

1. After loading it is important to check the complete information of data as it can indication many of the hidden information such as null values in a column or a row

2.Check whether any null values are there or not.



**In our case we don’t have missing values**

**Activity 5: Data Visualisation**

* Data visualization is where a given data set is presented in a graphical format. It helps the detection of patterns, trends and correlations that might go undetected in text-based data.
* Understanding your data and the relationship present within it is just as important as any algorithm used to train your machine learning model. In fact, even the most sophisticated machine learning models will perform poorly on data that wasn’t visualized and understood properly.
* To visualize the dataset we need libraries called Matplotlib and Seaborn.
* The Matplotlib library is a Python 2D plotting library which allows you to generate plots, scatter plots, histograms, bar charts etc.

Let’s visualize our data using Matplotlib and searborn library.

Matplotlib :- Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

Some of the major Pros of Matplotlib are:

* Generally easy to get started for simple plots
* Support for custom labels and texts
* Great control of every element in figure
* High-quality output in many formats
* Very customizable in general
* **Plt.bar():** Make a bar plot.
* **Plt.plot():** Can be used to make a line graph.
* **Plt.subplots():** Add a subplot to the current figure.It returns two arguments fig and ax where fig is figure and ax can be either a single Axes object or an array of Axes objects.
* **plt.xticks():** A list of positions at which ticks should be placed.
* **plt.xlabel(label\_name):** Set the label for the x-axis.
* **plt.ylabel(label\_name):** Set the label for the y-axis.
* **plt.shows():** Display a figure.
* **plt.grid():** Configure the grid lines.

**Seaborn :-** Seaborn is an **open-source Python library built on top of matplotlib**. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library. The graphs created can also be customized easily.

# 1.Numerical Data Ploting

* replot()
* scatterplot()
* lineplot()

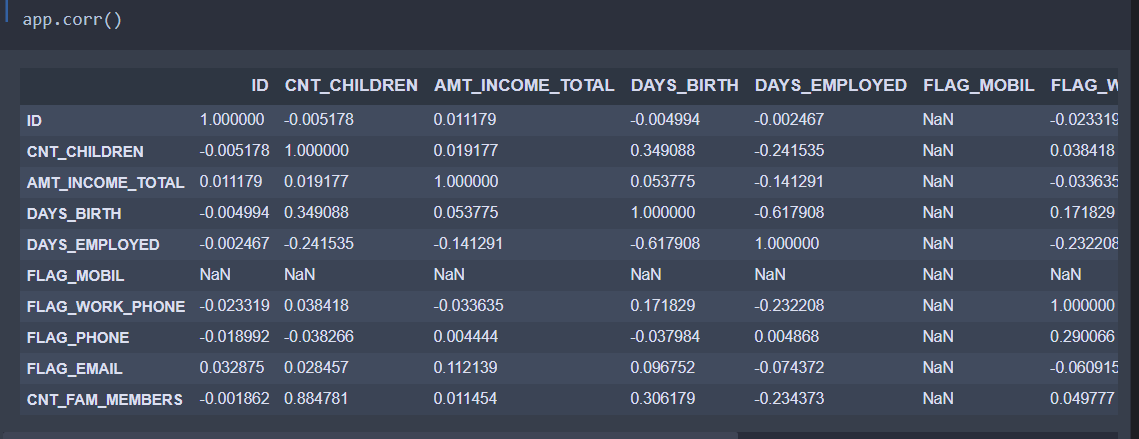
# 2. Categorical Data Ploting

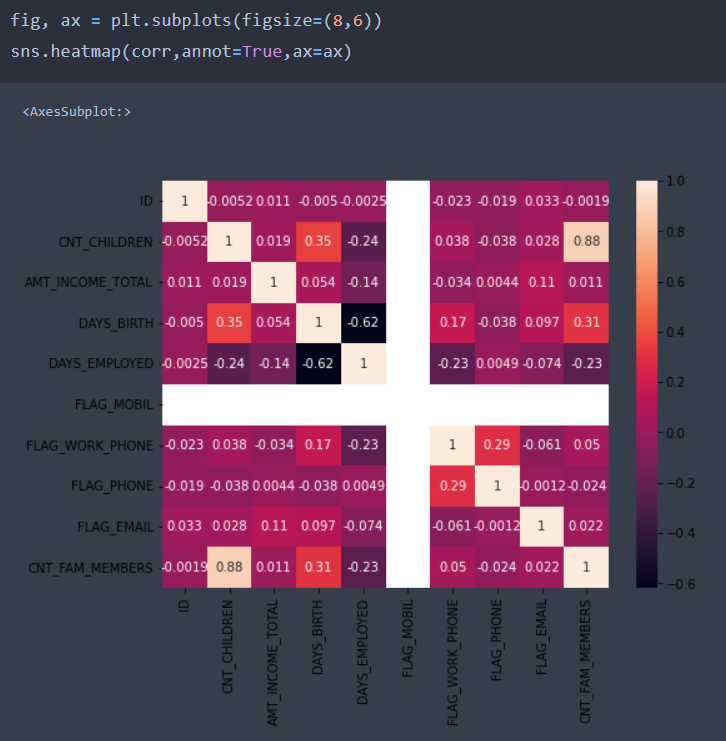
* barplot()
* countplot()
* boxplot()
* catplot()

# 3.Visualization Distribution of the Data

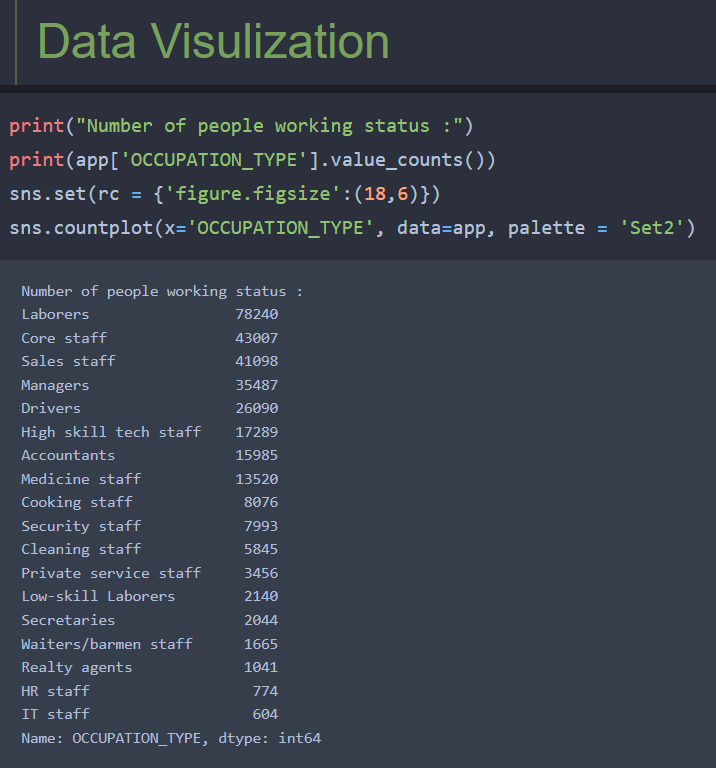
* distplot()
* jointplot()

**app.corr()** gives the correlation between the columns

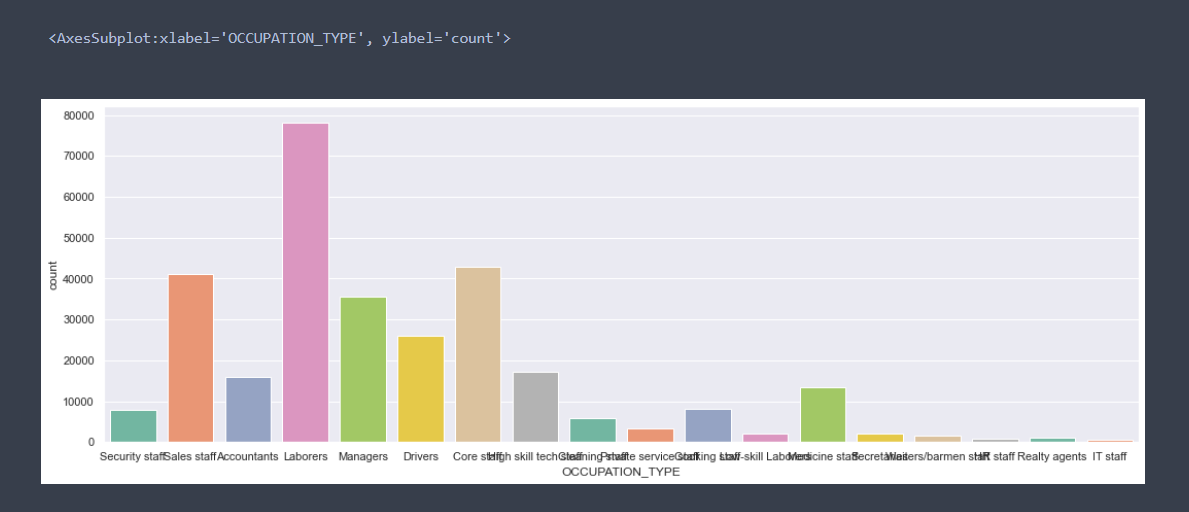




* Correlation strength varies based on colour, lighter the colour between two variables, more the strength between the variables, darker the colour displays the weaker correlation
* We can see the correlation scale values on left side of the above image



First we are checking how many different-different types of occupation present in OCCUPATION\_TYPE column then we are going to visualize.



