**Bank Loan Classification**

*By: Bhuwan Khatiwada.*

# **Problem Statement:**

The task requires to analyze a dataset of bank loans to extract hidden meanings and information from the dataset and to create a machine learning algorithm capable of making predictions of bank loans from the dataset provided.

# **Dataset Description**

The dataset was provided in a spreadsheet format i.e. an Excel file. The dataset consisted of 16 columns including the target variable. The figure below shows the snapshot of the dataset.

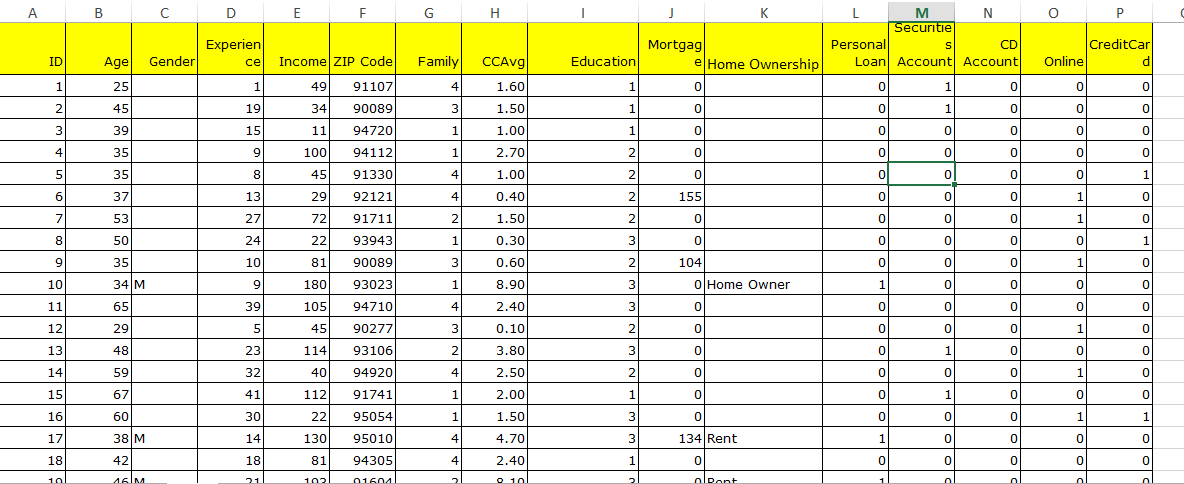


Figure 1: Bank Loan Dataset

The columns of the dataset are as follows:

* ID: ID of the customer
* Age: Age of the customer
* Gender: M for Male, F for Female and O for Others
* Experience: Amount of work experience in years
* Income: Amount of annual income (in thousands)
* Home Ownership: Home Owner, Rent and Home Mortgage.
* Zipcode: Postal code in which the client lives
* Family: Number of family members
* CCAvg: Average monthly spending with the credit card (in thousands)
* Education: Education level (1: bachelor's degree, 2: master's degree, 3: advanced/professional degree)
* Mortgage: Value of home mortgage, if any (in thousands)
* Securities Account: Does the customer have a securities account with the bank?
* CD Account: Does the customer have a certificate of deposit account (CD) with the bank?
* Online: Does the customer use the internet banking facilities?
* CreditCard: Does the customer use a credit card issued by the bank?
* Personal Loan: Did this customer accept the personal loan offered in the last campaign? (Target Variable)

# **Data Exploration and Preprocessing:**

The dataset was loaded into a pandas dataframe using the pandas module. Data exploration was performed using the several functions provided by the pandas library such as df.head(), df.tail(), df.info() etc.

