Name: Yash Brid

Roll No.: 2019130008

Batch: TE COMPS Batch A

Course Code: DA (Data Analytics)

Experiment No. 1

Aim: Perform EDA such as number of data samples, number of features, number of classes, number of data samples per class, removing missing values, conversion to numbers, using seaborn library to plot different graphs.

Theory:

a) What is EDA?

Exploratory Data Analysis is used to explore data, and possibly formulate hypotheses that might cause new data collection and experiments. EDA focuses more narrowly on checking assumptions required for model fitting and hypothesis testing. It also checks while handling missing values and making transformations of variables as needed. EDA build a robust understanding of the data, issues associated with either the info or process. it's a scientific approach to get the story of the data.

TYPES OF EXPLORATORY DATA ANALYSIS:

- Univariate Non-graphical
- Multivariate Non-graphical
- Univariate graphical
- Multivariate graphical

b) Why to perform EDA?

EDA is one of the crucial step in data science that allows us to achieve certain insights and statistical measure that is essential for the business continuity, stockholders and data scientists. We initially make several hypothesis by looking at the data before we hit the modelling. And its quite a good practice cause that will engage you more with EDA part. EDA helps you in confirming and validating the hypothesis you make. And from here you start your feature engineering part and take a flight to machine learning modelling.

Collab Link:

https://colab.research.google.com/drive/1x54_aTJ_VWVt9VTisrJV1xgUe0CDQcaP?usp=sharing

Dataset:

https://docs.google.com/spreadsheets/d/1PUgfWyJSOjjB55KxOgAx6EYYbekeU31q/edit?usp=sharing&ouid=114792866370786386824&rtpof=true&sd=true

Github Link: https://github.com/bridyash13/Data_Analytics_Lab_Batch_A

Code:

a) Importing the required package

In [98]:
```
import pandas as pd
import os
```

b) Importing the dataset

In [99]:
```
raw_data = pd.read_excel('Customer.xlsx',nrows=10000)
raw_data
```

Out[99]:

	UserID	Age	Job	Marital Status	Education	Loan_Outstanding	Loan_Default_History	Own_Housing	Call_Duration	EVR	CPI	CCI	Convert
0	1328507	41	Services	Single	High School	Yes	No	Yes	264	-1.8	93.075	-47.1	yes
1	1331344	31	Unemployed	Single	9th Grade	No	No	Yes	55	-1.8	93.075	-47.1	no
2	1334159	34	Admin	Single	UG/PG	No	No	Yes	255	-1.8	93.075	-47.1	no
3	1336062	31	Admin	Single	UG/PG	Yes	No	No	85	-1.8	93.075	-47.1	no
4	1344735	41	Supervisor	Married	Diploma	No	Unknown	No	247	-1.8	93.075	-47.1	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9995	2531684	46	Manager	Married	UG/PG	No	No	No	87	1.4	93.918	-42.7	no
9996	2531940	35	Supervisor	Married	4th Grade	Yes	Unknown	No	180	1.4	93.918	-42.7	no
9997	2532782	35	Unknown	Single	High School	No	Unknown	Yes	277	1.4	93.918	-42.7	no
9998	2533135	30	Admin	Married	6th Grade	Yes	No	Yes	53	1.4	93.918	-42.7	no
9999	2537758	35	Self-employed	Married	UG/PG	No	No	No	100	1.4	93.918	-42.7	no

10000 rows × 13 columns

c) Removing unknown data

In [8]:
```
data.drop(data[(data['Job']=='Unknown') | (data['Loan_Outstanding']=='Unknown') | (data['Loan_Default_History']=='Unknown') | (da
data
```

Out[8]:

	UserID	Age	Job	Marital Status	Education	Loan_Outstanding	Loan_Default_History	Own_Housing	Call_Duration	EVR	CPI	CCI	Convert
0	1328507	41	Services	Single	High School	Yes	No	Yes	264	-1.8	93.075	-47.1	yes
1	1331344	31	Unemployed	Single	9th Grade	No	No	Yes	55	-1.8	93.075	-47.1	no
2	1334159	34	Admin	Single	UG/PG	No	No	Yes	255	-1.8	93.075	-47.1	no
3	1336062	31	Admin	Single	UG/PG	Yes	No	No	85	-1.8	93.075	-47.1	no
6	1349864	36	Supervisor	Married	9th Grade	Yes	No	Yes	99	-1.8	93.075	-47.1	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9992	2527260	49	Technician	Married	9th Grade	No	No	Yes	183	1.4	93.918	-42.7	no
9994	2530536	32	Student	Single	High School	No	No	Yes	853	1.4	93.918	-42.7	yes
9995	2531684	46	Manager	Married	UG/PG	No	No	No	87	1.4	93.918	-42.7	no
9998	2533135	30	Admin	Married	6th Grade	Yes	No	Yes	53	1.4	93.918	-42.7	no
9999	2537758	35	Self-employed	Married	UG/PG	No	No	No	100	1.4	93.918	-42.7	no

d) Metadata of the numerical data in the dataset

In [9]:
```
data.describe()
```

Out[9]:

