BANK LOAN CASE STUDY

FINAL PROJECT - 2

2022-2023

PARITOSH GANDRE

paritoshkrcg@gmail.com

PROJECT DESCRIPTION

The project I am assigned as my final project number 2 is BANK LOAN CASE STUDY. When we talk about bank loan, the very first thing that comes to our mind is the rate of Interest or for any other student a question arises which is How do these banks make profit by lending money to people.

To answer this question detailed study and analysis is carried out by every bank. In this analysis they find out which person is trustable have higher chances of paying back the loan with the interest signed at the time of contract.

Same question will be answered by me by the end of the analysis of the project.

APPROACH

So starting off with the project, my very first step will be to analyze and make sure that I understand the whole dataset and tables provided by the team.

Then I will see if there are any missing values, noisy data present in the dataset and will get rid of it. Then I'll be able to carry out further processing of data with my queries to each and every question.

TECH-STACK USED

I have used JupyterLab for python, and MS Excel

INSIGHTS

First of all, the dataset in itself was humongous. Today I got to know how big of a data can be and what these data analysts do every day to solve or come up with a solution for their organization.

There are many charts present in the report, reflecting the results achieved from queries. More of the insights will be mentioned in the respective results section.

RESULT

APPLICATION DATASET

- DATA PRE-PROCESSING:
  Here we are going to delete null values or null columns or rows from the dataset.
- So the very first step is to find columns with more than 50% of null values.

- By using : df.columns[df.isnull().mean()>0.50] we will get to know the colum names with more than 50% of null values:

```
[8]: df.columns[df.isnull().mean()>0.50]

[8]: Index(['OWN_CAR_AGE', 'EXT_SOURCE_1', 'APARTMENTS_AVG', 'BASEMENTAREA_AVG',
            'YEARS_BUILD_AVG', 'COMMONAREA_AVG', 'ELEVATORS_AVG', 'ENTRANCES_AVG',
            'FLOORSMIN_AVG', 'LANDAREA_AVG', 'LIVINGAPARTMENTS_AVG',
            'LIVINGAREA_AVG', 'NONLIVINGAPARTMENTS_AVG', 'NONLIVINGAREA_AVG',
            'APARTMENTS_MODE', 'BASEMENTAREA_MODE', 'YEARS_BUILD_MODE',
            'COMMONAREA_MODE', 'ELEVATORS_MODE', 'ENTRANCES_MODE', 'FLOORSMIN_MODE',
            'LANDAREA_MODE', 'LIVINGAPARTMENTS_MODE', 'LIVINGAREA_MODE',
            'NONLIVINGAPARTMENTS_MODE', 'NONLIVINGAREA_MODE', 'APARTMENTS_MEDI',
            'BASEMENTAREA_MEDI', 'YEARS_BUILD_MEDI', 'COMMONAREA_MEDI',
            'ELEVATORS_MEDI', 'ENTRANCES_MEDI', 'FLOORSMIN_MEDI', 'LANDAREA_MEDI',
            'LIVINGAPARTMENTS_MEDI', 'LIVINGAREA_MEDI', 'NONLIVINGAPARTMENTS_MEDI',
            'NONLIVINGAREA_MEDI', 'FONDKAPREMONT_MODE', 'HOUSETYPE_MODE',
            'WALLSMATERIAL_MODE'],
           dtype='object')
```

- Here we can see which columns need to be dropped from the dataset so that they don't cause any further inconsistency in the result.
- By using : df.drop(df.columns[df.isnull().mean()>0.50],axis=1) we can drop the columns mentioned above.

```
[12]: df.drop(df.columns[df.isnull().mean()>0.50],axis=1)
```

[12]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUIT
0	100002	1	Cash loans	M	N	Y	0	202500.0	406597.5	24700
1	100003	0	Cash loans	F	N	N	0	270000.0	1293502.5	35698
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	135000.0	6750
3	100006	0	Cash loans	F	N	Y	0	135000.0	312682.5	29686
4	100007	0	Cash loans	M	N	Y	0	121500.0	513000.0	21865
...	...	...	...	...	...	...	...	...	...	
307506	456251	0	Cash loans	M	N	N	0	157500.0	254700.0	27558
307507	456252	0	Cash loans	F	N	Y	0	72000.0	269550.0	12001
307508	456253	0	Cash loans	F	N	Y	0	153000.0	677664.0	29979
307509	456254	1	Cash loans	F	N	Y	0	171000.0	370107.0	20205
307510	456255	0	Cash loans	F	N	N	0	157500.0	675000.0	49117

307511 rows × 81 columns

- So earlier the number of columns were 122 and now it has become 81, gives us the proof that tables have been deleted.
- It also happens that some of columns can be irrelevant to our analysis, so we have to drop them as well.
- The list of unwanted columns is: *Manually Selected*

  'FLAG_MOBIL','FLAG_EMP_PHONE','FLAG_WORK_PHONE','FLAG_PHONE','FLAG_CONT_PHONE','FLAG_EMAIL','CNT_FAM_MEMBERS', 'REGION_RATING_CLIENT','REGION_RATING_CLIENT_W_CITY','EXT_SOURCE_3','YEAR_BEGINEXPLUATATION_AVG', 'YEAR_BEGINEXPLUATATION_MODE','YEAR_BEGINEXPLUATATION_MEDI','TOTALAREA_MODE','EMERGENCYSTATE_MODE', 'DAYS_LAST_PHONE_CHANGE','FLAG_DOCUMENT_2','FLAG_DOCUMENT_3','FLAG_DOCUMENT_4','FLAG_DOCUMENT_5', 'FLAG_DOCUMENT_6','FLAG_DOCUMENT_7','FLAG_DOCUMENT_8','FLAG_DOCUMENT_9','FLAG_DOCUMENT_10','FLAG_DOCUMENT_11', 'FLAG_DOCUMENT_12','FLAG_DOCUMENT_13','FLAG_DOCUMENT_14','FLAG_DOCUMENT_15','FLAG_DOCUMENT_16', 'FLAG_DOCUMENT_17','FLAG_DOCUMENT_18','FLAG_DOCUMENT_19','FLAG_DOCUMENT_20','FLAG_DOCUMENT_21'

[23]:

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AMT_INCOME_TOTAL	AMT_CREDIT	AMT_ANNUITY	...	OBS_30_CNT_SOCIAL_
0	100002	1	Cash loans	M	N	Y	0	202500.0	406597.5	24700.5	...	
1	100003	0	Cash loans	F	N	N	0	270000.0	1293502.5	35698.5	...	
2	100004	0	Revolving loans	M	Y	Y	0	67500.0	135000.0	6750.0	...	
3	100006	0	Cash loans	F	N	Y	0	135000.0	312682.5	29686.5	...	
4	100007	0	Cash loans	M	N	Y	0	121500.0	513000.0	21865.5	...	
...	...	...	...	...	...	...	...	...	...	...	...	
307506	456251	0	Cash loans	M	N	N	0	157500.0	254700.0	27558.0	...	
307507	456252	0	Cash loans	F	N	Y	0	72000.0	269550.0	12001.5	...	
307508	456253	0	Cash loans	F	N	Y	0	153000.0	677664.0	29979.0	...	
307509	456254	1	Cash loans	F	N	Y	0	171000.0	370107.0	20205.0	...	
307510	456255	0	Cash loans	F	N	N	0	157500.0	675000.0	49117.5	...	

307511 rows × 45 columns

- The column number has dropped down to 45 which indicates that the query was successful.
- Further part is to replace blanks within the columns with either mean or mode or median depending on the column.
- First column here is OCCUPATION_TYPE:

```
7]: df['OCCUPATION_TYPE'].isnull().sum()

7]: 96391
```

```
]: pd.value_counts(df['OCCUPATION_TYPE'])

]: Laborers                 55186
   Sales staff              32102
   Core staff               27570
   Managers                 21371
   Drivers                  18603
   High skill tech staff    11380
   Accountants               9813
   Medicine staff            8537
   Security staff            6721
   Cooking staff             5946
   Cleaning staff            4653
   Private service staff     2652
   Low-skill Laborers        2093
   Waiters/barmen staff      1348
   Secretaries               1305
   Realty agents              751
   HR staff                   563
   IT staff                   526
   Name: OCCUPATION_TYPE, dtype: int64
```

- Fill null values with mode in OCCUPATION_TYPE column because it is a categorical column:

```
[59]: df_relevant_columns['OCCUPATION_TYPE'].fillna('Laborers', inplace=True)

[61]: df_relevant_columns['OCCUPATION_TYPE'].value_counts()

[61]: Laborers                 151577
      Sales staff               32102
      Core staff                27570
      Managers                  21371
      Drivers                   18603
      High skill tech staff     11380
      Accountants                9813
      Medicine staff             8537
      Security staff             6721
      Cooking staff              5946
      Cleaning staff             4653
      Private service staff      2652
      Low-skill Laborers         2093
      Waiters/barmen staff       1348
      Secretaries                1305
      Realty agents               751
      HR staff                    563
      IT staff                    526
      Name: OCCUPATION_TYPE, dtype: int64

[40]: df_dropped_columns['OCCUPATION_TYPE'].isnull().sum()

[40]: 0
```

- As you can see, I have filled the null values with MODE **"Laborers"** and then recalculated the count of each value where null values are 0.

