**Exploratory Data Analysis**

Data science is often thought to consist of advanced statistical and machine learning techniques. However, another key component to any data science endeavor is often undervalued or forgotten: exploratory data analysis (EDA). It is a classical and under-utilized approach that helps you quickly build a relationship with the new data.

It is always better to explore each data set using multiple exploratory techniques and compare the results. This step aims to understand the dataset, identify the missing values & outliers if any using visual and quantitative methods to get a sense of the story it tells. It suggests the next logical steps, questions, or areas of research for your project.

## **Steps in Data Exploration and Preprocessing:**

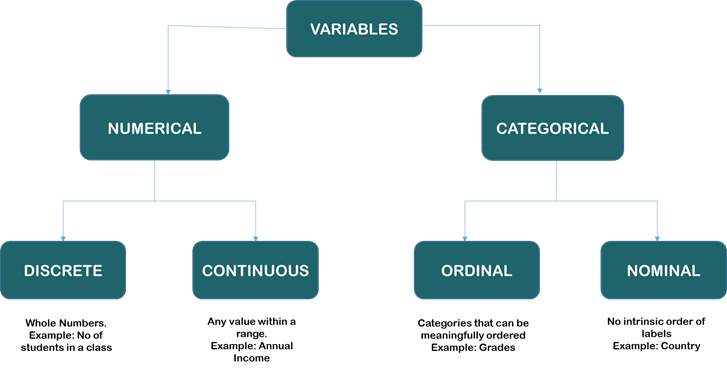
1. Identification of variables and data types
2. Analyzing the basic metrics
3. Non-Graphical Univariate Analysis
4. Graphical Univariate Analysis
5. Bivariate Analysis
6. Variable transformations
7. Missing value treatment
8. Outlier treatment
9. Correlation Analysis
10. Dimensionality Reduction

## **Dataset:**

To share my understandings and techniques, I know, I’ll take an example of a dataset from a recent Analytics Vidhya Website’s competition — [Loan Default Challenge](https://datahack.analyticsvidhya.com/contest/ltfs-datascience-finhack-an-online-hackathon/). Let’s try to catch hold of a few insights from the data set using EDA. The sample dataset contains 29 columns and 233155 rows.

## **Variable identification:**

The very first step in exploratory data analysis is to identify the type of variables in the dataset. Variables are of two types — Numerical and Categorical. They can be further classified as follows:



* **Classification of Variables:**

Once the type of variables is identified, the next step is to identify the Predictor (Inputs) and Target (output) variables.

In the above dataset, the numerical variables are,

Unique ID, disbursed\_amount, asset\_cost, ltv, Current\_pincode\_ID, PERFORM\_CNS.SCORE, PERFORM\_CNS.SCORE.DESCRIPTION, PRI.NO.OF.ACCTS, PRI.ACTIVE.ACCTS, PRI.OVERDUE.ACCTS, PRI.CURRENT.BALANCE, PRI.SANCTIONED.AMOUNT, PRI.DISBURSED.AMOUNT, NO.OF\_INQUIRIES

And the categorical variables are,

branch\_id, supplier\_id, manufacturer\_id, Date.of.Birth, Employment.Type, DisbursalDate, State\_ID, Employee\_code\_ID, MobileNo\_Avl\_Flag, Aadhar\_flag, PAN\_flag, VoterID\_flag, Driving\_flag, Passport\_flag, loan\_default

The target value is **loan\_default,**and the rest 28 features can be assumed as the predictor variables.

## **Importing Libraries:**

#importing libraries

import pandas as pd

import numpy as np

import matplotlib as plt

import seaborn as sns

Pandas library is a data analysis tool used for data manipulation, Numpy for scientific computing, and Matplotlib & Seaborn for data visualization.

## **Importing Dataset:**

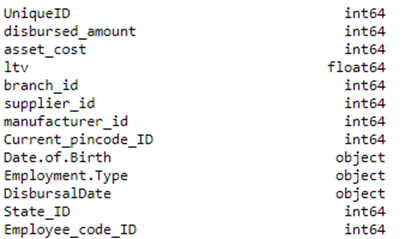
train = pd.read\_csv("/content/gdrive/MyDrive/dataset/train\_sample.csv")

Let’s import the dataset using the read\_csv method and assign it to the variable ‘train.’