	UserID	Age	Call_Duration	EVR	CPI	CCI
count	7.956000e+03	7956.000000	7956.000000	7956.000000	7956.000000	7956.000000
mean	5.237818e+06	40.393539	256.802036	-0.087406	93.286227	-39.085960
std	2.573983e+06	11.141041	262.114280	1.755057	0.445081	5.360007
min	1.325823e+06	17.000000	0.000000	-3.000000	92.201000	-47.100000
25%	2.796742e+06	32.000000	102.000000	-1.800000	93.075000	-47.100000
50%	5.132599e+06	38.000000	175.000000	1.400000	93.444000	-36.100000
75%	7.445578e+06	47.000000	317.000000	1.400000	93.444000	-36.100000
max	9.874218e+06	92.000000	4199.000000	1.400000	94.215000	-31.400000

e) Overview of the data in the dataset

In [10]:
```
print("\nEducation Levels:",data.Education.unique())
print("\nJobs:",data.Job.unique())
Ages = data.Age.unique()
Ages.sort()
print("\nAges:",Ages)
print("\nPeople with oustanding loans:",len(data[data['Loan_Outstanding']=='Yes']))
print("\nPeople who own a house:",len(data[data['Own_Housing']=='Yes']))
```

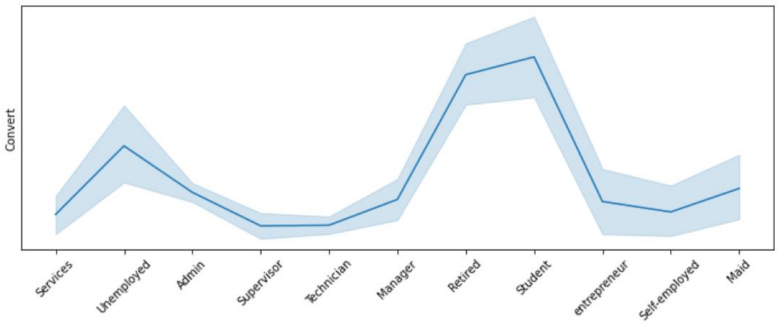
```
Education Levels: ['High School' '9th Grade' 'UG/PG' '6th Grade' 'Diploma' '4th Grade'
 'Unknown' 'Illiterate']

Jobs: ['Services' 'Unemployed' 'Admin' 'Supervisor' 'Technician' 'Manager'
 'Retired' 'Student' 'entrepreneur' 'Self-employed' 'Maid']

Ages: [17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64
 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88
 92]

People with oustanding loans: 1217

People who own a house: 4542
```

f) Data Visualization

In [12]:
```
import seaborn as sb
import matplotlib.pyplot as plt
```

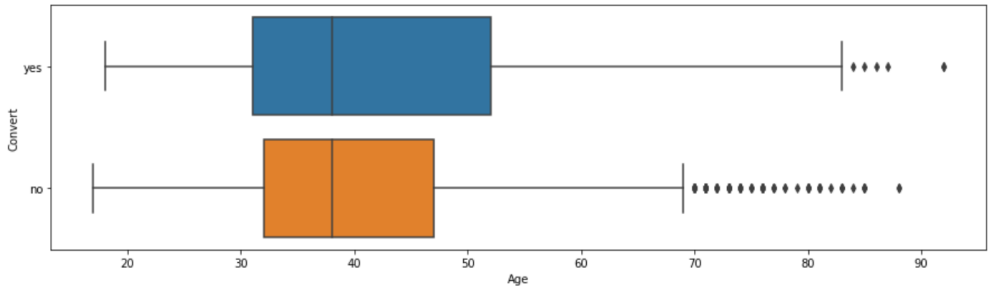
Lineplot of Customer conversion based on their occupation

In [13]:
```
plt.figure(figsize=(12,4))
sb.lineplot(x="Job", y="Convert", data=data)
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.show()
```