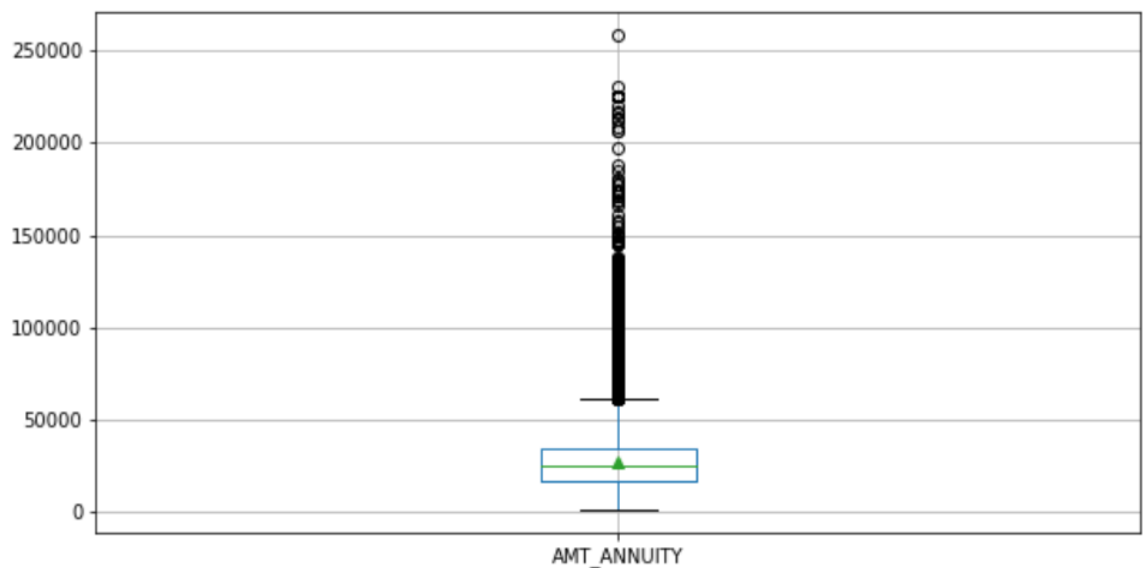
FINDING OUTLIERS

- Next column is AMT_ANNUITY, in this column we have to find outliers first, because it is a continuous value column so there is a high chance that it may contain outliers.

- To find outliers I have plotted the values with box-whisker plot which gives us accurate representation of outliers in the table.
- Then I have filled the null values with median value "**24903.0**".

```
[75]: df_relevant_columns.boxplot(column=["AMT_ANNUITY"],figsize=(10,5))
```

```
[75]: <matplotlib.axes._subplots.AxesSubplot at 0x2c48f138d60>
```




```
[76]: df_relevant_columns["AMT_ANNUITY"].median()
```

```
[76]: 24903.0
```

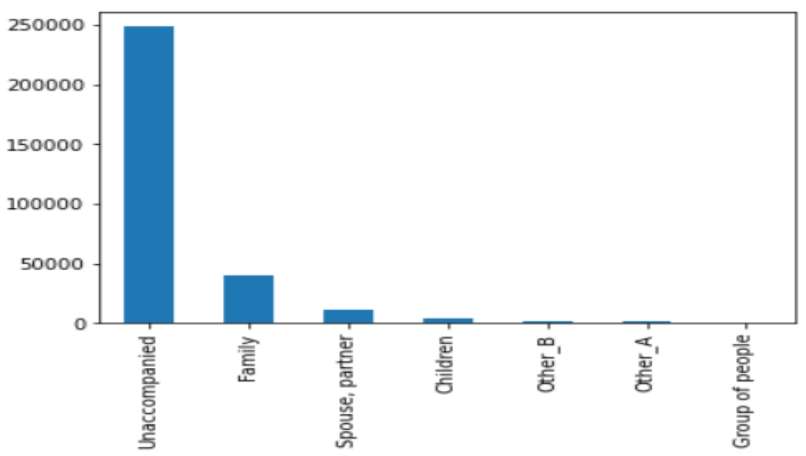
```
7]: df_relevant_columns["AMT_ANNUITY"].fillna(df_relevant_columns["AMT_ANNUITY"].median(),inplace = True)
```

```
8]: df_relevant_columns["AMT_ANNUITY"].isnull().sum()
```

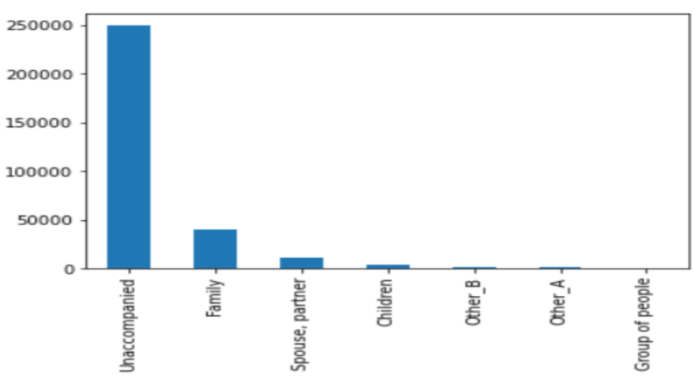
```
8]: 0
```

- Replacing blanks in NAME_TYPE_SUITE column with mode because it is a categorical column: Mode is “Unaccompanied”

```
[35]: df_relevant_columns['NAME_TYPE_SUITE'].mode()

[35]: 0    Unaccompanied
      dtype: object

[38]: df_relevant_columns['NAME_TYPE_SUITE'].value_counts().plot.bar()

[38]: <matplotlib.axes._subplots.AxesSubplot at 0x1ad412a85e0>
```




```
[36]: df_relevant_columns['NAME_TYPE_SUITE'].isnull().sum()

[36]: 1292

[39]: df_relevant_columns['NAME_TYPE_SUITE'].fillna("Unaccompanied",inplace = True)

[40]: df_relevant_columns['NAME_TYPE_SUITE'].value_counts().plot.bar()

[40]: <matplotlib.axes._subplots.AxesSubplot at 0x1ad413dcd30>
```



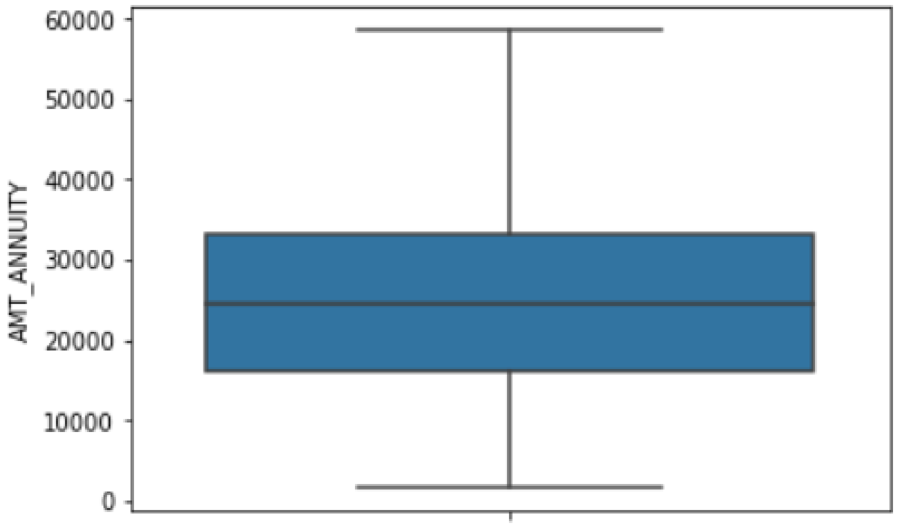

```
[42]: df_relevant_columns['NAME_TYPE_SUITE'].value_counts()

[42]: Unaccompanied      249818
      Family              40149
      Spouse, partner     11370
      Children             3267
      Other_B              1770
      Other_A               866
      Group of people       271
      Name: NAME_TYPE_SUITE, dtype: int64
```

- Here we finish the process of filling null values with median and mode successfully.
- Moving on to the next step which is getting rid of outliers because they impact the overall column calculation and since they are outliers we do not need to count them for some columns.
- First column is AMT_ANNUITY:
- Outliers are mostly present beyond the Q3 + 1.5 * IQR range. (Q3 = 75 percentile, IQR = Inter Quantile Range)
- Since I have already visualized the distribution of AMT_ANNUITY column earlier with boxplot, I will simply give a condition that select data points below a number only which are not outliers. In this way I will only select the normalized data.

```
[106]: df_amt_out = df_relevant_columns[df_relevant_columns['AMT_ANNUITY']<58698]
```

```
[107]: sns.boxplot(y = 'AMT_ANNUITY', data = df_amt_out)
```

```
[107]: <matplotlib.axes._subplots.AxesSubplot at 0x1ad42831fa0>
```



- For further outlier removal, I did not find a column where I can remove or replace outliers because most of the columns have real-life data and in real-life it may happen that these outliers are legitimate for a user, so removing these outliers can only cause disrupt in our analysis: such as AMT_INCOME_TOTAL, AMT_CREDIT, DAYS_BIRTH , DAYS_EMPLOYED, etc..

ANALYSIS

- The very next step after removing the outliers is Analysis of columns where we will visualize the distribution of data by percentage or real values and see what percentage a value holds in that column.
- Our first column for analysis is **Target Variable** which shows almost 92% of total clients had no problem in paying off the loan but remaining 8% had.

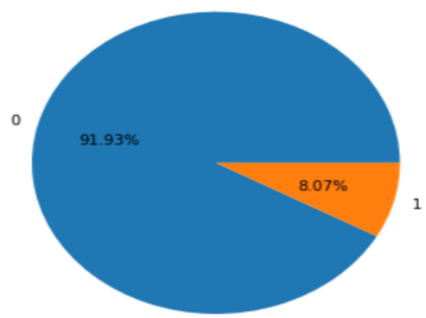
137]:
```
new_df = df_relevant_columns['TARGET'].value_counts().rename_axis("categories").reset_index(name="counts")
new_df
```

137]:

	categories	counts
0	0	282686
1	1	24825

138]:
```
xTargetLabel = new_df.categories
yTargetLabel = new_df.counts
```

139]:
```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.axis('equal')

ax.pie(yTargetLabel,labels =xTargetLabel,autopct='%1.2f%%')
plt.show()
```



- Here 0 means client has no issues with paying back the loan and 1 means the contrary.

- Next column for analysis is CODE_GENDER, which contains values like F: Female, M: Male , and XNA.

