Analyse des défauts de paiement des prêts : Impact des caractéristiques des emprunteurs

Import Librairies

In [1]:
```
import pandas as pd
from pandasql import sqldf
```

Data Preparation

In [2]:
```
data = pd.read_csv("BA-ProjectData.csv",sep=',', header=0)
```

In [3]:
```
data
```

Out[3]:

	CaseNo	Default	CreditBalPerc	DebtPerc	LateUpto60Days	Late60to90Days	LateOve
0	2.0	0	0.957151	0.121876	0	0	
1	4.0	0	0.233810	0.036050	0	0	
2	5.0	0	0.907239	0.024926	1	0	
3	8.0	0	0.754464	0.209940	0	0	
4	10.0	0	0.189169	0.606291	0	0	
...	...	...	...	...	...	...	
75750	149974.0	0	1.026395	0.494819	0	0	
75751	149980.0	1	0.224711	0.057235	0	0	
75752	149982.0	0	0.810012	0.121752	0	0	
75753	149983.0	0	0.021046	0.250272	0	0	
75754	149986.0	0	0.954409	0.324962	0	0	

75755 rows × 15 columns

In [4]:
```
data.sample(5)
```

Out[4]:

	CaseNo	Default	CreditBalPerc	DebtPerc	LateUpto60Days	Late60to90Days	LateOve
44185	82887.0	0	0.247048	1.255310	0	0	
72481	142869.0	0	0.001857	1.373528	0	0	
2989	5594.0	0	0.000000	0.225320	0	0	
67517	132154.0	0	0.022485	0.284559	0	0	
21031	39231.0	0	0.028999	0.431009	0	0	

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 75755 entries, 0 to 75754
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   CaseNo          75755 non-null  float64
 1   Default         75755 non-null  int64  
 2   CreditBalPerc   75755 non-null  float64
 3   DebtPerc        75755 non-null  float64
 4   LateUpto60Days  75755 non-null  int64  
 5   Late60to90Days  75755 non-null  int64  
 6   LateOver90Days  75755 non-null  int64  
 7   Income          75755 non-null  float64
 8   NoOfOpenLoans   75755 non-null  int64  
 9   NoOfHomeLoans   75755 non-null  int64  
 10  Dependents      75755 non-null  int64  
 11  Age             75755 non-null  int64  
 12  Job             75755 non-null  object 
 13  Status          75755 non-null  object 
 14  Education       75755 non-null  object 
dtypes: float64(4), int64(8), object(3)
memory usage: 8.7+ MB
```

In [6]:
```
data.describe().round(2)
```

Out[6]:

	CaseNo	Default	CreditBalPerc	DebtPerc	LateUpto60Days	Late60to90Days	LateOv
count	75755.00	75755.00	75755.00	75755.00	75755.00	75755.00	
mean	72415.58	0.10	0.36	0.41	0.16	0.05	
std	42980.21	0.29	0.36	0.41	0.55	0.30	
min	2.00	0.00	0.00	0.00	0.00	0.00	
25%	35348.50	0.00	0.05	0.17	0.00	0.00	
50%	70921.00	0.00	0.21	0.31	0.00	0.00	
75%	108985.00	0.00	0.61	0.49	0.00	0.00	
max	149986.00	1.00	4.00	2.98	7.00	6.00	

In [7]:
```
data.duplicated().sum()
```

Out[7]:
```
0
```

In [8]:
```
data.nunique()
```

Out[8]:
```
CaseNo            75755
Default               2
CreditBalPerc     65456
DebtPerc          70823
LateUpto60Days        8
Late60to90Days        7
LateOver90Days        8
Income            11002
NoOfOpenLoans        18
NoOfHomeLoans         6
Dependents            6
Age                  50
Job                  10
Status                3
Education             4
dtype: int64
```

In [9]:
```
data[['Default', 'LateUpto60Days', 'Late60to90Days','LateOver90Days','NoOfHomeLoans',
```

Out[9]:
```
Default                                                [0, 1]
LateUpto60Days                    [0, 1, 7, 2, 3, 5, 6, 4]
Late60to90Days                       [0, 1, 2, 5, 3, 4, 6]
LateOver90Days                    [0, 1, 2, 4, 3, 7, 5, 6]
NoOfHomeLoans                           [0, 1, 4, 2, 3, 5]
Dependents                              [1, 0, 2, 3, 4, 5]
Status                        [married, single, divorced]
Education          [tertiary, secondary, unknown, primary]
dtype: object
```

In [10]:
```
#Ajout ratio entre le solde de crédit et le revenu
data['CreditBalance_Income_Ratio']= data['CreditBalPerc']/data['Income']
```

In [ ]:
```
```

In [11]:
```
#Ajout colonne Category Revenu (faible,moyen,elevé)
R1 = data['Income'].quantile(0.25)
R3 = data['Income'].quantile(0.75)

def cat_income(income):
    if (income<= R1):
        return 'Weak income'
    elif (income<=R3) and (income > R1):
        return 'middle income'
    else:
        return 'high income'

data['Cat_income']= data['Income'].apply(cat_income)
```

DATA ANALYSIS

In [18]:
```
#TOP Income Job
data.sort_values(by='Income', ascending=False).head(5)[['Age','Job', 'Income']]
```

Out[18]:

	Age	Job	Income
23221	59	admin.	287662.0
57888	44	technician	261666.0
12723	57	self-employed	261666.0
49599	50	blue-collar	261666.0
51010	52	blue-collar	251608.0

In [19]:
```
#Worst Income Job
data.sort_values(by='Income', ascending=True).head(5)[['Age','Job', 'Income']]
```

Out[19]:

	Age	Job	Income
54821	36	self-employed	500.0
66066	44	technician	500.0
6151	34	unemployed	500.0
29461	51	unemployed	500.0
66214	54	unemployed	500.0

In [26]:
```
#percentage of clients defaulted from the loan
percent_default=round((len(data[data['Default']==1])/len(data)*100),2)
print(percent_default)
```