Similarly, in NAME\_HOUSING\_TYPE, FLAG\_OWN\_CAR and NAME\_INCOME\_TYPE

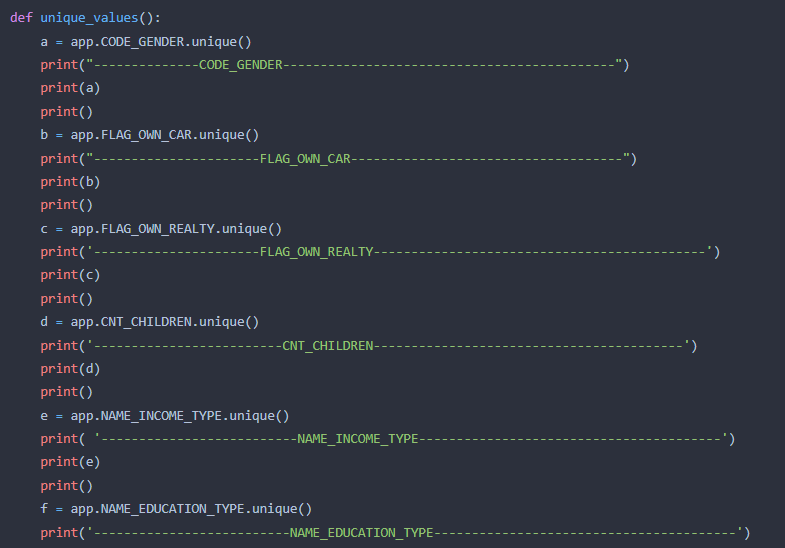


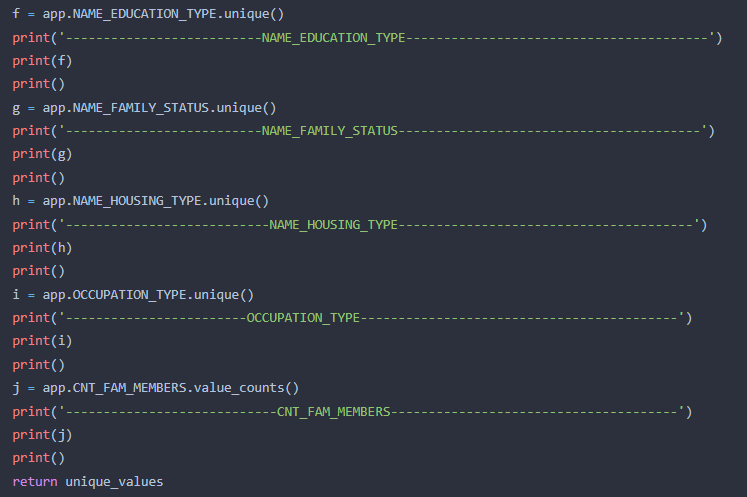




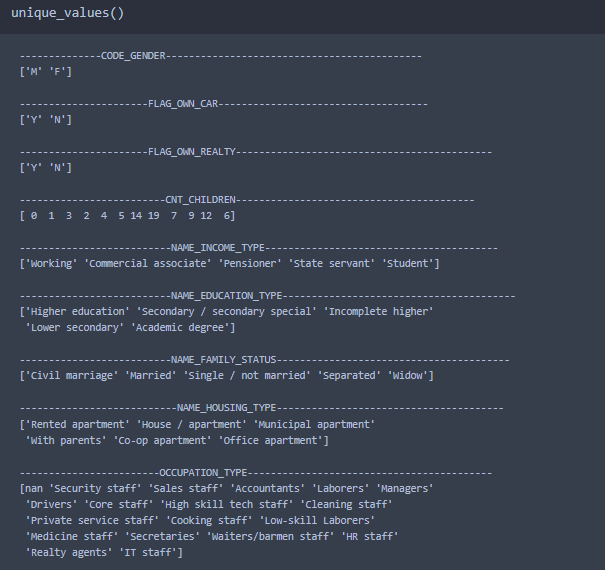
**Activity 6:** **Data Cleanning**

Checking the unique values on categorical column:-

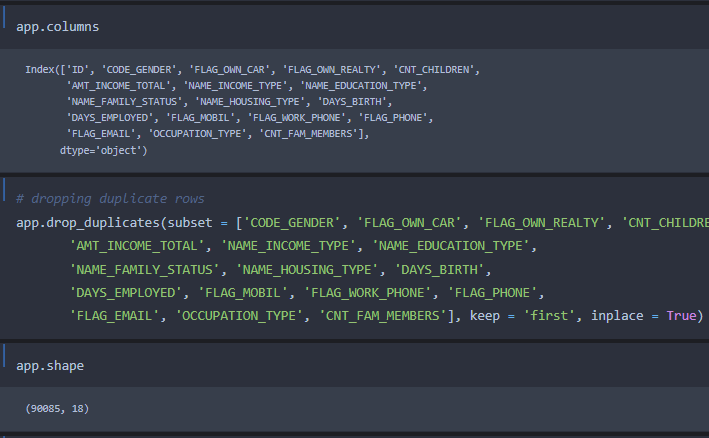




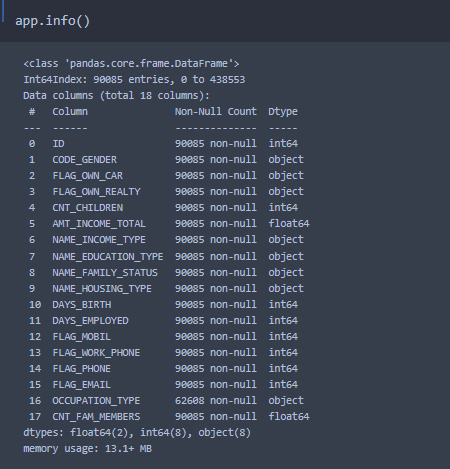
Output:-



Removing Unwanted Columns :-

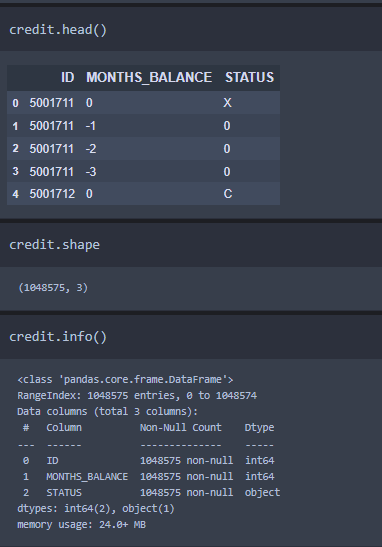


After removing the unwanted column we are checking the data type, columns name, missing values etc. And we are able to check by using only one function **app.info()**



**Now we are moving to our second data frame(credit):-**

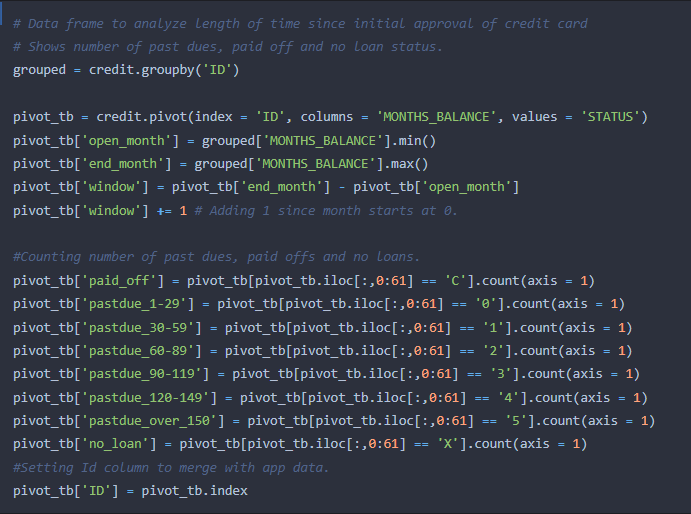
Similarly, we are checking top 5 row by using **credit.head()** and short information about the data set by using **credit.info()** function.



We are grouping the ID column and saving it as a variable grouped then we are creating a pivot table to expand all these value for understanding very clearly.

We are using as a index ID and and for column we are using MONTHS\_BALANCE and STATUS as a value.