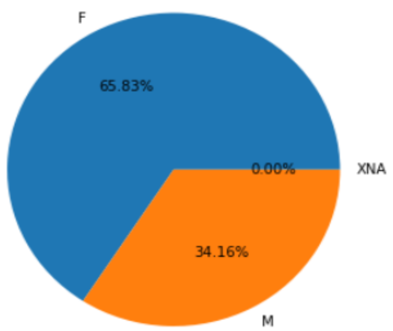
```
[140]: new_df_gender = df_relevant_columns['CODE_GENDER'].value_counts().rename_axis("Gender").reset_index(name="counts")
new_df_gender
```

[140]:

	Gender	counts
0	F	202448
1	M	105059
2	XNA	4

```
[141]: xGenderLabel = new_df_gender.Gender
yGenderLabel = new_df_gender.counts
```

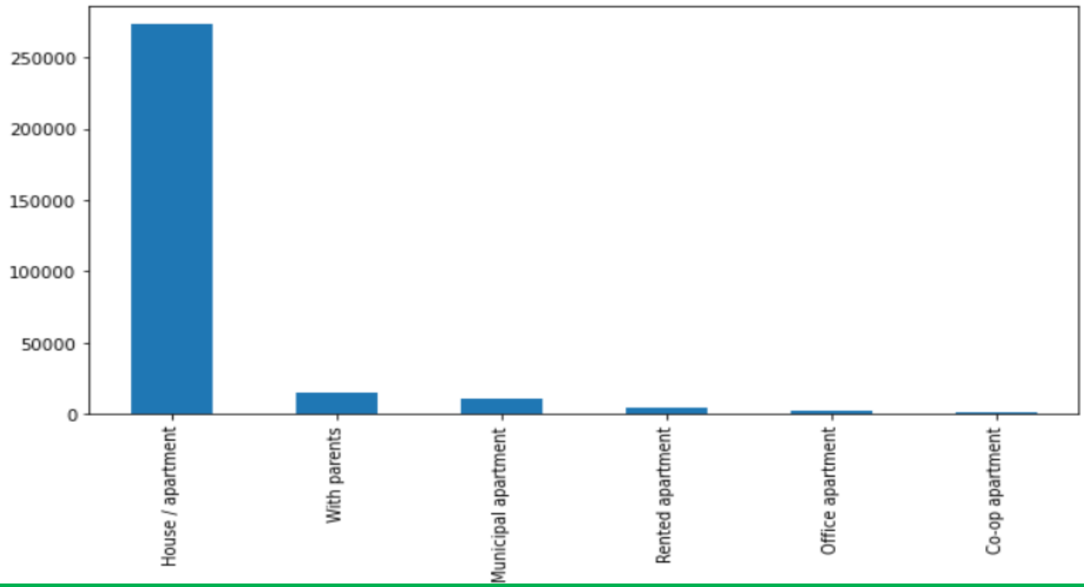
```
[154]: fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.axis('equal')

ax.pie(yGenderLabel,labels =xGenderLabel,autopct='%1.2f%%')
plt.show()
```



- What I am doing is converting the values into data frame to feed them to the pie chart in proper syntax.
- Here we can see that 66% of client are women and 34% of men and almost negligible percentage of XNA's.

- Next column I have chosen is NAME_HOUSING_TYPE.
- This column tells us about the housing situation of the client if they are living in their own apartment, with parents, office apartment, etc.

```
[159]: df_relevant_columns['NAME_HOUSING_TYPE'].value_counts()

[159]: House / apartment      272868
       With parents            14840
       Municipal apartment     11183
       Rented apartment         4881
       Office apartment         2617
       Co-op apartment          1122
       Name: NAME_HOUSING_TYPE, dtype: int64

[157]: df_relevant_columns['NAME_HOUSING_TYPE'].value_counts().plot.bar(figsize=(10,5))

[157]: <matplotlib.axes._subplots.AxesSubplot at 0x1ad42ce0eb0>
```



- Then I have calculated the percentage of these values in the column.

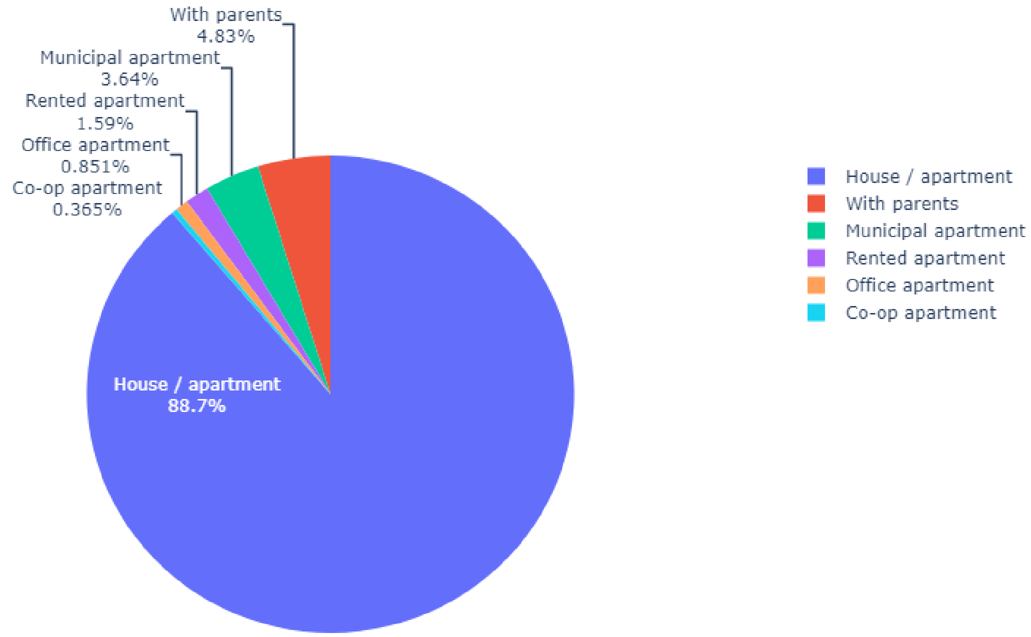
```
In [95]: new_perc_nmt = perc_NMT.rename_axis("types").reset_index(name="counts")
         new_perc_nmt
```

Out[95]:

	types	counts
0	House / apartment	88.734387
1	With parents	4.825844
2	Municipal apartment	3.636618
3	Rented apartment	1.587260
4	Office apartment	0.851026
5	Co-op apartment	0.364865

```
In [96]: xPercNMT = new_perc_nmt.types
         yPercNMT = new_perc_nmt.counts
```

```
import plotly.graph_objects as go

labels = xPercNMT
values =yPercNMT

fig = go.Figure(data=[go.Pie(labels=labels, values=values, textinfo='label+percent',
                             insidetextorientation='radial'
                            )])
fig.show()
```

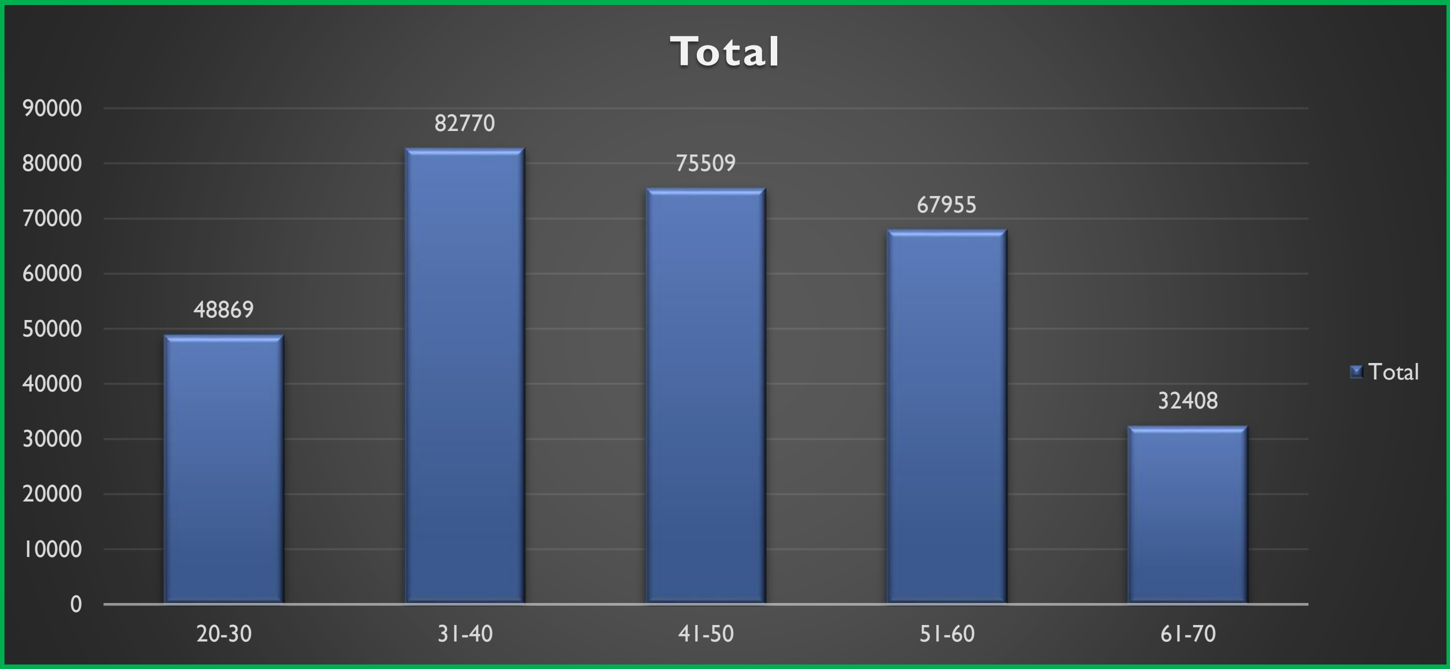


- With this data the bank can target those people who don't live in their own house, because there can be chances that they may fancy living in their own house.
- Just by selecting appropriate clients the bank can benefit itself.

UNIVARIATE ANALYSIS

- Univariate analysis is done on only one variable.
- I have selected age group i.e. DAYS_BIRTH column which tells us the client's age in days. (e.g. If I am 2 years old then in days I will be -730 days from today)
- First of all I created a column named YEARS_BIRTH in which I have given a formula (DAYS_BIRTH/-365), to get positive value I divided values by -365 because the values are negative.
- Then I created another column for year class interval with If conditioning: e.g. if YEARS_BIRTH >20 and YEARS_BIRTH<30 = class interval is 20-30 likewise for each interval.