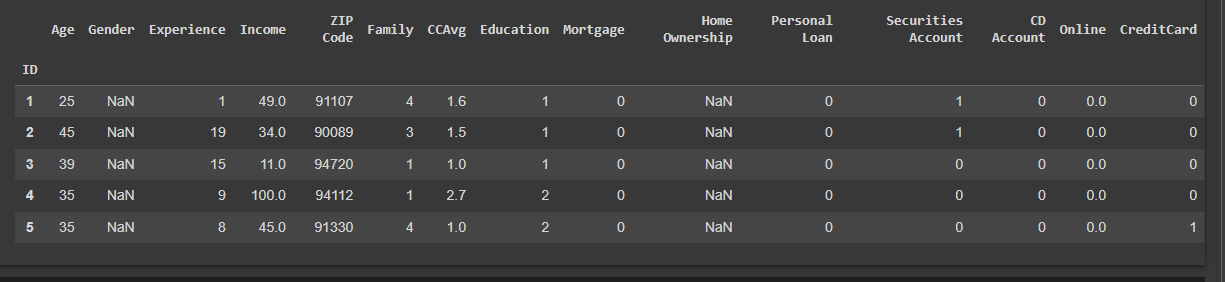


Figure 2: Output using the head() function

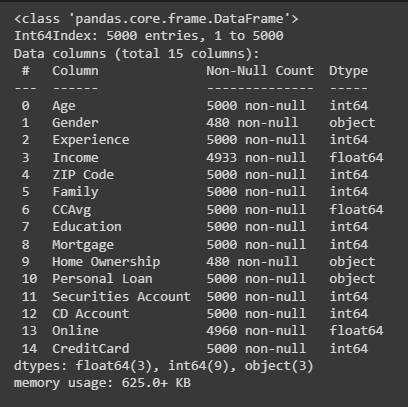


Figure 3:Output of info() function

The dataset has 5000 entries as seen from the info() function (also can be checked using df.shape()). Info function also returned the data types of the variables which helps us in better understanding the dataset.

After performing basics data exploration, it is necessary to preprocess the data for further processing. For the preprocessing phase following tasks were performed.

## **Dropping Columns**

The column gender has only 480 non-null values out of 5000 entries. This could be because people are not comfortable sharing their gender and chose not to give the information during data collection.

Normally, for missing values imputation could be applied but since the actual data is in very scare quantity imputation may cause the dataset to lose its original characteristics. Therefore, it makes more sense to completely drop the column instead.

Similar is the case with the case with the column "Home-ownership". Although, this variable may have a noteworthy impact on the target variable, due to the risk of hampering the characteristics of data this column will also be dropped.

## **Renaming Columns**

For the given dataset, there is no need for checking duplicate entries because the nature of variables allows for duplicates.

Here, the columns "Income", "CCAVG" and "Home Mortgage" have their values described in thousands. So, this information is added in respective column names for more clarity.

## **Changing Data Types**

Here, the data type of "ZIP Code" is "int64" but since we cannot perform any meaningful mathematical arithmetic computation with this variable, the data type of this variable is changed to "object" type.

Also, the "Education" variable is shown as of type "int64" while it is a categorical variable so converting the type of "Education" as well to “Category”

Similarly, the variables "Securities Account", "CD Account", "Online" and "CreditCard" are also going to be changed to categorical type.

Finally, the target variable "Personal Loan" is also changed to categorial type as there can only be two possible values for this variable.

## **Handling Empty Values**

The columns “Income(in thousands)” and “Online” has several empty values which needed to be filled. The empty values in “Online” column was replaced or filled using backfill and the empty values of the “Income(in thousands)” was replaced with the mean values of the income values.

## **Handling Outliers**

Finally, for the data preprocessing phase the outliers were removed from the columns in which they were detecting along with the whole entry relating to the outliers.

# **Descriptive Statistics**

The Pandas df.describe() was used to generate a table that gave results of certain statistics like mean, median, quartiles , min and max values etc.