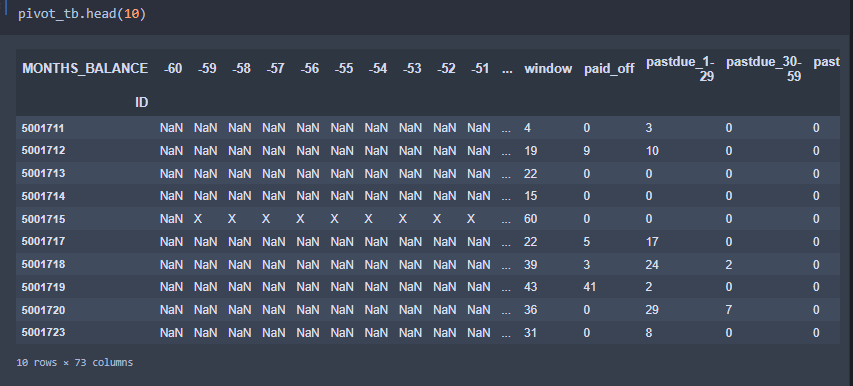
* Minimum MONTHS\_BALANCE as a open\_month
* Maximum MONTHS\_BALANCE as a end\_months
* And for window we are substracting end\_months – open\_months
* In window we are adding 1 since month start at o



Counting number of past due, paid offs and no loans. And assigning the values where:-

* Paid\_off means loan paid on time
* Pastdue\_1-29 means due less than 1 month
* Pastdue\_30-59 means due greater than 1 month
* Pastdue\_60-89 means due greater than 2 month
* Pastdue\_90-119 means due greater than 3 month
* Pastdue\_120-149 means due greater than 4 month
* Pastdue\_over-150 means due greater than 5 month
* X means no\_loan

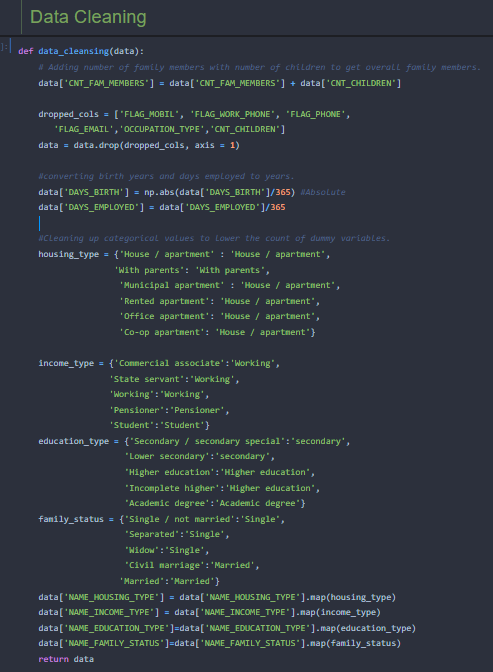
Output:-



**Data Cleaning:- Data Set app**

Dropping some more unwanted columns like:- FLAG\_MOBIL, FLAG\_WORK\_PHONE, FLAG\_PHONE, FLAG\_EMAIL, OCCUPATION\_TYPE, CNT\_CHILDREN

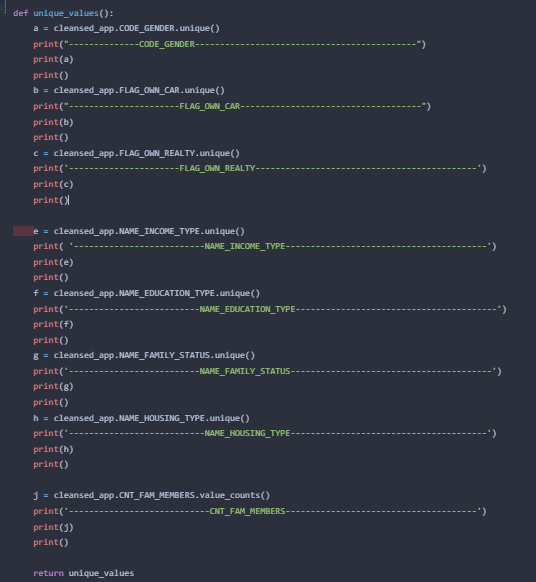
* DAYS\_BIRTH converting it into a date column by dividing 365 and for negative values applying absolute to convert into a positive number because days birth not in negative.
* Similarly we are converting DAYS\_EMPLOYED column but not applying absolute because negative values means past employee.



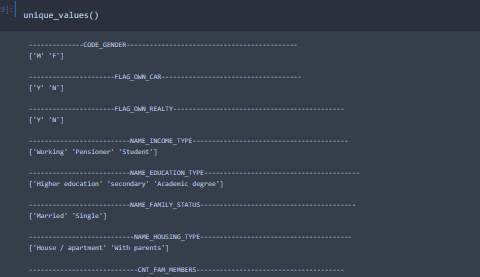
Output :-



Again checking the unique values present in given columns :-

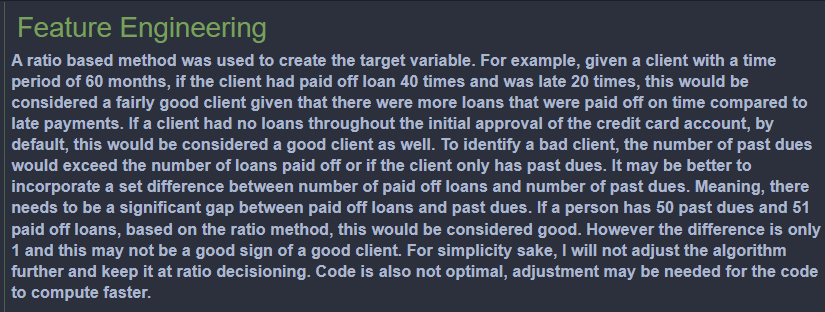


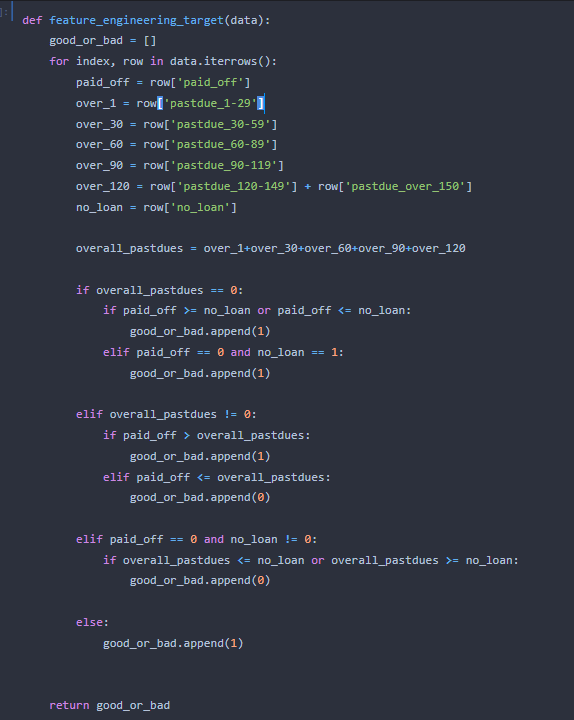
Output :-



Now this is looking nice.

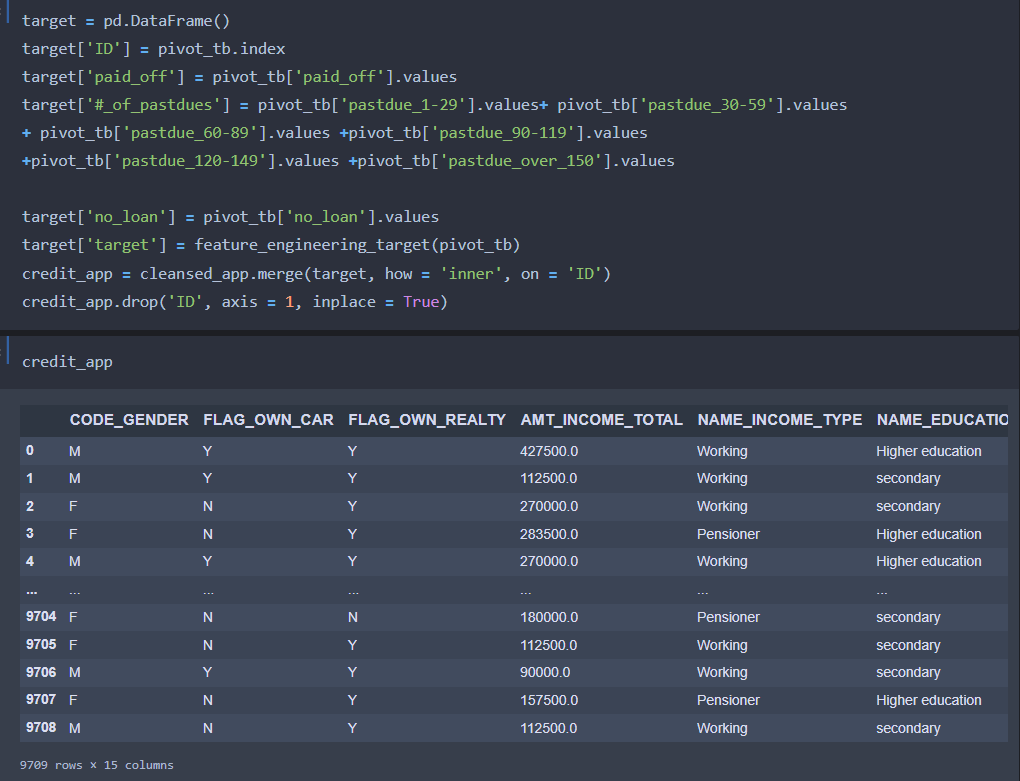
**Activity 7: Feature Engineering :- Data Set credit**