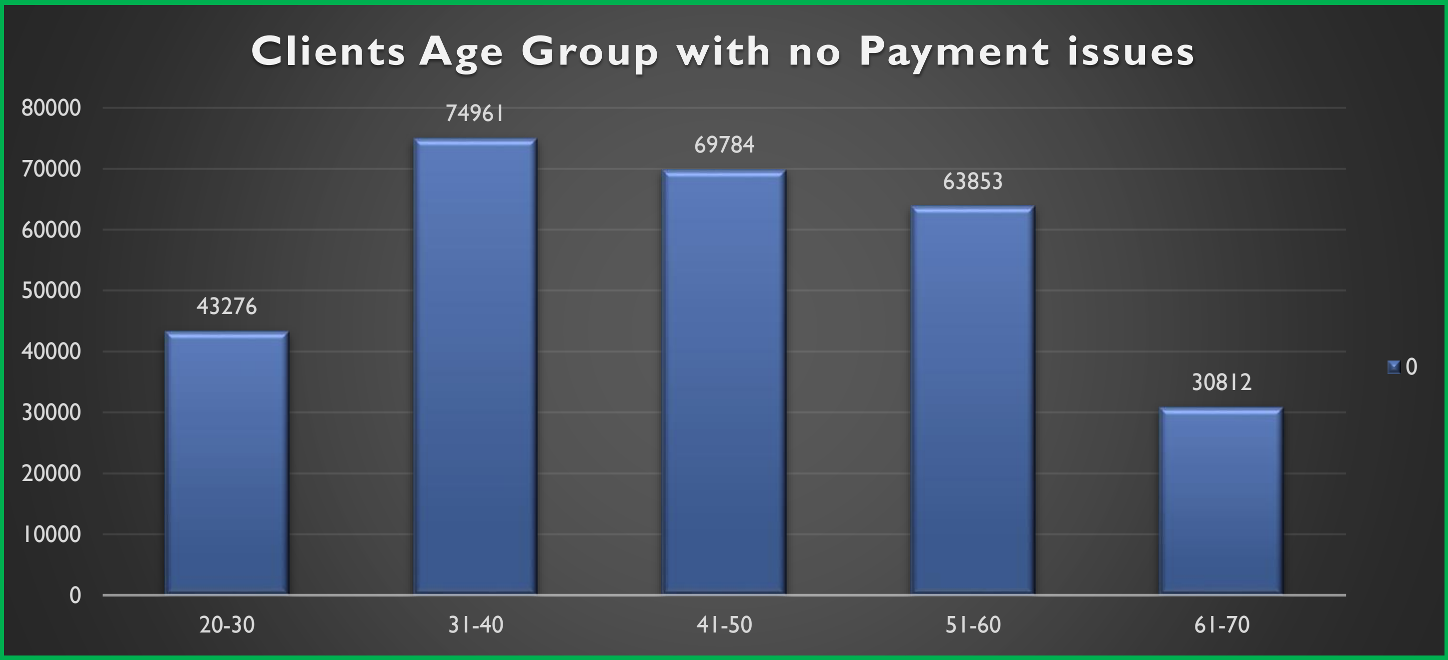
Row Labels	Count of YEARS_BIRTH_RANGE
20-30	48869
31-40	82770
41-50	75509
51-60	67955
61-70	32408
Grand Total	**307511**



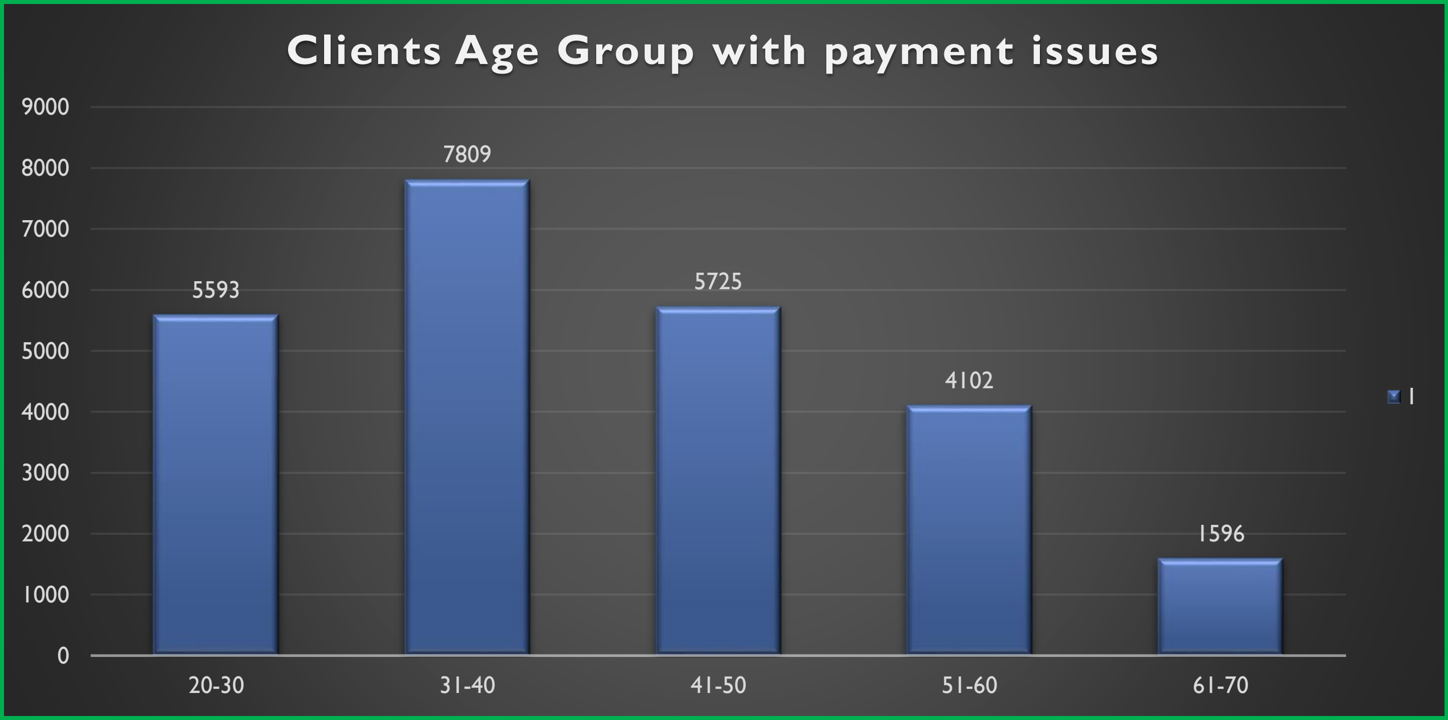
- Here we also have to find out how many clients have payment issues and who don't.

Count of TARGET	Column Labels	
Row Labels	**0**	**Grand Total**
20-30	43276	43276
31-40	74961	74961
41-50	69784	69784
51-60	63853	63853
61-70	30812	30812
Grand Total	**282686**	**282686**

- From this we can see most of the client belong to 31-40 age range and at the highest when repaying the loan to banks.
- Also it is the same range where clients have issues repaying the loan back to the bank.



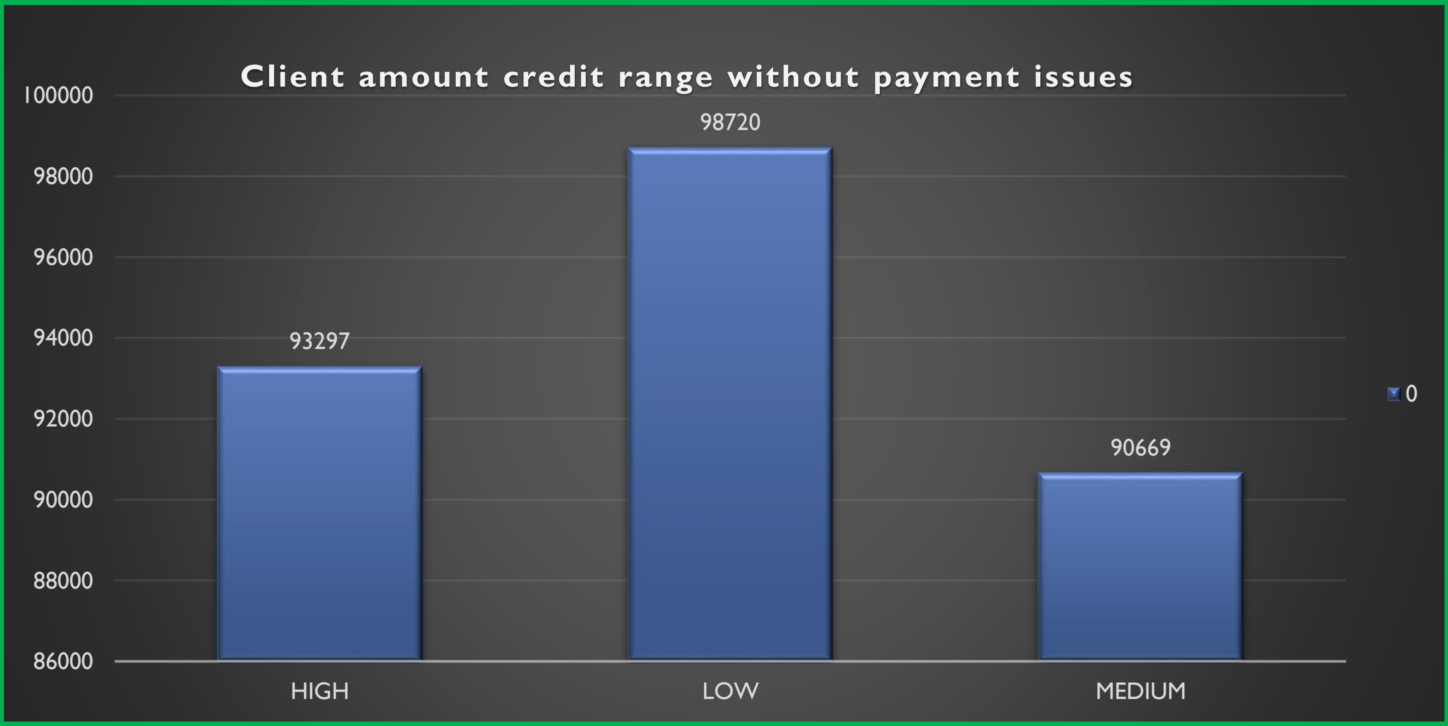
Count of TARGET	Column Labels	
Row Labels	1	Grand Total
20-30	5593	5593
31-40	7809	7809
41-50	5725	5725
51-60	4102	4102
61-70	1596	1596
Grand Total	**24825**	**24825**



