

BANK PERSONAL LOAN - VISUALIZATION

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
import seaborn as sns
```

```
In [8]: df=pd.read_csv("Bank_Personal_Loan.csv")
```

```
In [9]: df
```

Out[9]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Selected
0	1	25	1	49	91107	4	1.6	1	0	0	
1	2	45	19	34	90089	3	1.5	1	0	0	
2	3	39	15	11	94720	1	1.0	1	0	0	
3	4	35	9	100	94112	1	2.7	2	0	0	
4	5	35	8	45	91330	4	1.0	2	0	0	
...
4995	4996	29	3	40	92697	1	1.9	3	0	0	
4996	4997	30	4	15	92037	4	0.4	1	85	0	
4997	4998	63	39	24	93023	2	0.3	3	0	0	
4998	4999	65	40	49	90034	3	0.5	2	0	0	
4999	5000	28	4	83	92612	3	0.8	1	0	0	

5000 rows × 14 columns



```
In [10]: df.shape
```

Out[10]: (5000, 14)

```
In [11]: df.info
```

```
Out[11]: <bound method DataFrame.info of
ily CCAvg Education \
0      1    25      1    49    91107      4    1.6      1
1      2    45     19    34    90089      3    1.5      1
2      3    39     15    11    94720      1    1.0      1
3      4    35      9   100    94112      1    2.7      2
4      5    35      8    45    91330      4    1.0      2
...    ...    ...    ...    ...    ...    ...    ...    ...
4995  4996    29      3    40    92697      1    1.9      3
4996  4997    30      4    15    92037      4    0.4      1
4997  4998    63     39    24    93023      2    0.3      3
4998  4999    65     40    49    90034      3    0.5      2
4999  5000    28      4    83    92612      3    0.8      1

      Mortgage Personal Loan Securities Account CD Account Online \
0           0           0           1           0           0
1           0           0           1           0           0
2           0           0           0           0           0
3           0           0           0           0           0
4           0           0           0           0           0
...      ...      ...      ...      ...      ...
4995           0           0           0           0           1
4996          85           0           0           0           1
4997           0           0           0           0           0
4998           0           0           0           0           1
4999           0           0           0           0           1

      CreditCard
0           0
1           0
2           0
3           0
4           1
...      ...
4995           0
4996           0
4997           0
4998           0
4999           1

[5000 rows x 14 columns]>
```

```
In [12]: df.head(15)
```

Out[12]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securit Accol
0	1	25	1	49	91107	4	1.6	1	0	0	
1	2	45	19	34	90089	3	1.5	1	0	0	
2	3	39	15	11	94720	1	1.0	1	0	0	
3	4	35	9	100	94112	1	2.7	2	0	0	
4	5	35	8	45	91330	4	1.0	2	0	0	
5	6	37	13	29	92121	4	0.4	2	155	0	
6	7	53	27	72	91711	2	1.5	2	0	0	
7	8	50	24	22	93943	1	0.3	3	0	0	
8	9	35	10	81	90089	3	0.6	2	104	0	
9	10	34	9	180	93023	1	8.9	3	0	1	
10	11	65	39	105	94710	4	2.4	3	0	0	
11	12	29	5	45	90277	3	0.1	2	0	0	
12	13	48	23	114	93106	2	3.8	3	0	0	
13	14	59	32	40	94920	4	2.5	2	0	0	
14	15	67	41	112	91741	1	2.0	1	0	0	

In [13]:

df.tail(15)

Out[13]:

	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Se #
4985	4986	48	23	30	94720	3	1.70	2	162	0	
4986	4987	32	6	78	95825	1	2.90	3	0	0	
4987	4988	48	23	43	93943	3	1.70	2	159	0	
4988	4989	34	8	85	95134	1	2.50	1	136	0	
4989	4990	24	0	38	93555	1	1.00	3	0	0	
4990	4991	55	25	58	95023	4	2.00	3	219	0	
4991	4992	51	25	92	91330	1	1.90	2	100	0	
4992	4993	30	5	13	90037	4	0.50	3	0	0	
4993	4994	45	21	218	91801	2	6.67	1	0	0	
4994	4995	64	40	75	94588	3	2.00	3	0	0	
4995	4996	29	3	40	92697	1	1.90	3	0	0	
4996	4997	30	4	15	92037	4	0.40	1	85	0	
4997	4998	63	39	24	93023	2	0.30	3	0	0	
4998	4999	65	40	49	90034	3	0.50	2	0	0	
4999	5000	28	4	83	92612	3	0.80	1	0	0	