****

****

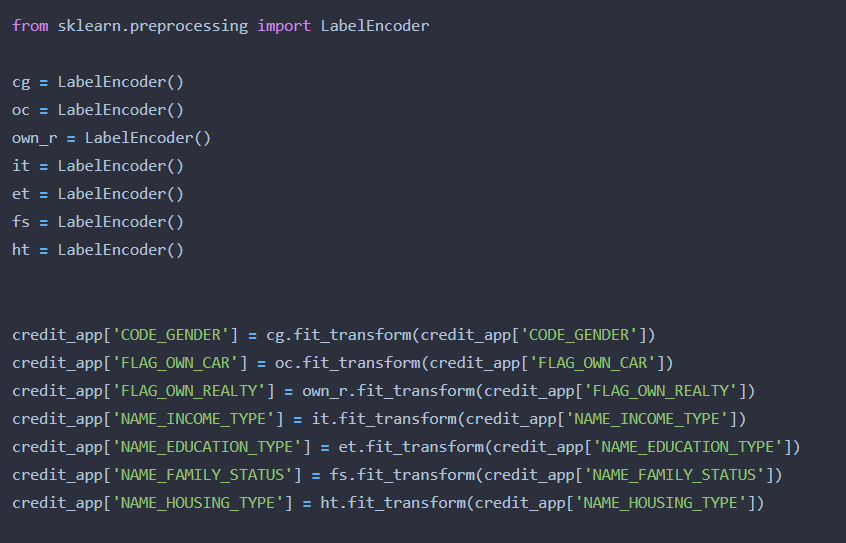
**Converting our credit data into binary format because at last we need to predict whether a person is eligible for credit card or not?**

**Converting our data into a target data frame and adding some more features**

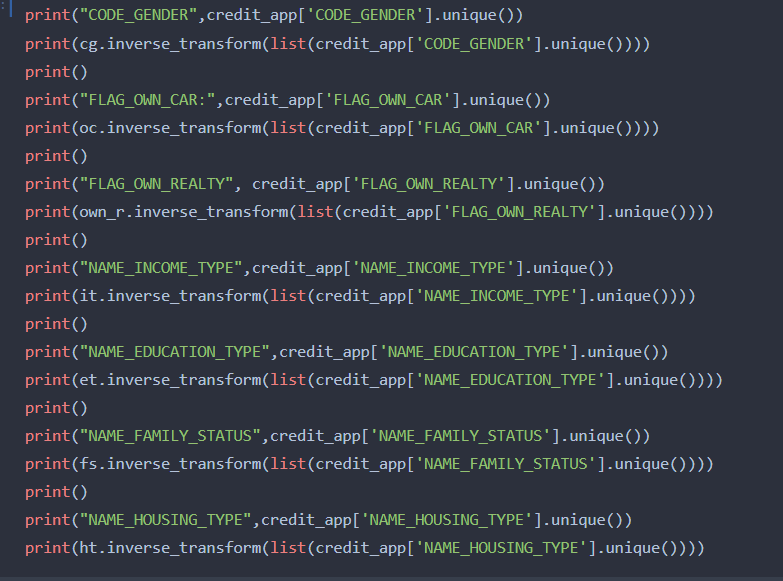
****

**Encoding :-**  For this data we are using label encoder for all categorical variable.

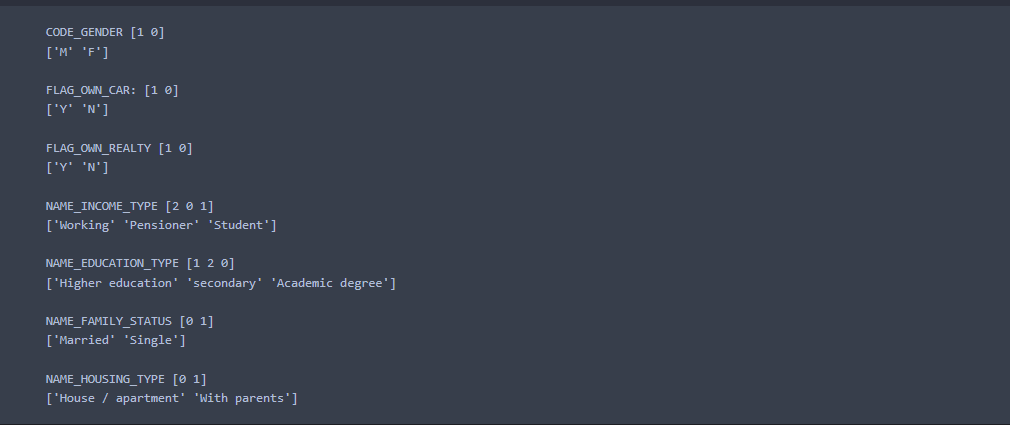
**Label Encoding** refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.



**inverse\_transform : -** Transform labels back to original encoding.

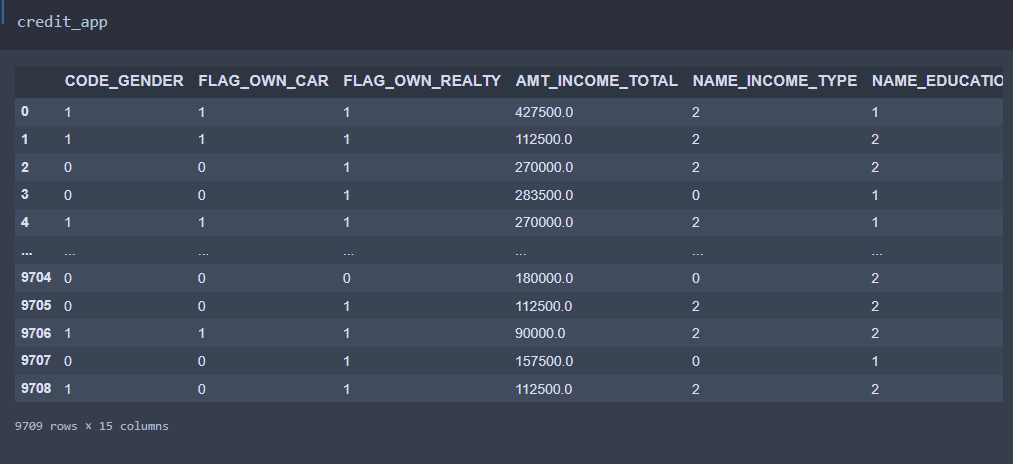


Output :-



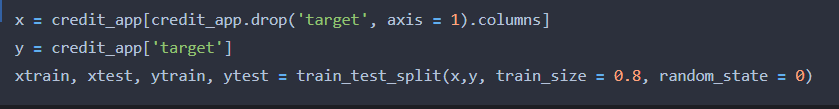
We are applied inverse\_transform because when we deployed our mode then we need the encoded value with original categories.

After encoding values looks like:-



**Activity 7: Splitting the Dataset into Dependent and Independent variable**

* In machine learning, the concept of dependent variable (y) and independent variables(x) is important to understand. Here, Dependent variable is nothing but output in dataset and independent variable is all inputs in the dataset.
* With this in mind, we need to split our dataset into the matrix of independent variables and the vector or dependent variable. Mathematically, Vector is defined as a matrix that has just one column.



And split the data into train and test in our case 0.8 means 80% data for training and remaining data for testing.

**Milestone 3: Model Buiding:**

Model building includes the following main tasks

* + Import the model building Libraries
  + Initializing the model
  + Training and testing the model
  + Evaluation of Model
  + Save the Model

**Activity 1: Training and Testing the Model**

* Once after splitting the data into train and test, the data should be fed to an algorithm to build a model.
* There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have it may be Classification algorithms are Regression algorithms.

1. Logistic Regression
2. Decision Tree Classifier
3. Random Forest Classifier
4. KNN
5. svm
6. xgboost Classifier

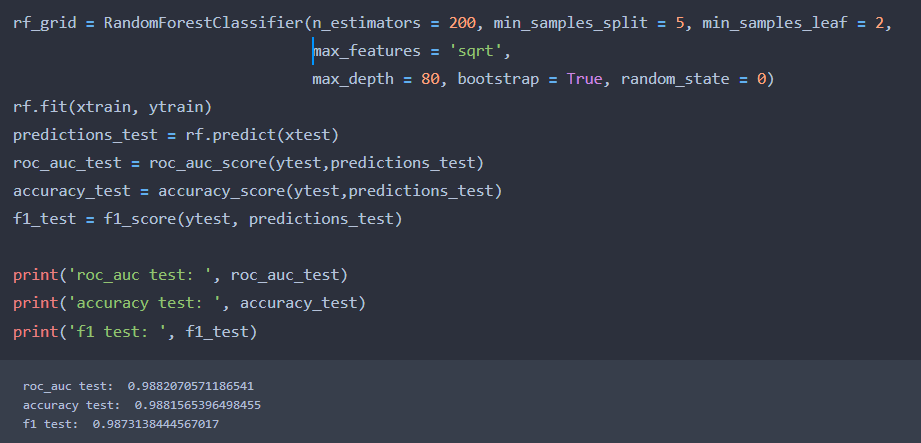
**Steps in Building the model:-**

* **Initialize the model**
* **Fit the models with x\_train and y\_train**
* **Predict the y\_train values and calculate the accuracy**
* **Predict the y\_test values and calculate the accuracy**

****

We’re going to use x\_train and y\_train obtained above in train\_test\_split section to train our Random Forest Classifier model. We’re using the fit method and passing the parameters as shown below.

**We are applying RandomizedsearchCV with random forest classifier with five fold cross validation to extract the best parameter. After that we fit our train and test data.**

**After training the model we need to check the accuracy of the model. In our case the model accuracy is 98%**

We are using the algorithm from Scikit learn library to build the model as shown below,

Once the model is trained, it’s ready to make predictions. We can use the **predict** method on the model and pass **x\_test** as a parameter to get the output as **predictions\_test.**

Notice that the prediction output is an array of real numbers corresponding to the input array.

**Activity 2: Model Evaluation**

After training the model, the model should be tested by using the test data which is been separated while splitting the data for checking the functionality of the model.