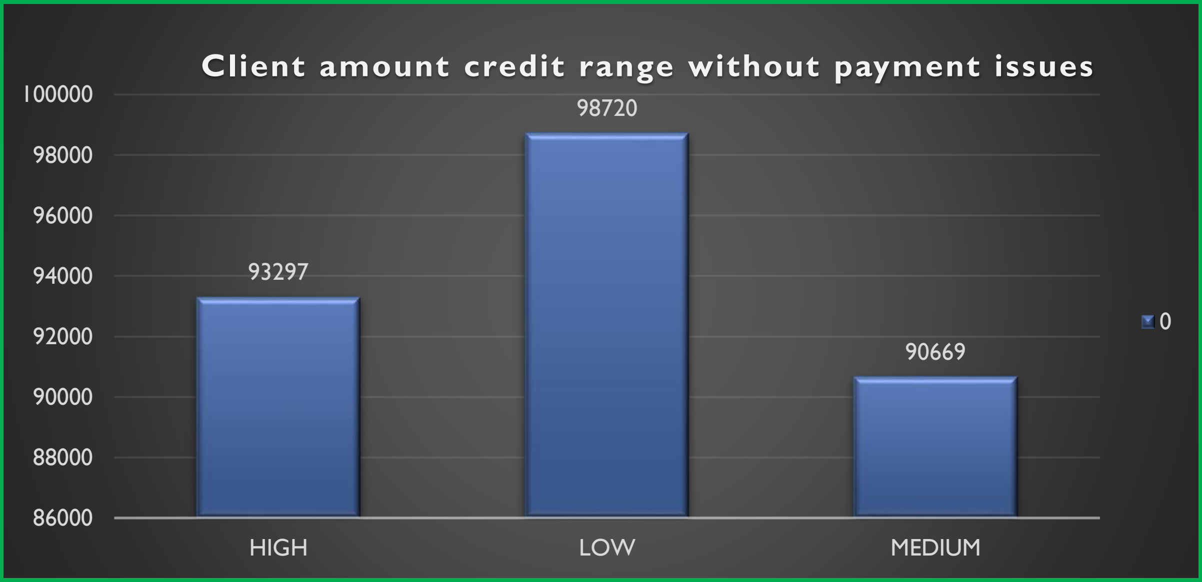
- Next column for analysis is AMT_CLIENT_CREDIT.

Count of TARGET	Column Labels	
Row Labels	**0**	**Grand Total**
HIGH	93297	93297
LOW	98720	98720
MEDIUM	90669	90669
Grand Total	**282686**	**282686**

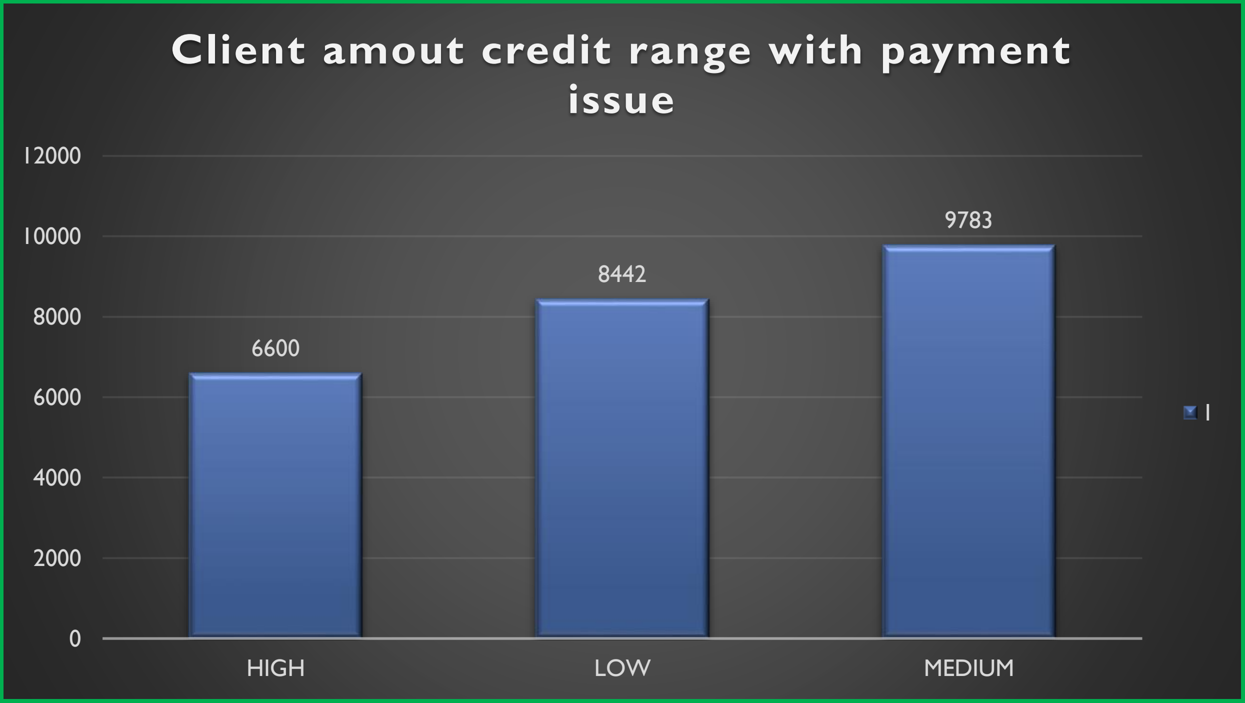
- Here we can see that clients belonging to the LOW category have the highest count when it comes to clients repaying the loans back to the banks.



- And with the MEDIUM category clients having the highest count of clients not repaying the loans back to the banks.

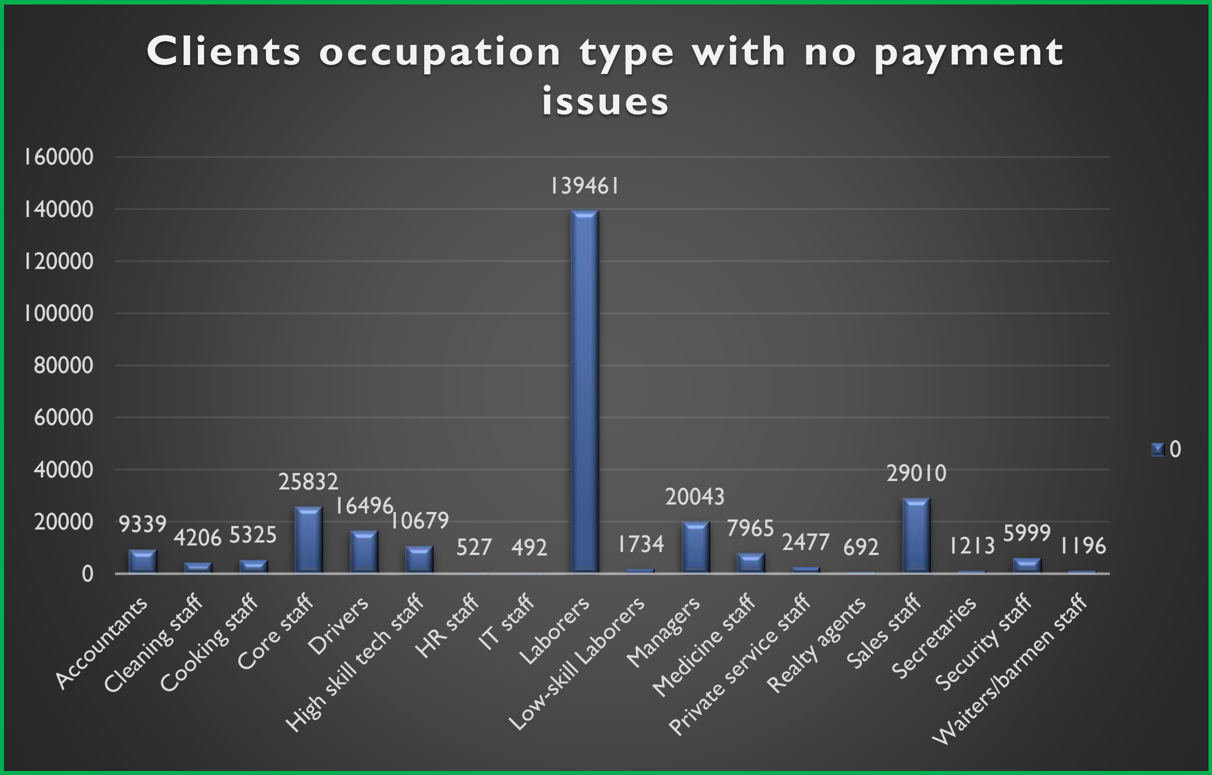


Count of TARGET	Column Labels	
Row Labels	1	Grand Total
HIGH	6600	6600
LOW	8442	8442
MEDIUM	9783	9783
Grand Total	**24825**	**24825**