```
In [29]: print(df.dtypes)
```

```
Age                int64
Experience          int64
Income             int64
Family             int64
CCAvg              float64
Education           int64
Mortgage           int64
Personal Loan       int64
Securities Account int64
CD Account          int64
Online             int64
CreditCard          int64
dtype: object
```

```
In [30]: print(df.isnull().sum())
```

```
Age                0
Experience          0
Income             0
Family             0
CCAvg              0
Education           0
Mortgage           0
Personal Loan       0
Securities Account 0
CD Account          0
Online             0
CreditCard          0
dtype: int64
```

```
In [31]: print(df['Personal Loan'].value_counts())
```

```
0    4520
1     480
Name: Personal Loan, dtype: int64
```

```
In [32]: print(df.describe())
```

	Age	Experience	Income	Family	CCAvg \
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	45.338400	20.104600	73.774200	2.396400	1.937938
std	11.463166	11.467954	46.033729	1.147663	1.747659
min	23.000000	-3.000000	8.000000	1.000000	0.000000
25%	35.000000	10.000000	39.000000	1.000000	0.700000
50%	45.000000	20.000000	64.000000	2.000000	1.500000
75%	55.000000	30.000000	98.000000	3.000000	2.500000
max	67.000000	43.000000	224.000000	4.000000	10.000000

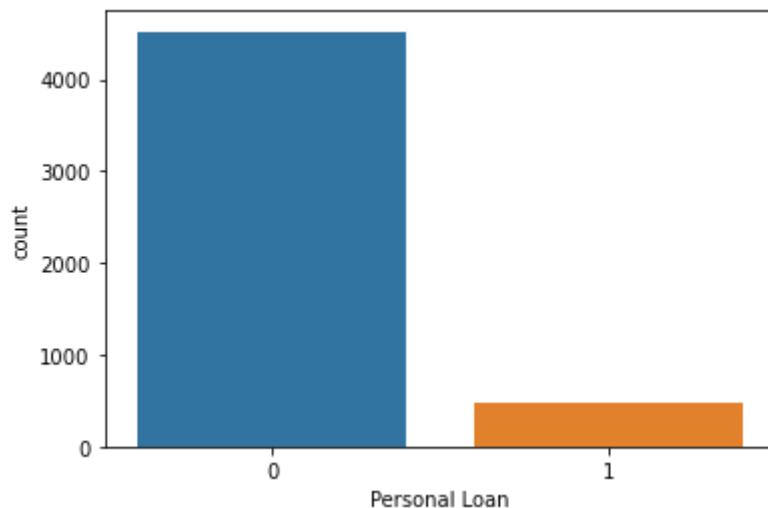
	Education	Mortgage	Personal Loan	Securities Account \
count	5000.000000	5000.000000	5000.000000	5000.000000
mean	1.881000	56.498800	0.096000	0.104400
std	0.839869	101.713802	0.294621	0.305809
min	1.000000	0.000000	0.000000	0.000000
25%	1.000000	0.000000	0.000000	0.000000
50%	2.000000	0.000000	0.000000	0.000000
75%	3.000000	101.000000	0.000000	0.000000
max	3.000000	635.000000	1.000000	1.000000

	CD Account	Online	CreditCard
count	5000.000000	5000.000000	5000.000000
mean	0.06040	0.596800	0.294000
std	0.23825	0.490589	0.455637
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	1.000000	0.000000
75%	0.000000	1.000000	1.000000
max	1.000000	1.000000	1.000000

Distribution of the target variable using countplot