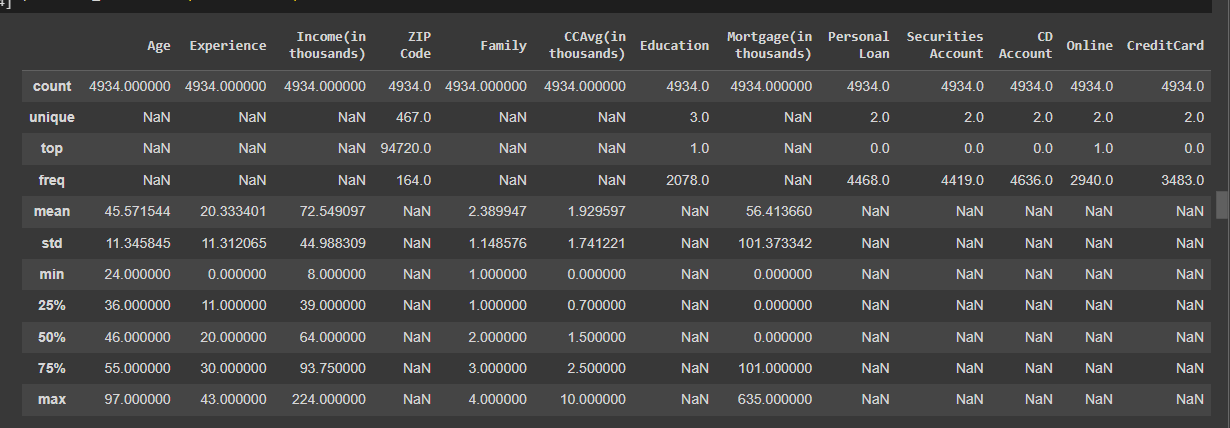


Figure 4 Descriptive Statistics of all columns

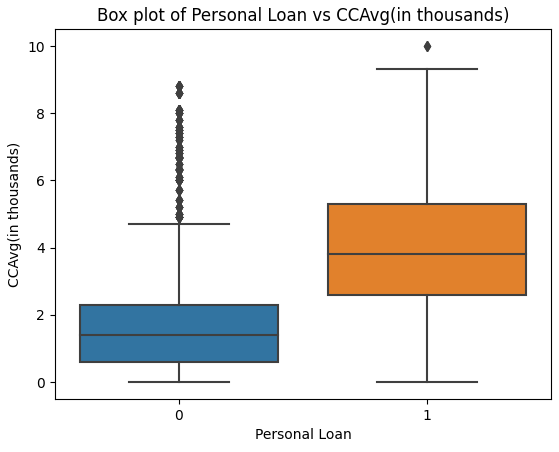
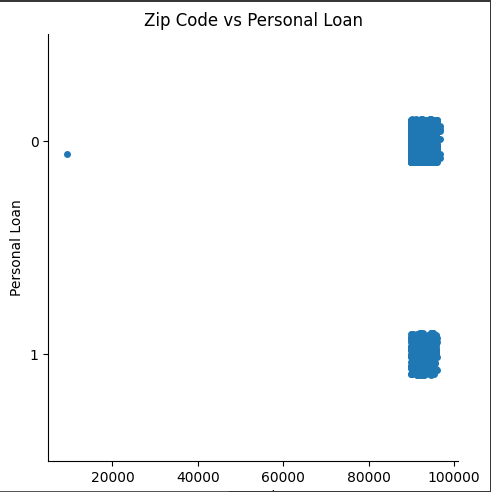
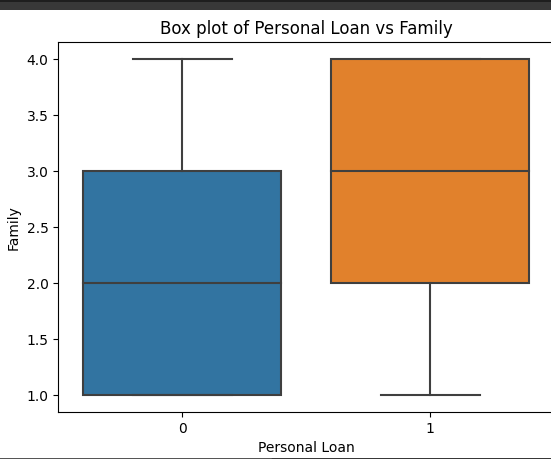
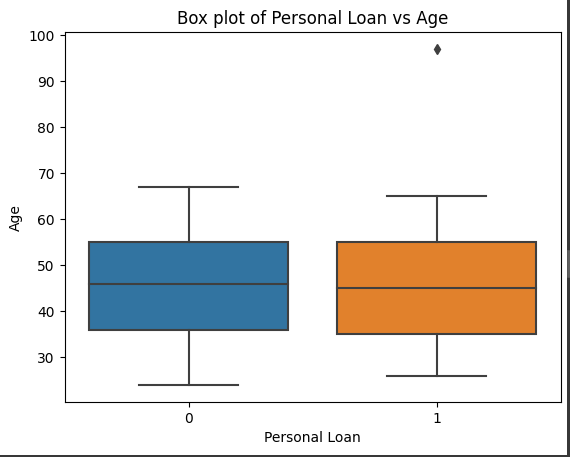
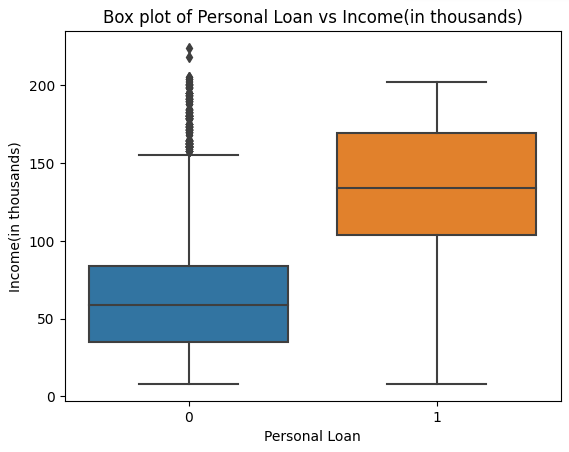
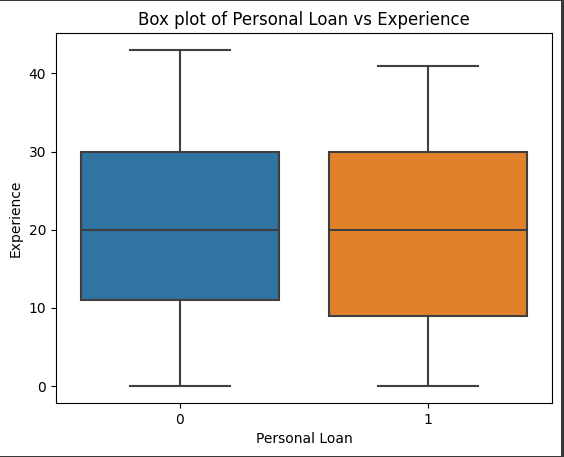
# **Data Visualization and Exploratory Data Analysis:**