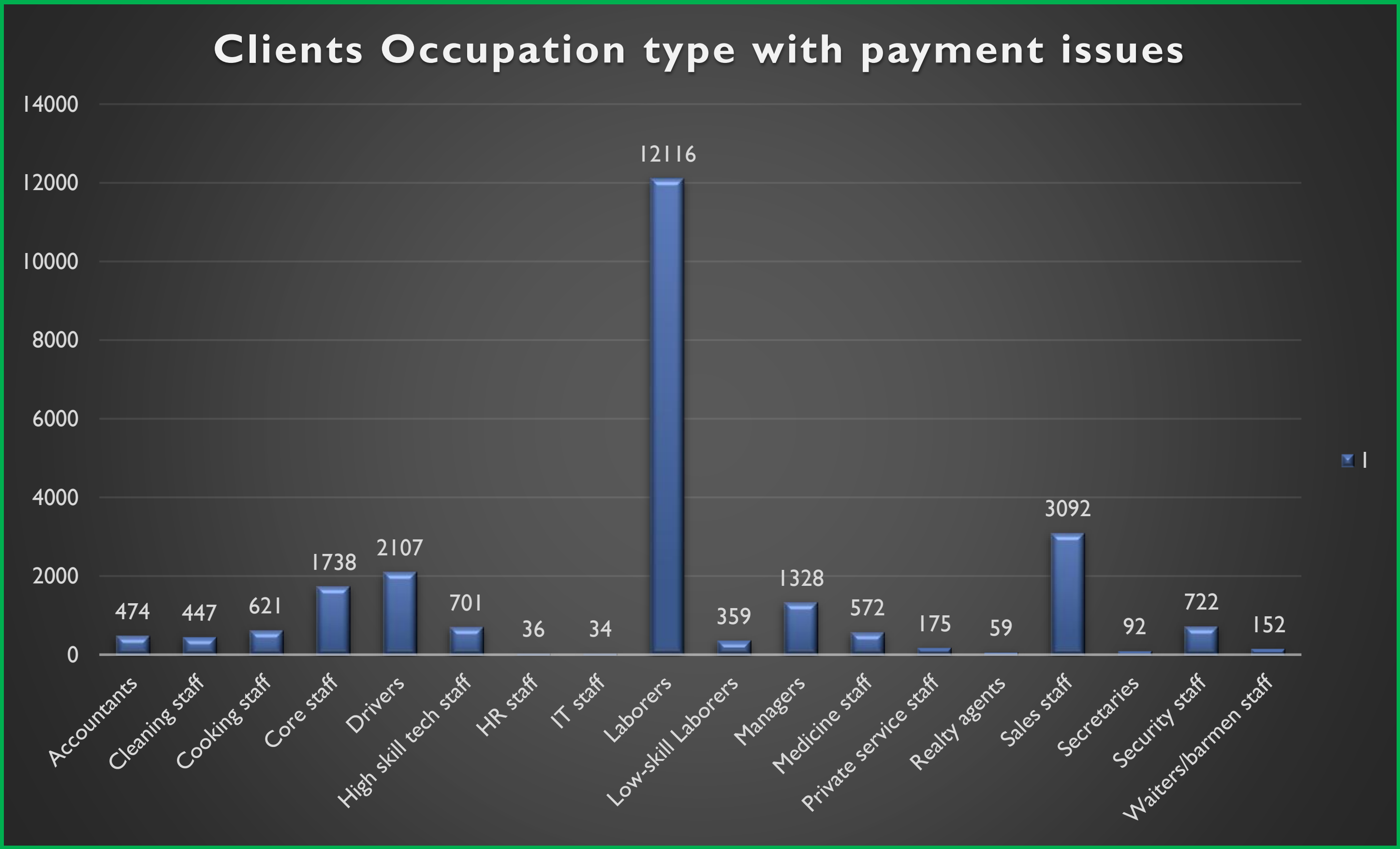
- Similarly I have counted number of clients and the range they belong to for the required columns from which the bank can take informed decisions on whom to give the loan.
- OCCUPATION_TYPE:

Count of TARGET	Column Labels	
Row Labels	0	Grand Total
Accountants	9339	9339
Cleaning staff	4206	4206
Cooking staff	5325	5325
Core staff	25832	25832
Drivers	16496	16496
High skill tech staff	10679	10679
HR staff	527	527
IT staff	492	492
Laborers	139461	139461
Low-skill Laborers	1734	1734
Managers	20043	20043
Medicine staff	7965	7965
Private service staff	2477	2477
Realty agents	692	692
Sales staff	29010	29010
Secretaries	1213	1213
Security staff	5999	5999
Waiters/barmen staff	1196	1196
Grand Total	**282686**	**282686**



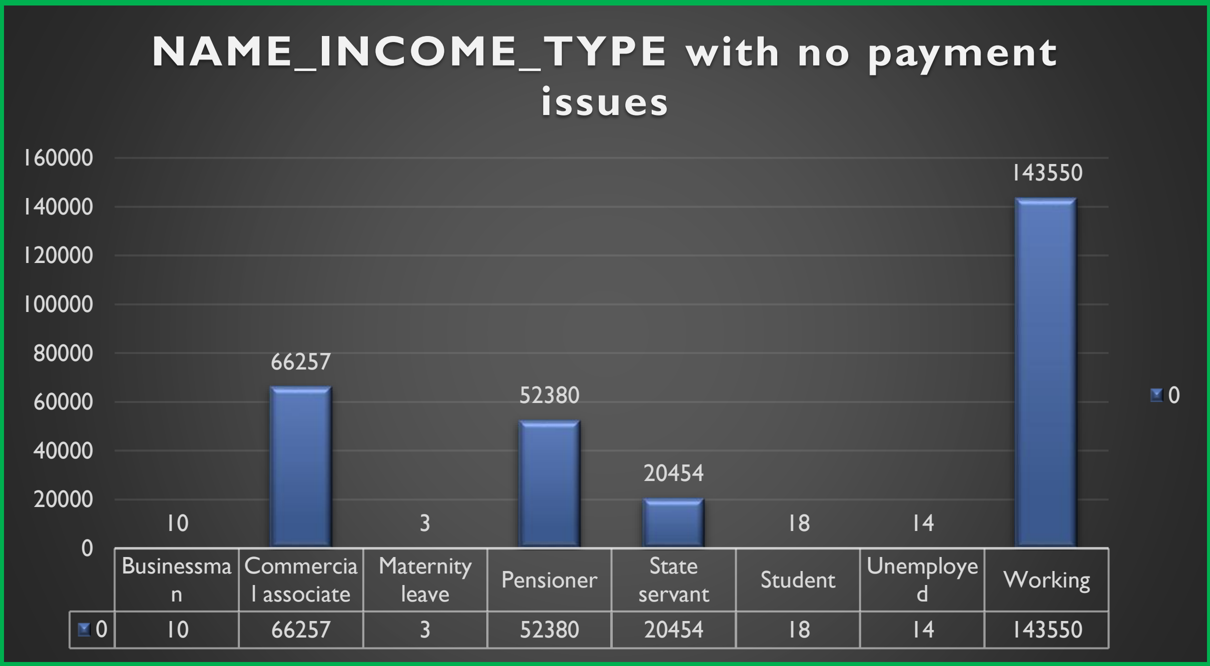
- Here “LABORERS” being the one with highest count of clients repaying the loan and highest count of clients facing issue in repaying the loan back to the bank.

Count of TARGET	Column Labels	
Row Labels	1	Grand Total
Accountants	474	474
Cleaning staff	447	447
Cooking staff	621	621
Core staff	1738	1738
Drivers	2107	2107
High skill tech staff	701	701
HR staff	36	36
IT staff	34	34
Laborers	12116	12116
Low-skill Laborers	359	359
Managers	1328	1328
Medicine staff	572	572
Private service staff	175	175
Realty agents	59	59
Sales staff	3092	3092
Secretaries	92	92
Security staff	722	722
Waiters/barmen staff	152	152
Grand Total	**24825**	**24825**



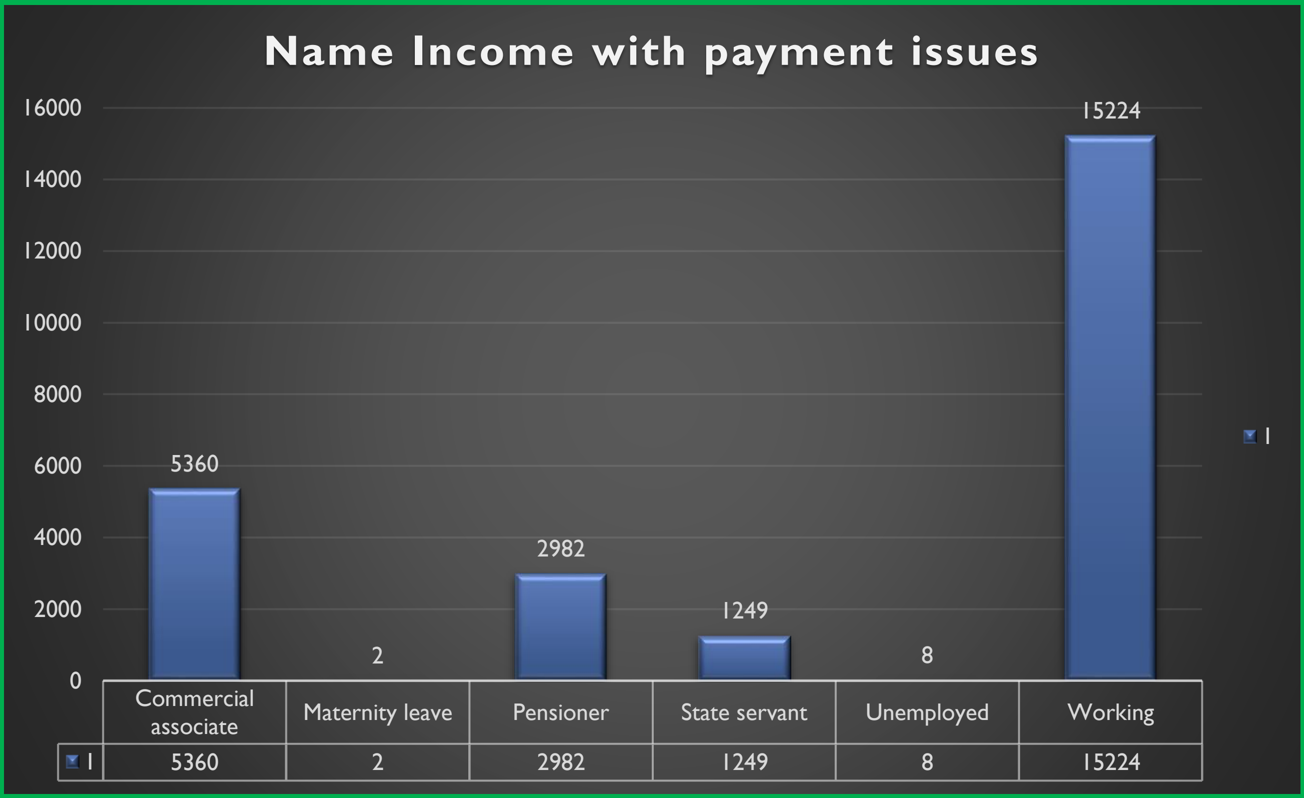
- NAME_INCOME_TYPE:

Count of TARGET	Column Labels	
Row Labels	0	Grand Total
Businessman	10	10
Commercial associate	66257	66257
Maternity leave	3	3
Pensioner	52380	52380
State servant	20454	20454
Student	18	18
Unemployed	14	14
Working	143550	143550
Grand Total	**282686**	**282686**



- Here we can see the "WORKING" category has the highest amount of clients repaying the loan and with this it is also the same category with highest count of clients for not paying the loan back to the bank.