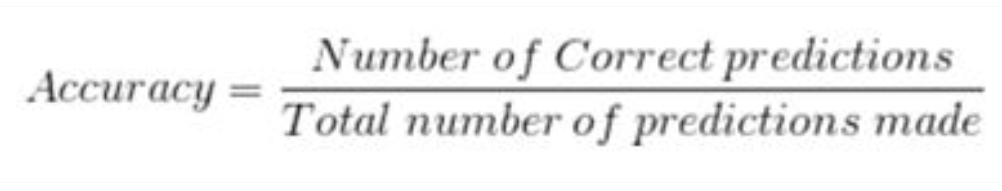
**Classification Evaluation Metrics:**

These model evaluation techniques are used to find out the accuracy of models built in classification type of machine learning models. We have three types of evaluation methods.

* Accuracy\_score
* Confusion matrix
* Roc- Auc test

1. Accuracy\_score

It is the ratio of number of correct predictions to the total number of input samples.



Select the model,which gives the best accuracy of all,and generate predictions and find the accuracy with training and testing data

2. Confusion Matrix

It is a matrix representation of the results of any binary testing



      Fig: Confusion Matrix of prediction of rainfall

1. True Positive: 98.94
2. True Negative: 1.22
3. False Positive: 1.06
4. False Negative: 98.78

3. Roc-Auc Curve

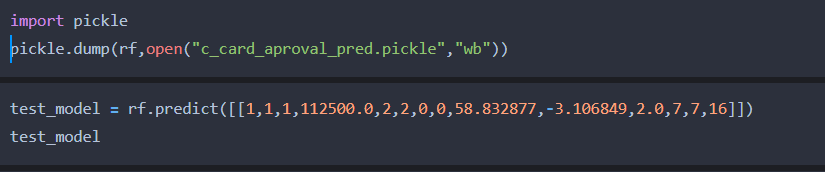
* AUC is the area under the ROC curve. AUC ROC indicates how well the probabilities from the positive classes are separated from the negative classes.

**Activity 3: Save the Model**

After building the model we have to save the model.

**Pickle** in **Python** is primarily **used** in serializing and deserializing a **Python** object structure. In other words, it's the process of converting a **Python** object into a byte stream to store it in a file/database, maintain program state across sessions, or transport data over the network. wb indicates write method and rd indicates read method.

**Saving the model :-**

****

**Milestone 4: Application Building**

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server side script

**Activity 1: Build HTML Code**

* + In this HTML page, we will create the front end part of the web page. In this page we will accept input from the user and Predict the values.

For more information regarding HTML

[**https://www.w3schools.com/html/**](https://www.w3schools.com/html/)

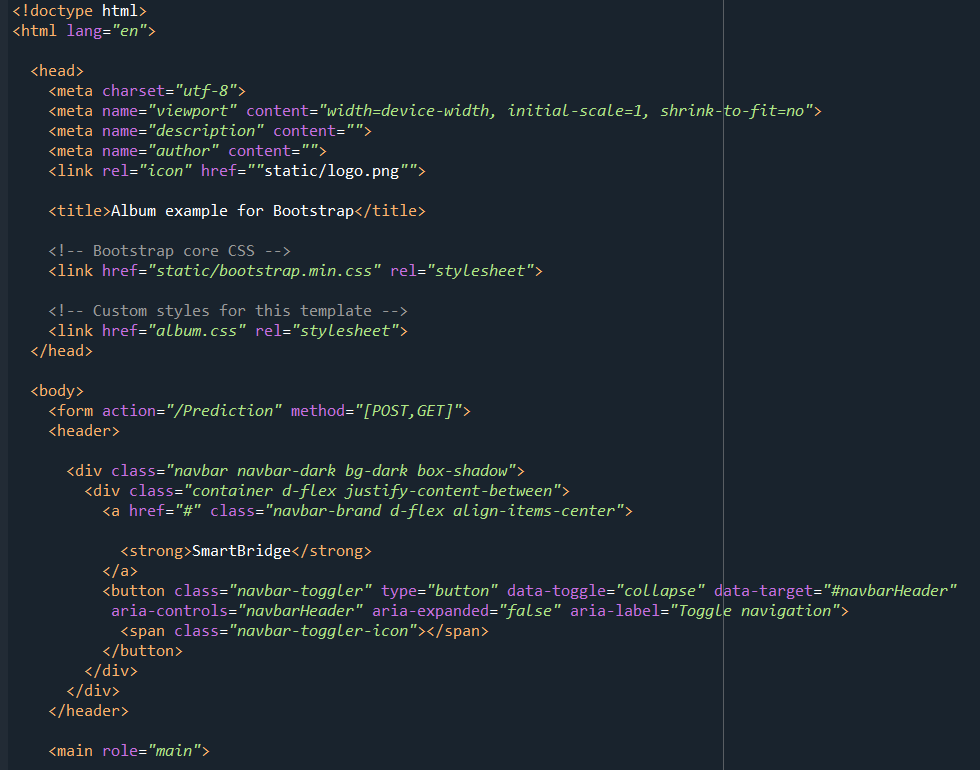
In our project we have 3 HTML files ,they are