Several plots were generated using different visualization libraries like Matplotlib and Seaborn. The visulizations were instrumental in showing the truths hidden inside the data.

Here are some of the findings,

* People with higher income were more likely to receive loan as shown by the plot.
* There is no relation between having an experience and getting a loan.
* Age also didn’t seem to have an impact in the chances of getting a loan.
* Its more likely to get a loan with higher number of family members.
* There is no relation with a Zip code and a persons chances of getting a loan.
* It is more likely to get a loan with Higher CCAvg value.

Figure5 Data Visualization Plots



# **Model Development**

For this task three models were used for experimental approach. The tree algorithms are DecisionTreeClassifier, RandomForestClassifier and HistGradientBoostingClassifier .

Using sklearn, the dataset was split into training and validation splits in ratio of 0.8 to 0.2.

For training, the target variable was removed from the dataset and other columns were used as features.

For experimentation, three different sets of features were used.:

* **Features-1:** All the columns of the **preprocessed** dataset.
* **Features-2:** The features which were assumed to have more impact based on the visualizations. i.e. "Income(in thousands)","CCAvg(in thousands)","Family", "Mortgage(in thousands)", "Securities Account","Online", "CD Account"
* **Features-3:** Only numerical variables from the above feature list.

For measuring the performance of the models accuracy was used as metric. The accuracy score of different models with different features are given below:

* **HistGradientBoostingClassifier**
  + **Feature-1:** 0.9837
  + **Feature-2:** 0.9452
  + **Feature-3:** 0.9442
* **DecisionTreeClassifier**
  + **Feature-1:** 0.9807
  + **Feature-2:** 0.9351
  + **Feature-3:** 0.9513
* **RandomForestClassifier**
  + **Feature-1:** 0.9767
  + **Feature-2:** 0.9503
  + **Feature-3:** 0.9493

# **Conclusion**

Here, the HistGradientBoostingClassiefier seemed to perform best among the used models and in all the models using the full processed datset as features produced the best results.