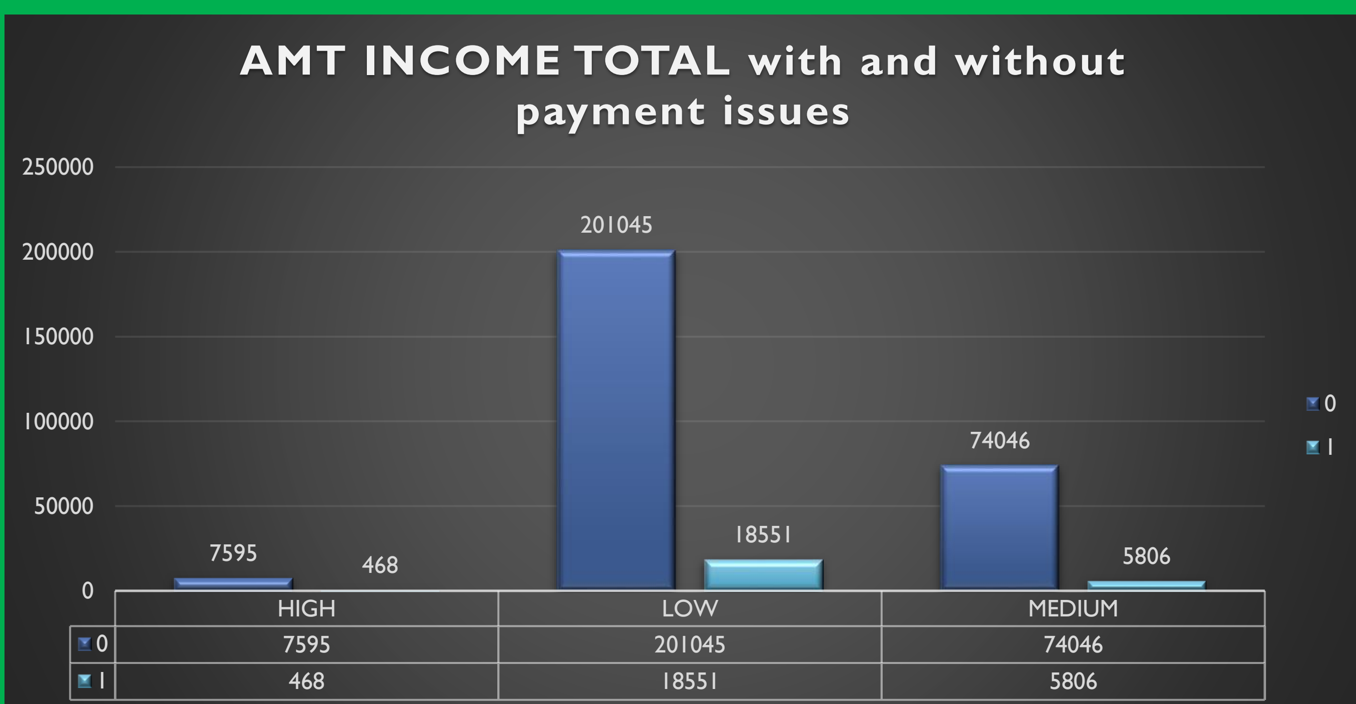
Count of TARGET	Column Labels	
Row Labels	1	Grand Total
Commercial associate	5360	5360
Maternity leave	2	2
Pensioner	2982	2982
State servant	1249	1249
Unemployed	8	8
Working	15224	15224
Grand Total	**24825**	**24825**



- AMT_TOTAL_INCOME:

Count of TARGET	Column Labels		
Row Labels	0	1	Grand Total
HIGH	7595	468	8063
LOW	201045	18551	219596
MEDIUM	74046	5806	79852
Grand Total	282686	24825	307511

- Here we can see "LOW" category has the highest count of clients when it comes to repaying the loan back to the bank and "LOW" category has the highest count of clients not paying the loan back to the bank.



- CNT_FAM_MEMBERS:

Count of TARGET	Column Labels		
Row Labels	**0**	**1**	**Grand Total**
1	62172	5675	67847
2	146350	12009	158359
3	47993	4608	52601
4	22561	2136	24697
5	3151	327	3478
6	353	55	408
7	75	6	81
8	14	6	20
9	6		6
10	2	1	3
11		1	1
12	2		2
13		1	1
14	2		2
15	1		1
16	2		2
20	2		2
Grand Total	**282686**	**24825**	**307511**

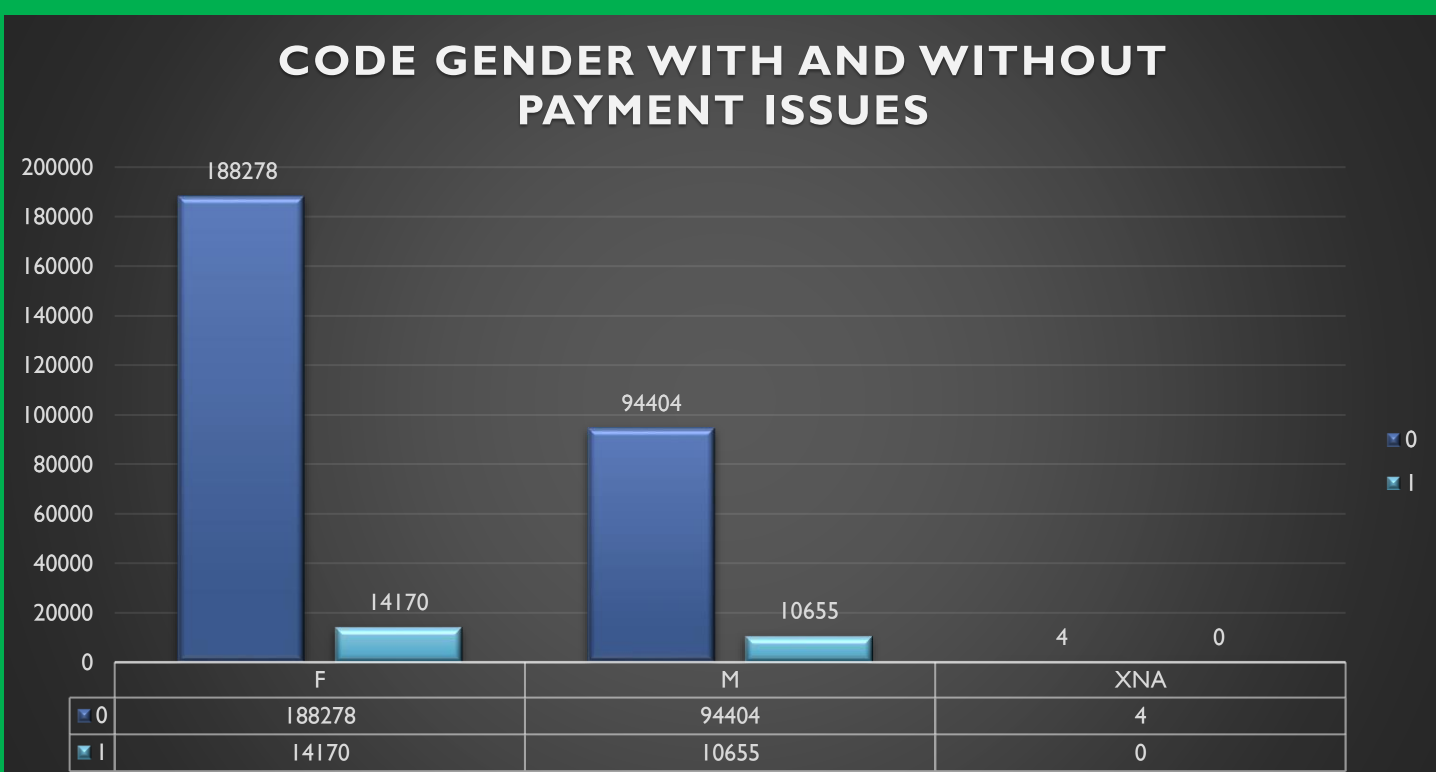
- Here you can see that clients with count of family members as 2 have higher chances of repaying the loan back to the bank compared to other counts of family members.
- And it is the same category which has the highest count of clients not paying the loan back to the bank which is clients with count of family members as 2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	20
0	62172	146350	47993	22561	3151	353	75	14	6	2		2		2	1	2	2
1	5675	12009	4608	2136	327	55	6	6		1	1		1				

- CODE_GENDER:

Count of TARGET	Column Labels		
Row Labels	0	1	Grand Total
F	188278	14170	202448
M	94404	10655	105059
XNA	4		4
Grand Total	282686	24825	307511