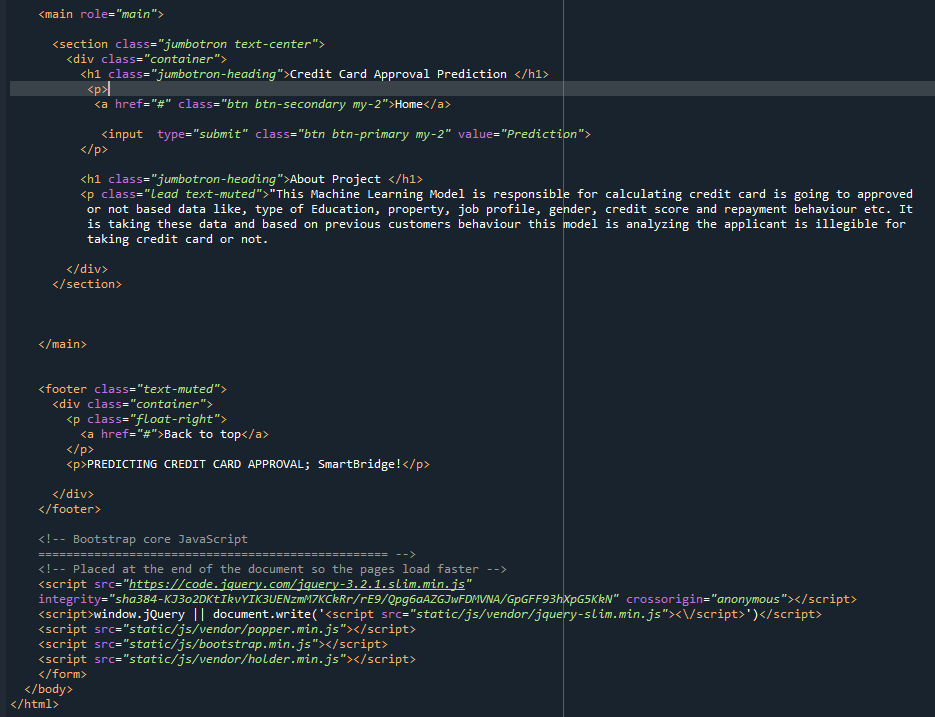
1.index.html

2.index1.html

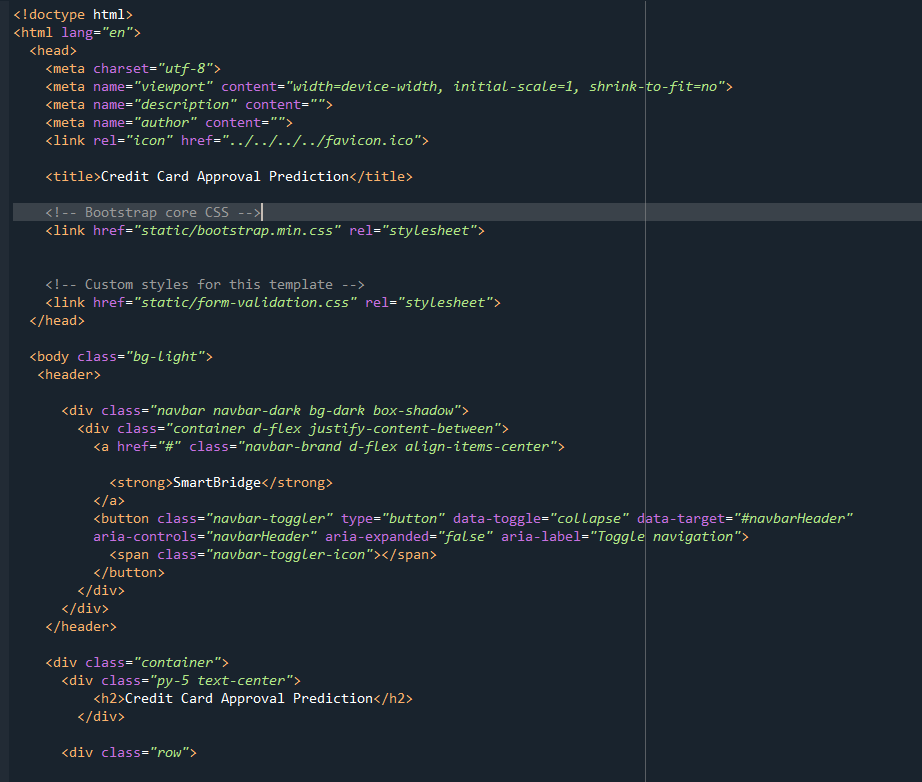
3.result.html

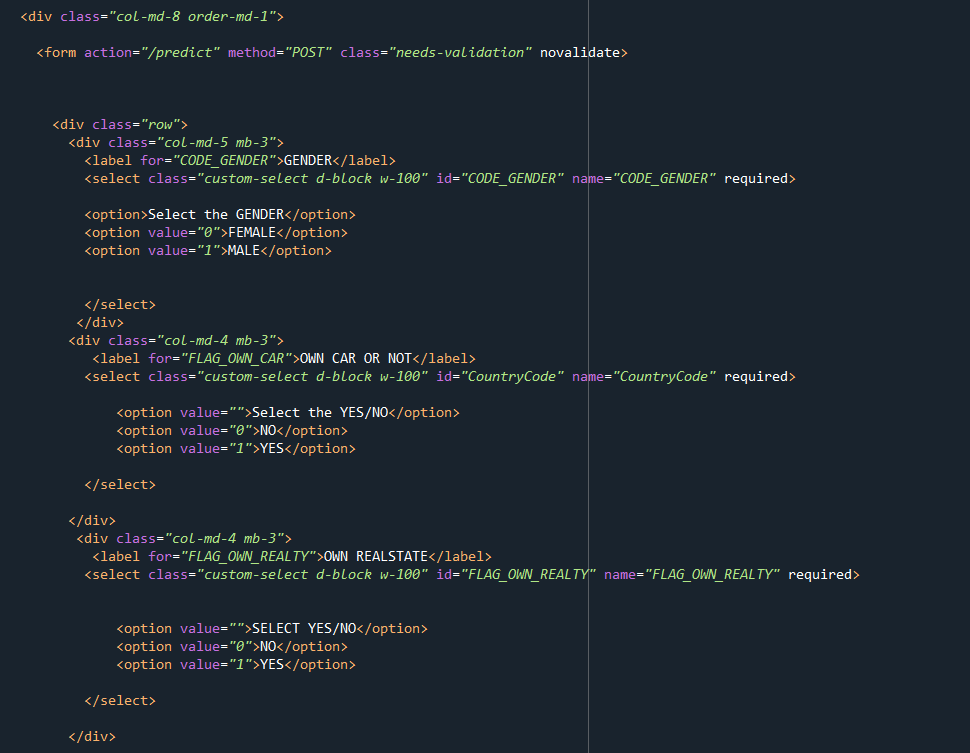
**index.html**

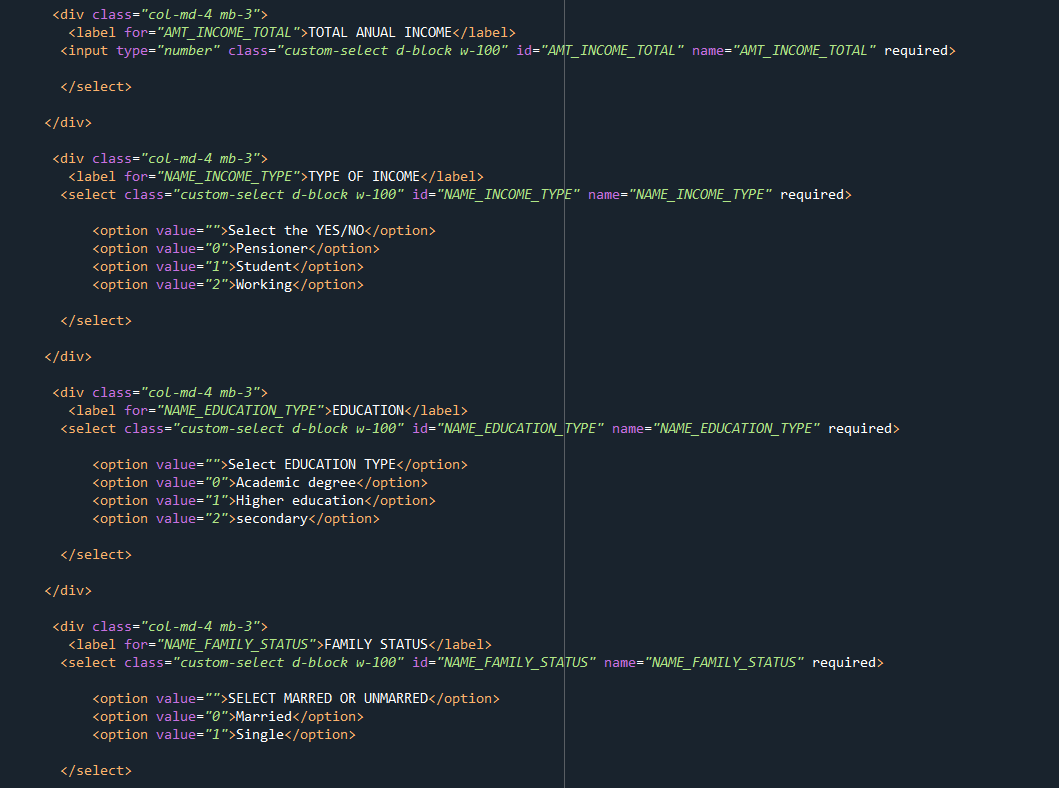
****

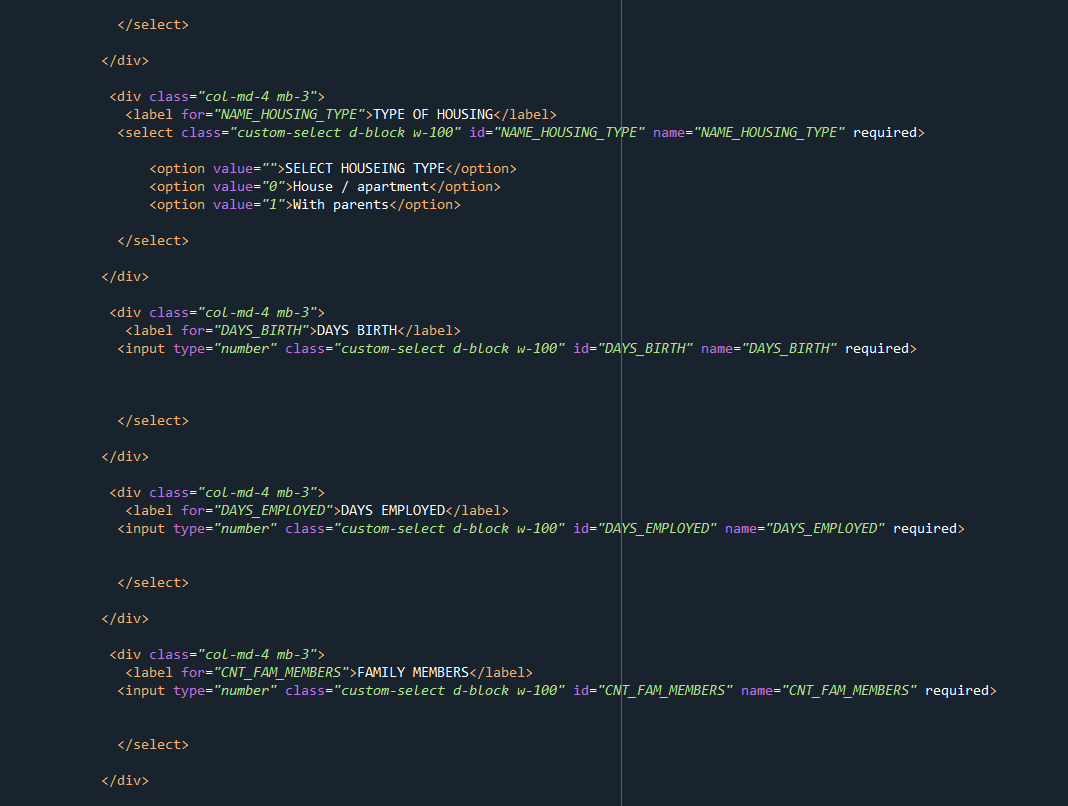
****

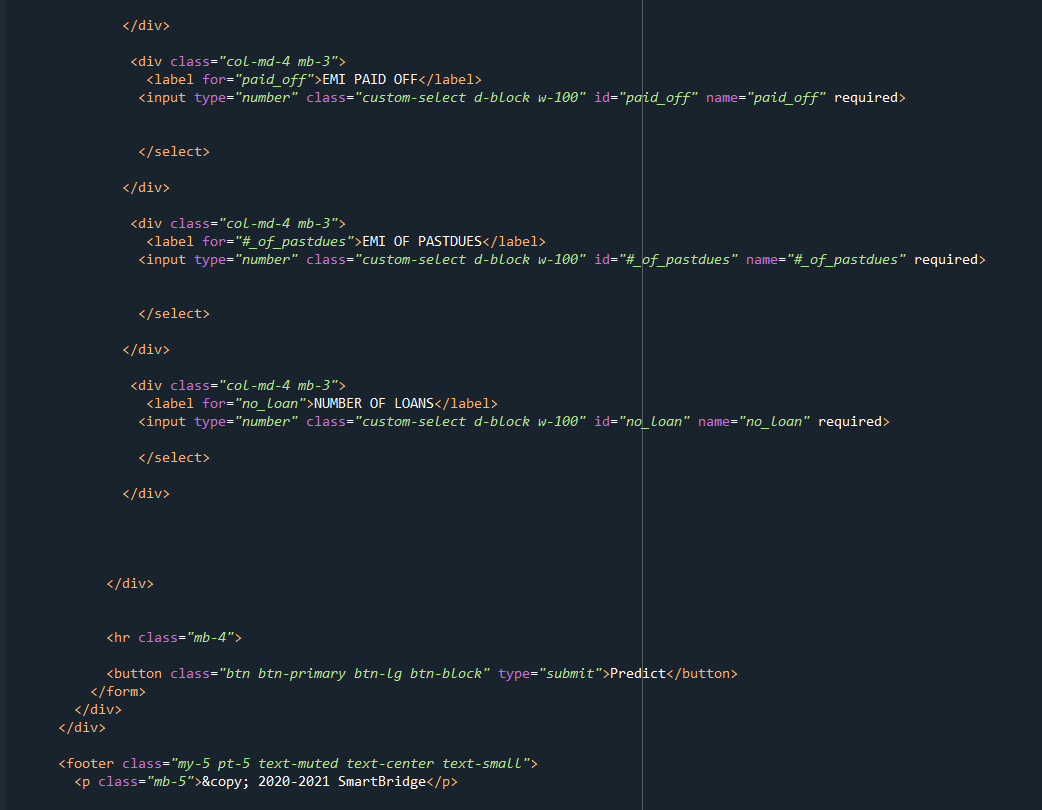
**Index1.html**

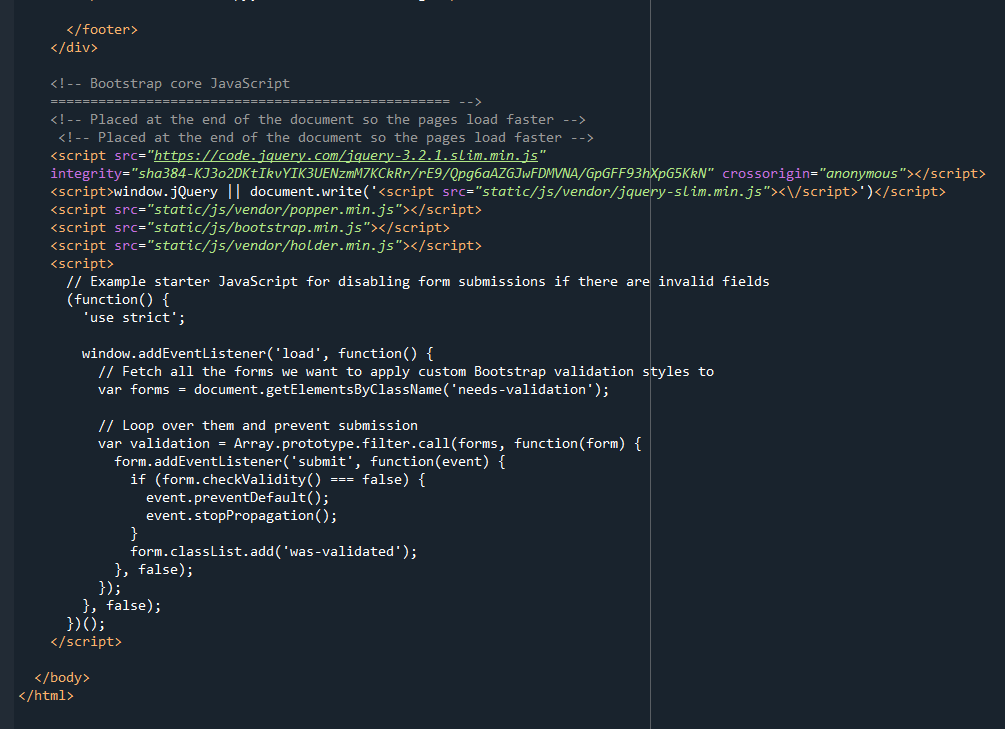
****

****

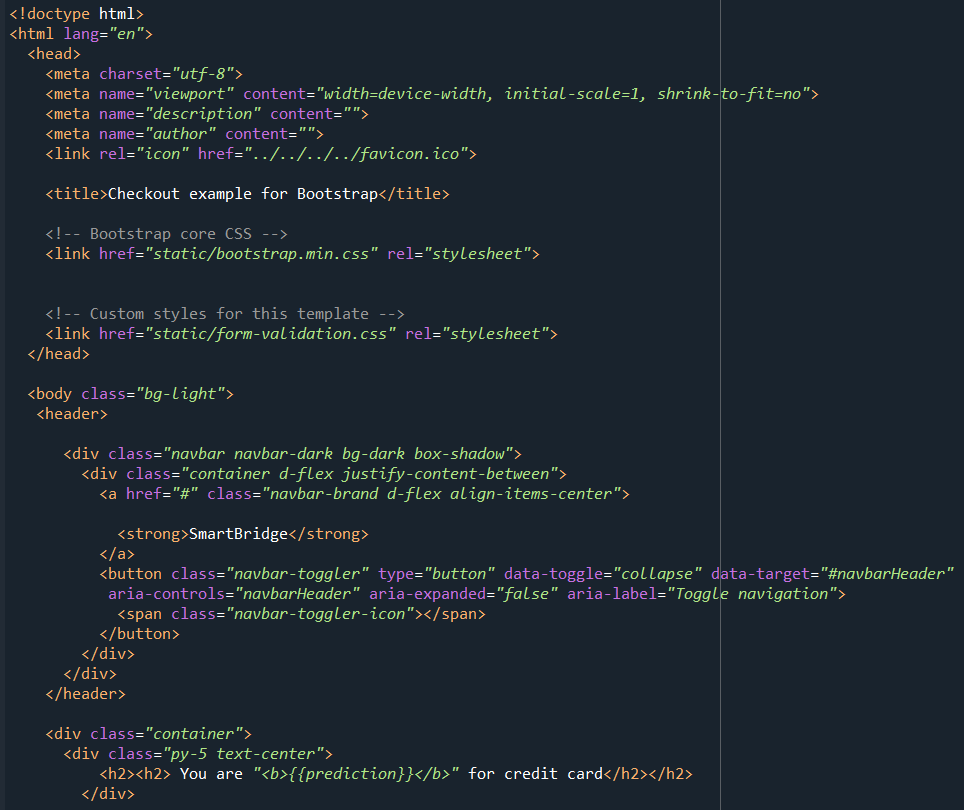
****

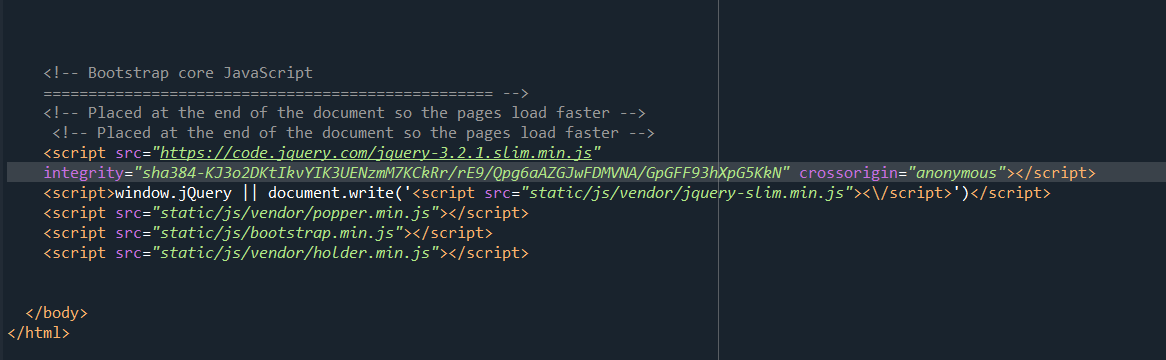
****

****

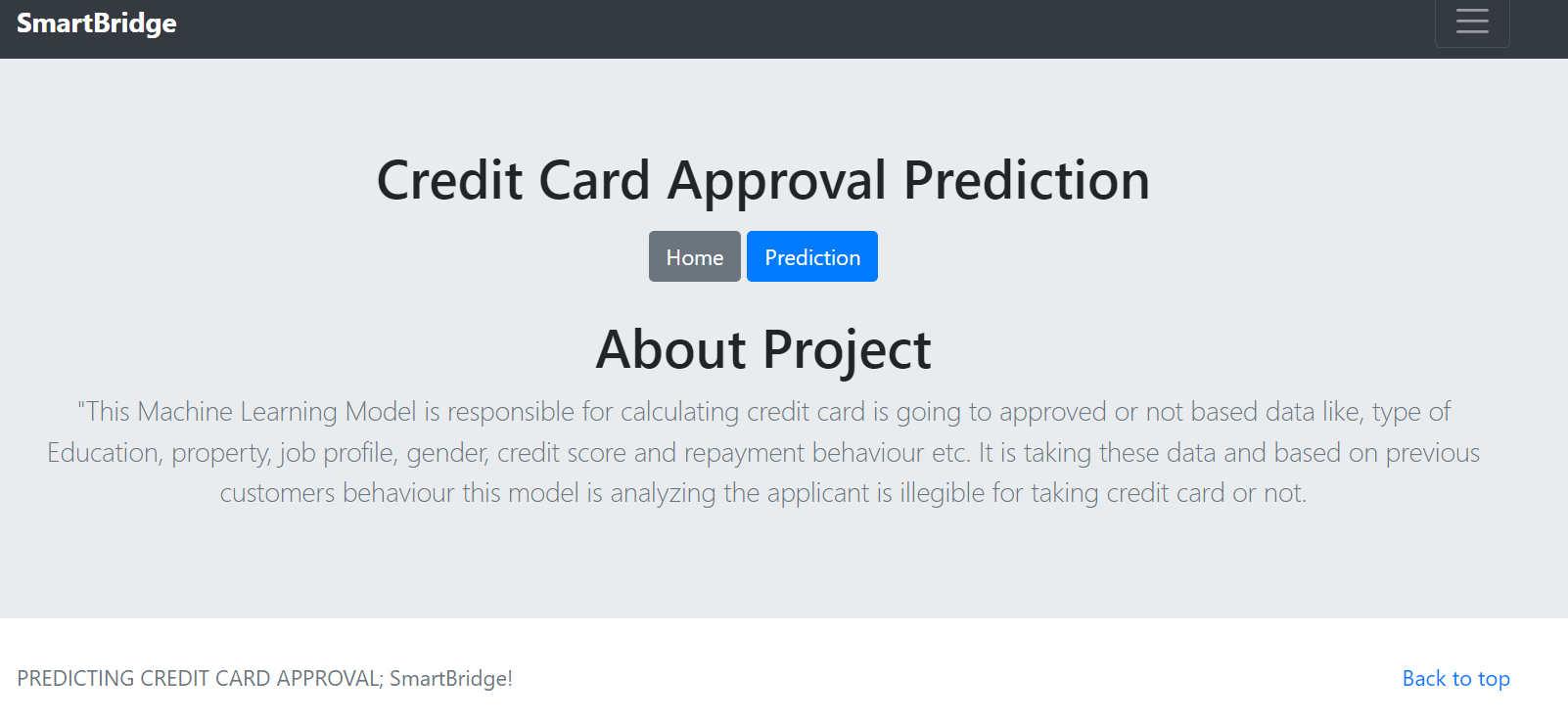
****

**Result.html**

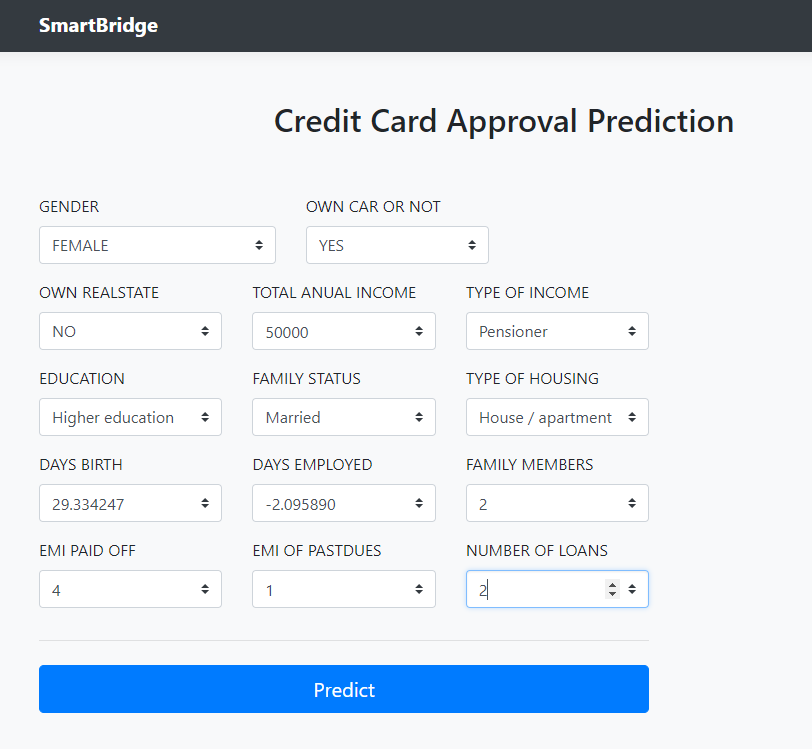
****

****

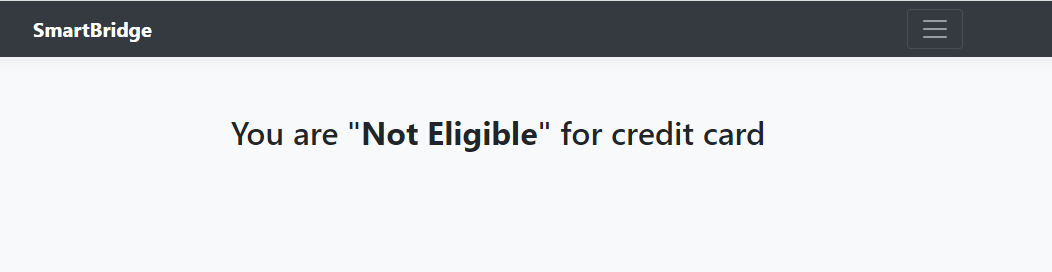
**The index.html page looks like**

****

**Index1.html**

****

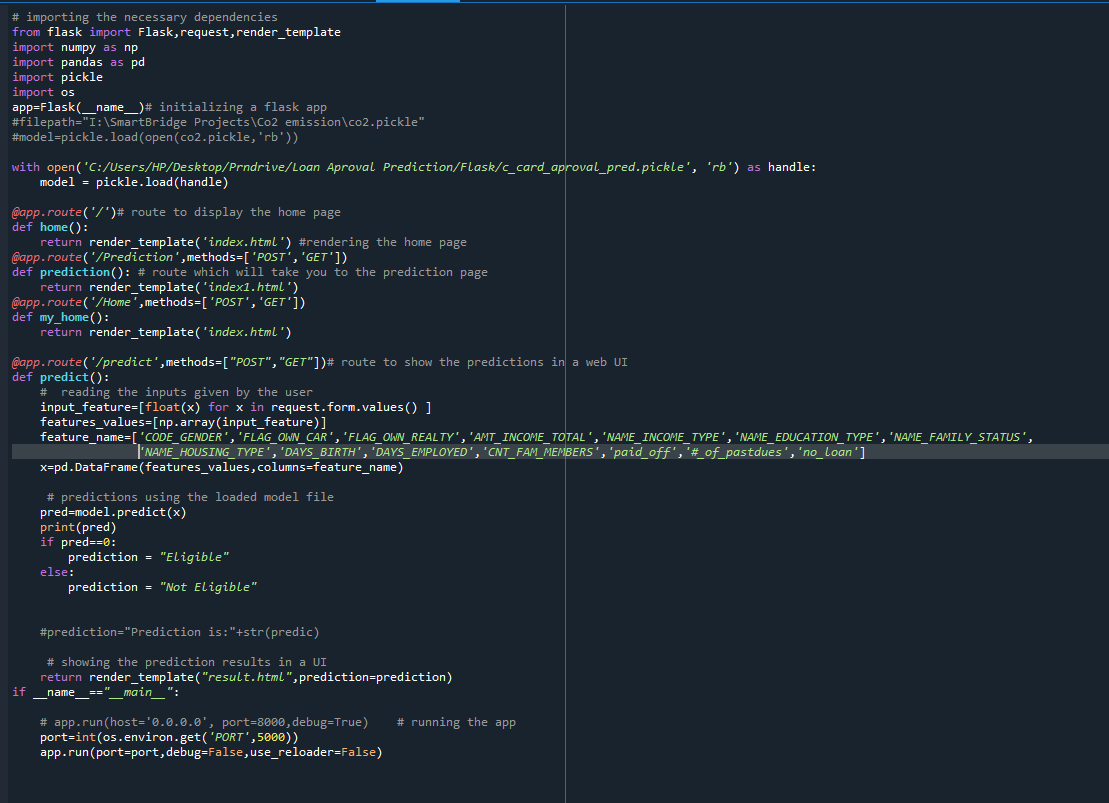