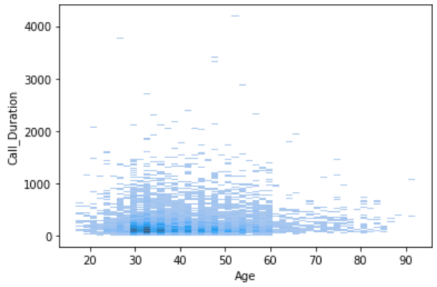
Boxplot of Customer conversion based on their age

```
In [14]: plt.figure(figsize=(15,4))
         sb.boxplot(x="Age",y="Convert",data=data)
         plt.show()
```



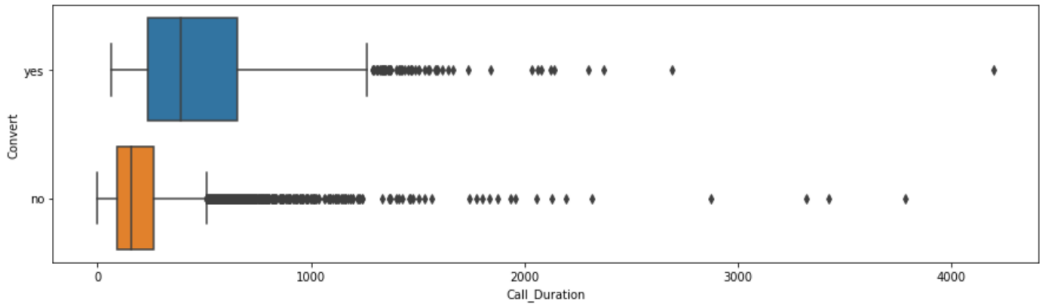
Histogram plot of call durations based on age

```
In [18]: sb.histplot(data,x="Age",y="Call_Duration",kde=True)
Out[18]: <AxesSubplot:xlabel='Age', ylabel='Call_Duration'>
```



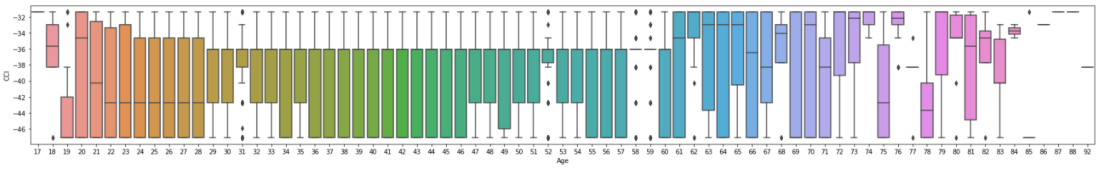
Boxplot of customer conversion based on call duration

```
In [205]: plt.figure(figsize=(15,4))
          sb.boxplot(x="Call_Duration",y="Convert",data=data)
          plt.show()
```



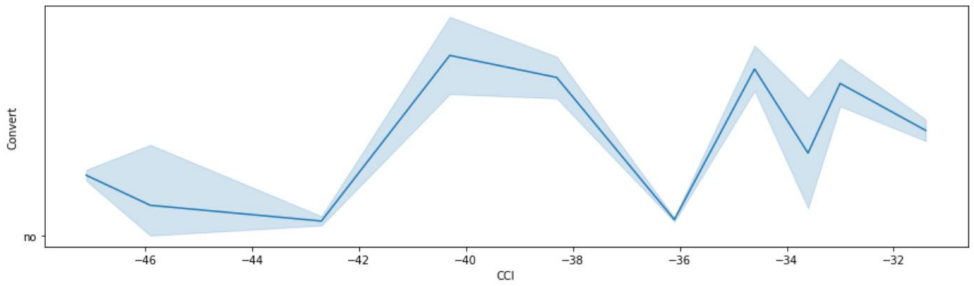
Boxplot of Customer Confidence Index based on their age

```
In [21]: plt.figure(figsize=(30,4))
         sb.boxplot(x="Age",y="CCI",data=data)
         plt.show()
```



LinePlot customer conversion on basis of Customer Confidence Index

```
In [25]: plt.figure(figsize=(15,4))
         sb.lineplot(x="CCI",y="Convert",data=data)
         plt.show()
```



Conclusion: In this experiment I learnt have learnt the following:

a) Working with panda and seaborn libraries.
b) Perform Exploratory Data Analysis methods such as removing unknown values, counting frequencies and values of features in a dataset.
c) Plot graphs of data from a dataset to better understand it.
d) Identify important features of a dataset based on numerical and visual data.