## **Identification of data types:**

The .dtypes method to identify the data type of the variables in the dataset.

train.dtypes



A snippet of output for the above code

Both Date.of.Birth and DisbursalDate are of the object type. We have to convert it to DateTime type during data cleaning.

## **Size of the dataset:**

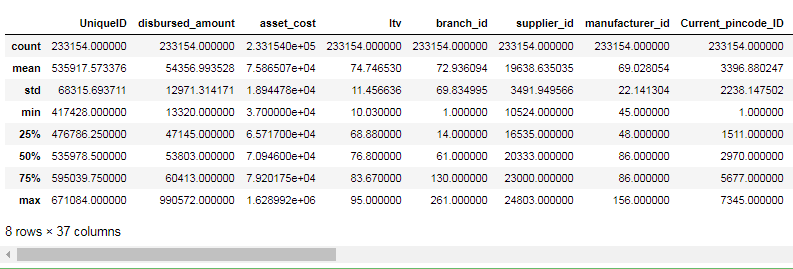
We can get the size of the dataset using the .shape method.

train.shape

## **Statistical Summary of Numeric Variables:**

Pandas describe() is used to view some basic statistical details like count, percentiles, mean, std, and maximum value of a data frame or a series of numeric values. As it gives the count of each variable, we can identify the missing values using this method.

train.describe()



A snippet of output for the above code

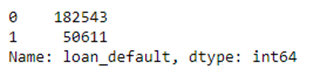
# ****Non-Graphical Univariate Analysis:****

## **To get the count of unique values:**

The value\_counts() method in Pandas returns a series containing the counts of all the unique values in a column. The output will be in descending order so that the first element is the most frequently-occurring element.

Let’s apply value counts to the loan\_default column.

train['loan\_default'].value\_counts()



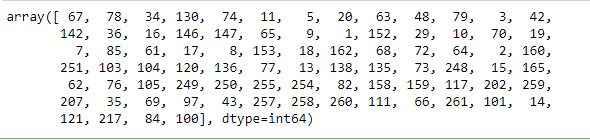
## **To get the list & number of unique values:**

The nunique() function in Pandas returns a series with several distinct observations in a column.

train['branch\_id'].nunique()

Similarly, the unique() function of pandas returns the list of unique values in the dataset.

train['branch\_id'].unique()

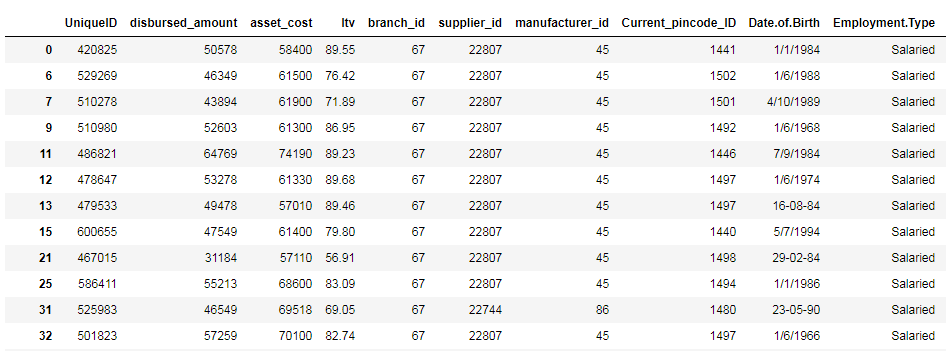


## **Filtering based on Conditions:**

Datasets can be filtered using different conditions, which can be implemented using logical operators in python. For example, == (double equal to), ≤ (less than or equal to), ≥(greater than or equal to), etc.

Let’s apply the same to our dataset and filter out the column which has the Employment.Type as “Salaried”