**The result.html page look like this**

****

**Activity 2: Main Python Script**

Let us build app.py flask file which is a web framework written in python for server-side scripting. Let’s see step by step procedure for building the backend application.

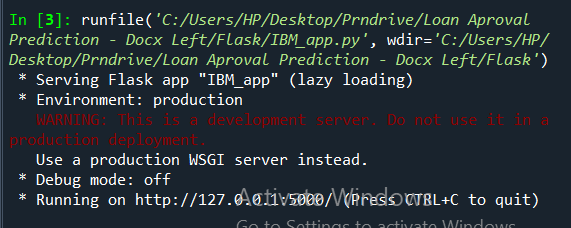
* In order to develop web api with respect to our model, we basically use Flask framework which is written in python.
* We are importing necessary libraries like Flask to host our model request
* Initialise the Flask application
* Loading the model using pickle
* Routes the api url
* Rendering the template. This helps to redirect to home page. In this home page ,we give our input and ask the model to predict
* we are taking the inputs from the form
* Predicting the values given by the user
* If output is zero output is Eligible
* If output is one output is Not Eligible
* At last The value of \_\_name\_\_ is set to \_\_main\_\_ when module run as main program other wise it is set to name of the module

****

**Activity 3: Run the App**

* + Open anaconda prompt from the start menu
  + Navigate to the folder where your python script is.
  + Now type “python app.py” command

Navigate to the localhost where you can view your web page,Then it will run on local host:5000

****

**Activity 4:**

* Copy the http link and paste it in google link tab,it will display the form page
* Enter the values as per the form and click on predict buttion
* It will redirect to the page based on prediction output