```
In [33]: sns.countplot(x='Personal Loan', data=df)
```

```
Out[33]: <AxesSubplot:xlabel='Personal Loan', ylabel='count'>
```

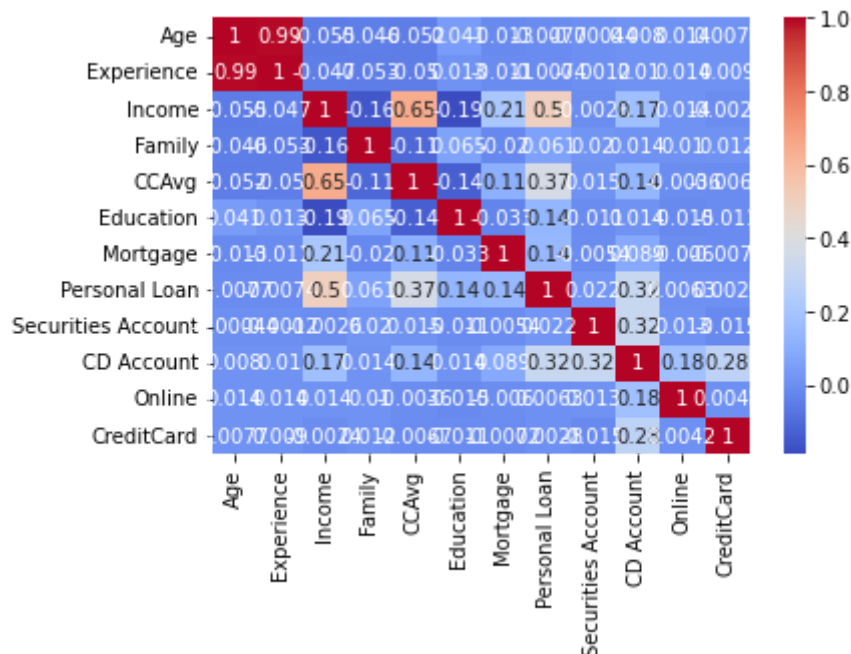


It appears that the dataset is imbalanced, with a small proportion of customers accepting a personal loan, as indicated by the count plot.

Correlation matrix using heatmap

```
In [35]: corr = df.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
```

```
Out[35]: <AxesSubplot:>
```

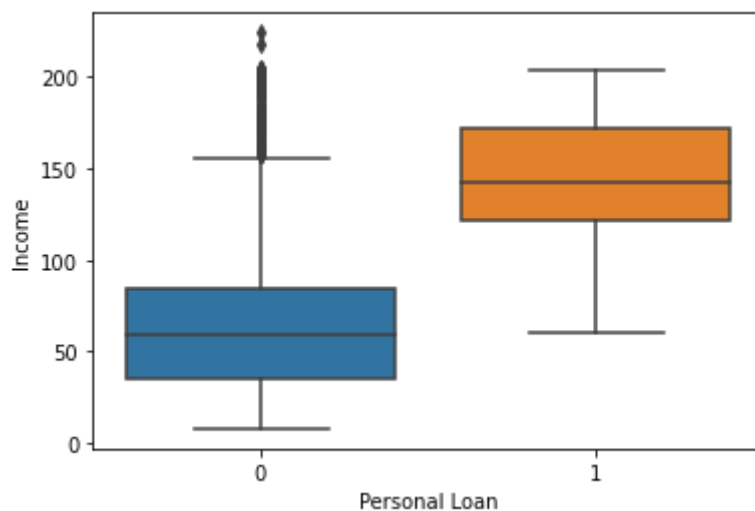


According to the correlation matrix, The most significant predictors of accepting a personal loan are income and education.

Distribution of income by personal loan using box plot

```
In [36]: sns.boxplot(x='Personal Loan', y='Income', data=df)
```

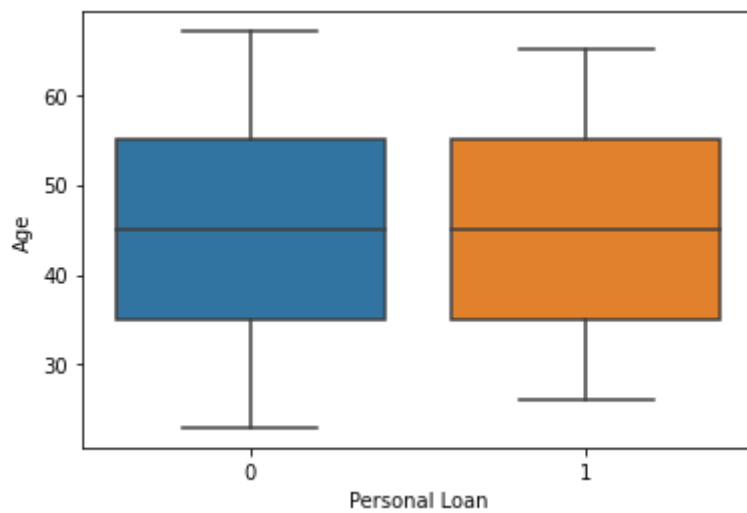
```
Out[36]: <AxesSubplot:xlabel='Personal Loan', ylabel='Income'>
```



Distribution of age by personal loan using box plot