```
9.54
```

In [27]:
```
#percentage of employed clients are loan defaulters?
data['Job'].unique()
```

Out[27]:
```
array(['technician', 'entrepreneur', 'blue-collar', 'management',
       'admin.', 'services', 'retired', 'self-employed', 'unemployed',
       'student'], dtype=object)
```

In [33]:
```
#percentage of employed clients are loan defaulters?
employed_client= data[(data["Job"]!='student') & (data["Job"]!='unemployed')]
percent_employ=round((len(employed_client[employed_client['Default']==1])/len(data)*1
print(percent_employ)
```

```
3.42
```

In [34]:
```
#percentage of unemployed clients are loan defaulters?
print(round(percent_default-percent_employ,2))
```

```
6.12
```

In [43]:
```
#percentage of married clients are loan defaulters?
married_client=data[data['Status']=='married']
percent_maried_default=round(len(married_client[married_client['Default']==1])/len(da
print(percent_maried_default)
```

```
5.71
```

In [48]:
```
#top 3 jobs according to employee counts
data["Job"].value_counts().sort_values(ascending=False).head(3)
```

Out[48]:
```
blue-collar    19660
management     13078
technician     11864
Name: Job, dtype: int64
```

In [70]:
```
#3 job types have the most loan defaulters
data_default=data[data['Default']==1]
top_job_default=data_default['Job'].value_counts().sort_values(ascending=False).head(
top_job_default
```

Out[70]:
```
unemployed       4605
self-employed     883
services          390
Name: Job, dtype: int64
```

In [71]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 75755 entries, 0 to 75754
Data columns (total 17 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   CaseNo                      75755 non-null  float64
 1   Default                     75755 non-null  int64
 2   CreditBalPerc               75755 non-null  float64
 3   DebtPerc                    75755 non-null  float64
 4   LateUpto60Days              75755 non-null  int64
 5   Late60to90Days              75755 non-null  int64
 6   LateOver90Days              75755 non-null  int64
 7   Income                      75755 non-null  float64
 8   NoOfOpenLoans               75755 non-null  int64
 9   NoOfHomeLoans               75755 non-null  int64
 10  Dependents                  75755 non-null  int64
 11  Age                         75755 non-null  int64
 12  Job                         75755 non-null  object
 13  Status                      75755 non-null  object
 14  Education                   75755 non-null  object
 15  CreditBalance_Income_Ratio  75755 non-null  float64
 16  Cat_income                  75755 non-null  object
dtypes: float64(5), int64(8), object(4)
memory usage: 9.8+ MB
```

Sauvegarder les modifications

In [72]:
```
data.to_csv('data_modified.csv', index=False)
```

Dashboard : Loan Default Analysis

Number of case

75,76K

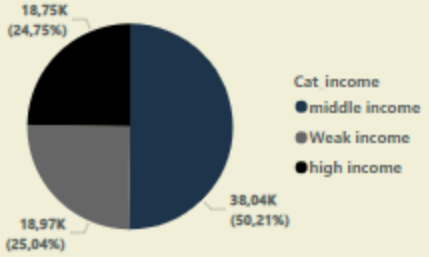
Filtre by Age

21 70

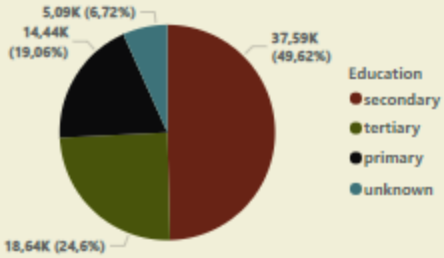
Filtre by Default

0 1

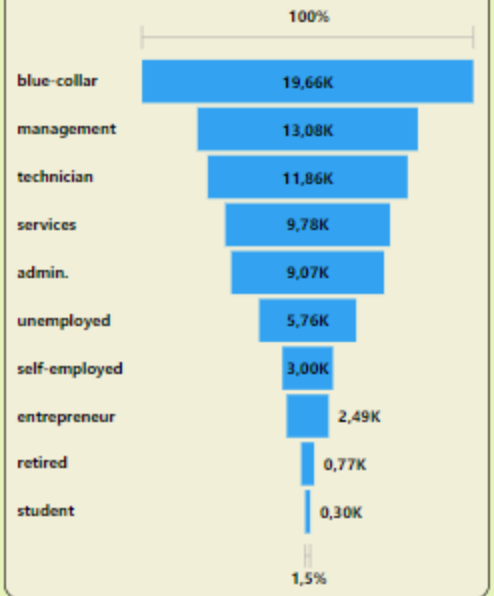
Income Category



Education



Count of Job



Count of Status