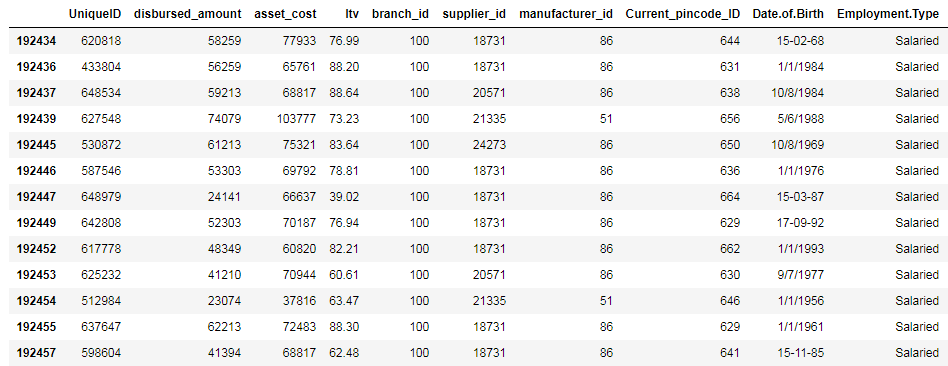
train[(train['Employment.Type'] == "Salaried")]



A snippet of output for the above code

Now let’s filter out the records based on two conditions using the AND (&) operator.

train[(train['Employment.Type'] == "Salaried") & (train['branch\_id'] == 100)]



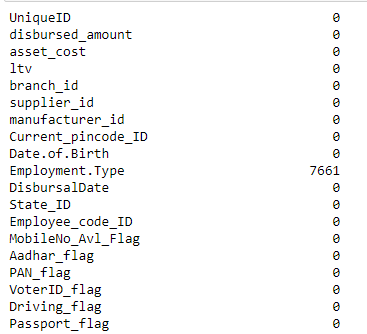
A snippet of output for the above code

You can try out the same example using the OR operator (|) as well.

## **Finding null values:**

When we import our dataset from a CSV file, many blank columns are imported as null values into the Data Frame, which can later create problems while operating that data frame. Pandas isnull() method is used to check and manage NULL values in a data frame.

train.apply(lambda x: sum(x.isnull()),axis=0)



A snippet of output for the above code

We can see that there are 7661 missing records in the column ‘Employment.Type’. These missing records should be either deleted or imputed in the data preprocessing stage. I will talk about different ways to handle missing values in detail in my next article.

**Data Type Conversion using to\_datetime() and astype() methods:**

Pandas astype() method is used to change the data type of a column. to\_datetime() method is used to change, particularly to DateTime type. When the data frame is imported from a CSV file, the data type of the columns is set automatically, which many times is not what it actually should have. For example, in the above dataset, Date.of.Birth and DisbursalDate are both set as object type, but they should be DateTime.

* **Example of to\_datetime():**

train['Date.of.Birth']= pd.to\_datetime(train['Date.of.Birth'])

* **Example of astype():**

train['ltv'] **=** train['ltv'].astype('int64')

# Graphical Univariate Analysis:

## **Bar Plot:**

## The **barplot()** shows the relation between a categorical variable and a continuous variable. The data is represented in rectangular bars where the length the bar represents the proportion of the data in that category.

from matplotlib import pyplot as plt

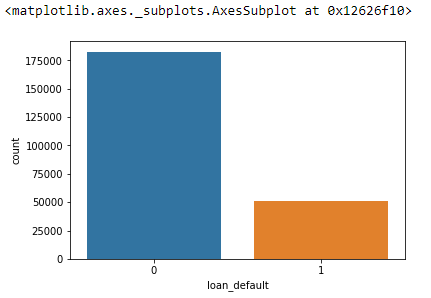
sns.barplot(x = "Employment.Type", y = "NO.OF\_INQUIRIES", hue = "loan\_default", data = train)

plt.show()

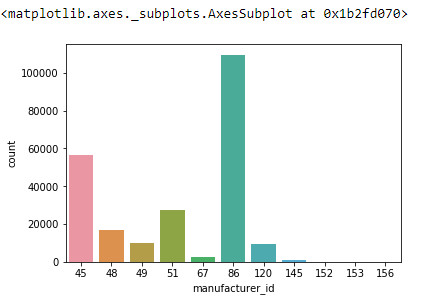
## **Count Plots:**

A count plot can be thought of as a histogram across a categorical, instead of numeric, variable. It is used to find the frequency of each category.

sns.countplot(train.loan\_default)



sns.countplot(train.manufacturer\_id)



Here we can see that category “86” is dominating over the other categories.

These are the basic, initial steps in exploratory data analysis. I hope you found this short article helpful.

**Lab Activity**

**Task 1:** Implement the given steps on the provided dataset in the Class.

**Task 2:** Perform the complete Exploratory Data Analysis on any classification dataset.