```
In [37]: sns.boxplot(x='Personal Loan', y='Age', data=df)
```

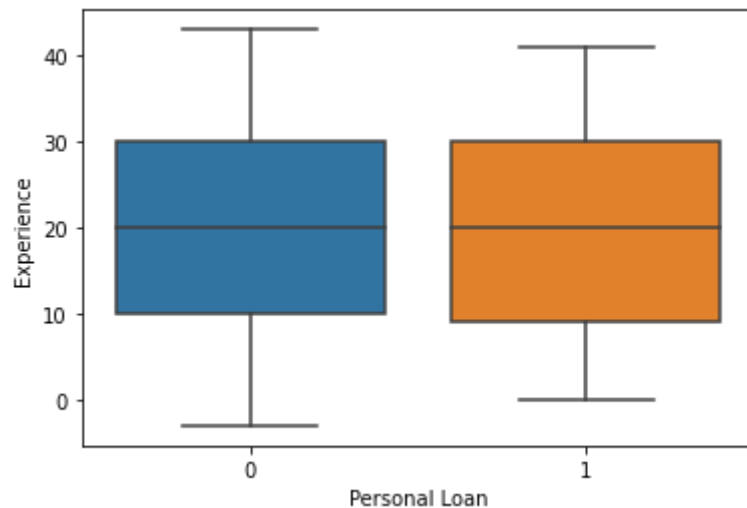
```
Out[37]: <AxesSubplot:xlabel='Personal Loan', ylabel='Age'>
```



Distribution of experience by personal loan using box plot

```
In [39]: sns.boxplot(x='Personal Loan', y='Experience', data=df)
```

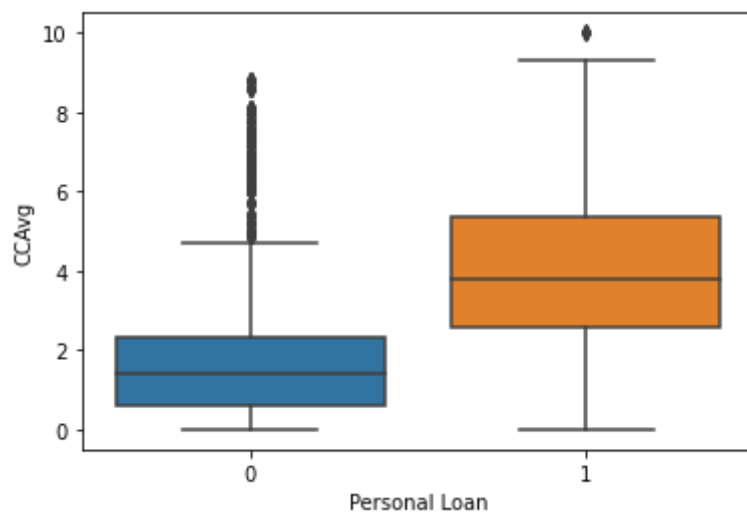
```
Out[39]: <AxesSubplot:xlabel='Personal Loan', ylabel='Experience'>
```



Distribution of CCAvg by personal loan using box plot

```
In [40]: sns.boxplot(x='Personal Loan', y='CCAvg', data=df)
```

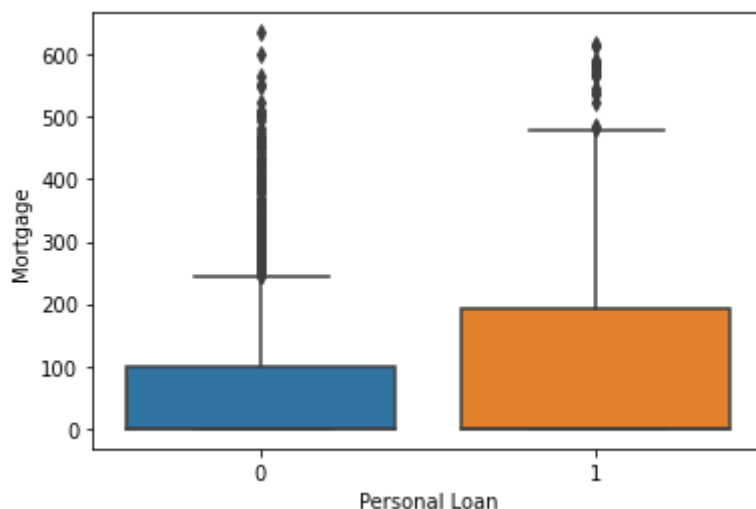
```
Out[40]: <AxesSubplot:xlabel='Personal Loan', ylabel='CCAvg'>
```



Distribution of mortgage by personal loan using box plot

```
In [41]: sns.boxplot(x='Personal Loan', y='Mortgage', data=df)
```

```
Out[41]: <AxesSubplot:xlabel='Personal Loan', ylabel='Mortgage'>
```

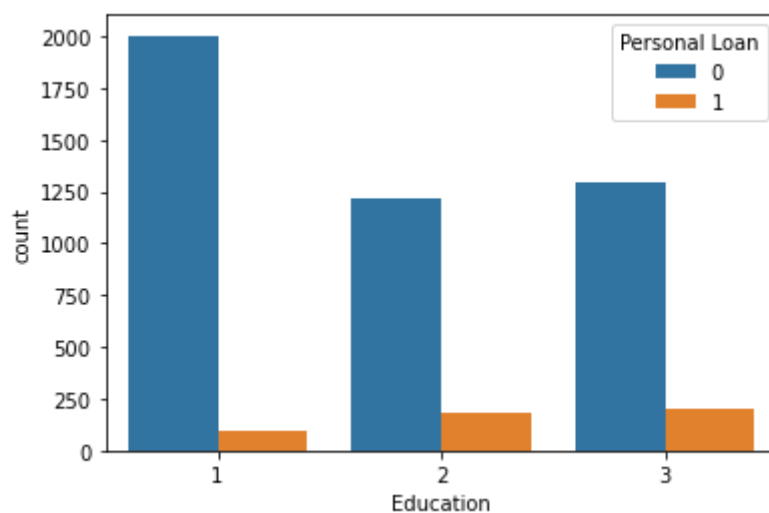


the boxplots suggest that customers who accepted personal loans tend to have higher incomes, more work experience, and slightly older ages compared to those who did not accept personal loans. Moreover, the boxplots reveal that customers who spent more on credit cards, had more mortgages, and more family members were more likely to accept personal loans.

Distribution of education by personal loan using count plot

```
In [42]: sns.countplot(x='Education', hue='Personal Loan', data=df)
```

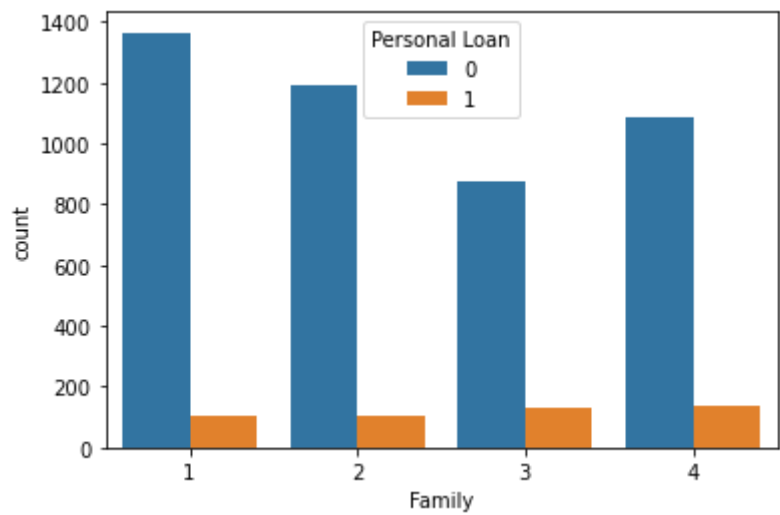
```
Out[42]: <AxesSubplot:xlabel='Education', ylabel='count'>
```



Distribution of family by personal loan using count plot

```
In [43]: sns.countplot(x='Family', hue='Personal Loan', data=df)
```

```
Out[43]: <AxesSubplot:xlabel='Family', ylabel='count'>
```

the count plots suggest that higher education levels are associated with a higher likelihood of accepting personal loans.