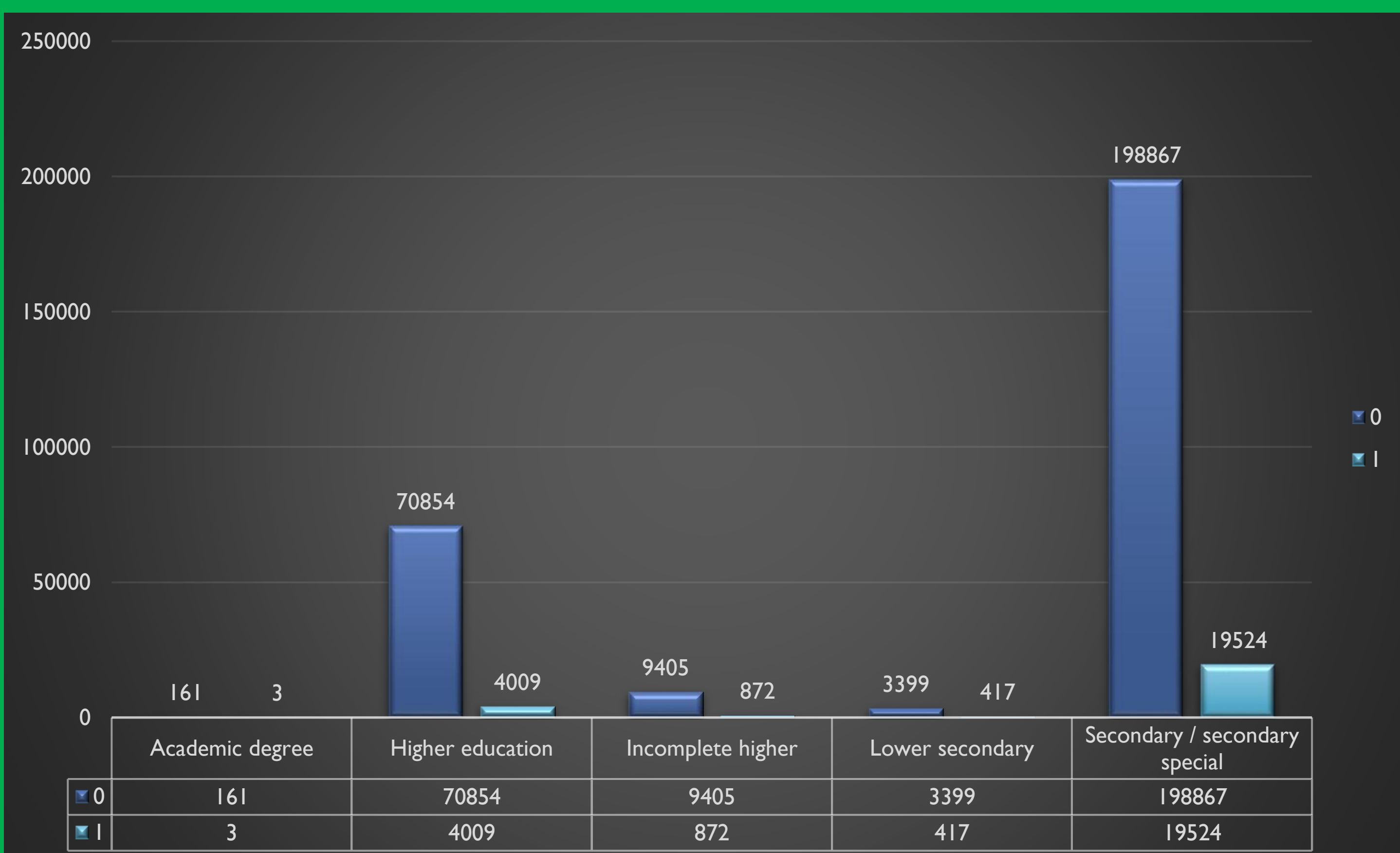
- ‘F’ category which stands for Females have the highest count of clients paying back the loan. Which is 188,278 out of 202,448 clients.
- For the clients not paying the loan back to the bank, ‘F’ category turns out to have the highest count of clients not paying the loan.
-



- NAME_EDUCATION_TYPE:

Count of TARGET	Column Labels		
Row Labels	0	1	Grand Total
Academic degree	161	3	164
Higher education	70854	4009	74863
Incomplete higher	9405	872	10277
Lower secondary	3399	417	3816
Secondary / secondary special	198867	19524	218391
Grand Total	**282686**	**24825**	**307511**

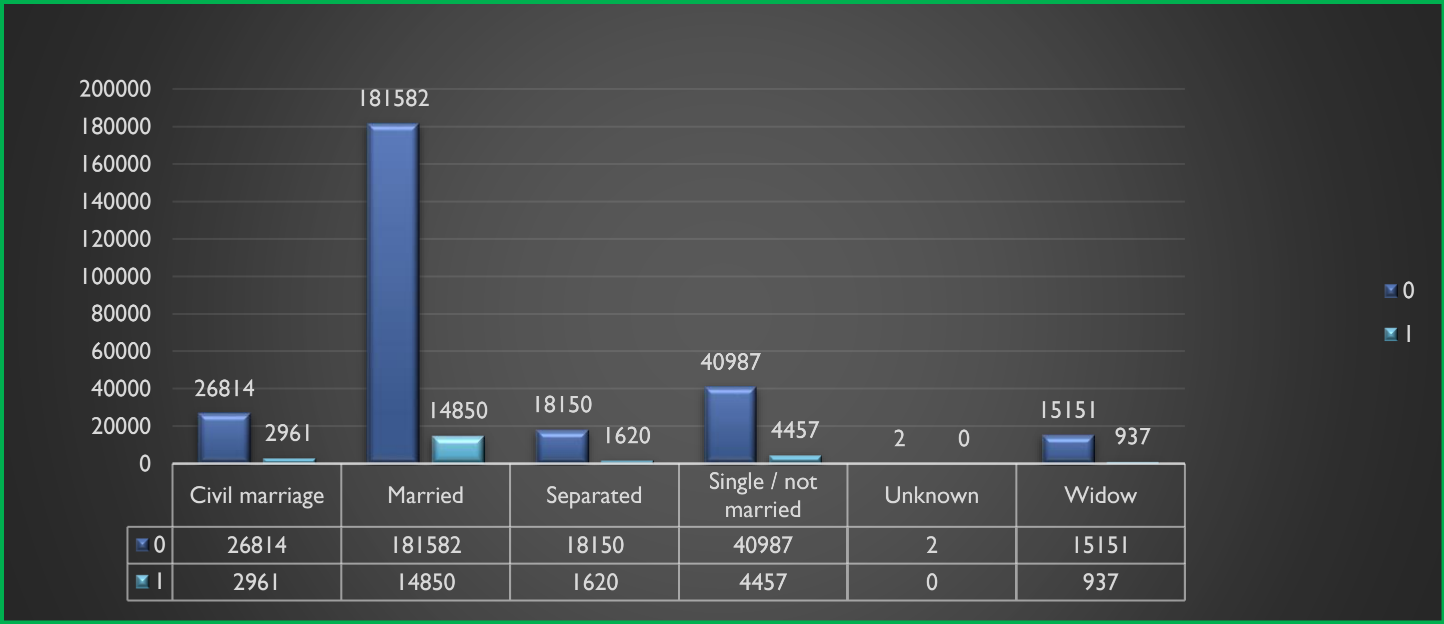
- Here we can infer that "Secondary/secondary special" category has the highest count of clients repaying the loans.
- With that it also becomes the one with highest count of clients not repaying their loans.



- NAME_FAMILY_STATUS

Count of TARGET	Column Labels		
Row Labels	0	1	Grand Total
Civil marriage	26814	2961	29775
Married	181582	14850	196432
Separated	18150	1620	19770
Single / not married	40987	4457	45444
Unknown	2		2
Widow	15151	937	16088
Grand Total	**282686**	**24825**	**307511**

- Here, "MARRIED" category has the highest count of clients repaying the loans.

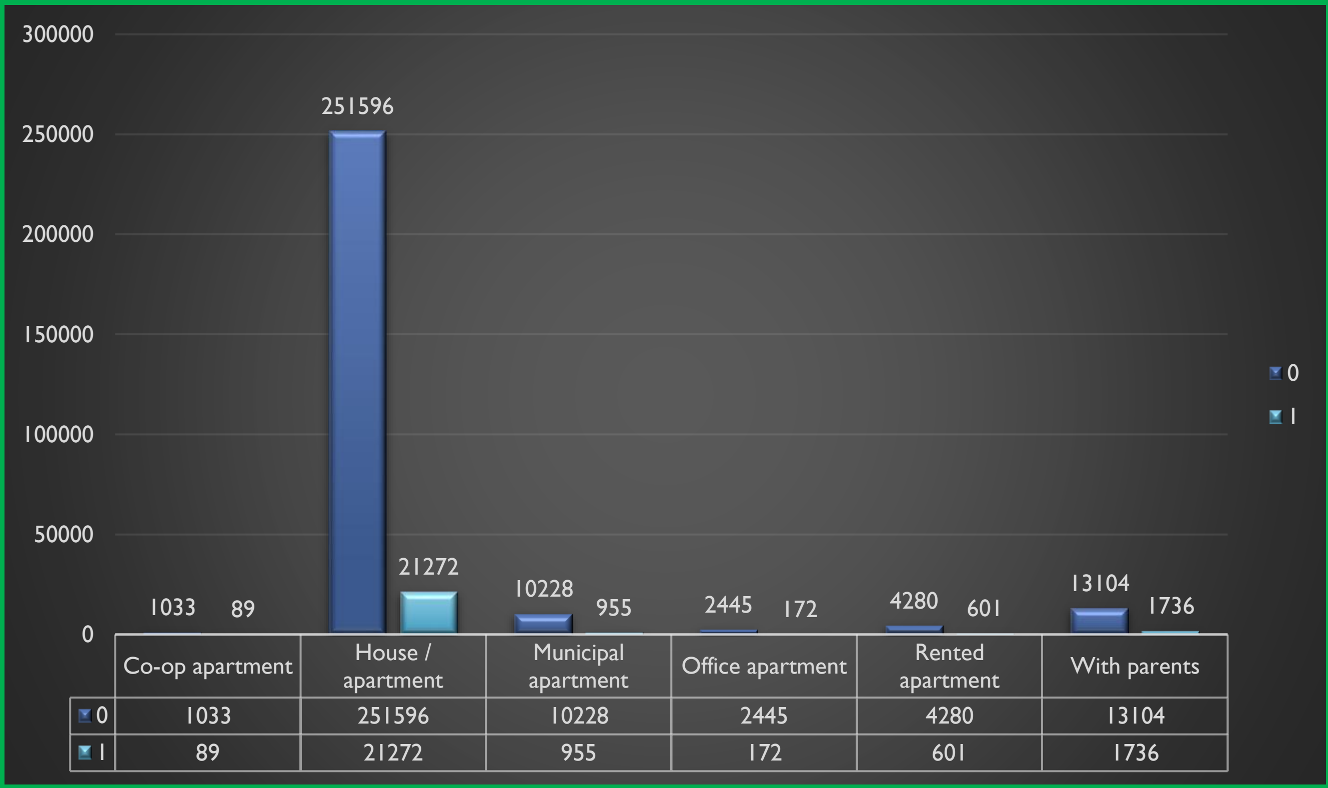


- For the non-payers category, "MARRIED" category tops the count of clients not paying the loan.

- NAME_HOUSING_TYPE:

Count of TARGET	Column Labels		
Row Labels	0	1	Grand Total
Co-op apartment	1033	89	1122
House / apartment	251596	21272	272868
Municipal apartment	10228	955	11183
Office apartment	2445	172	2617
Rented apartment	4280	601	4881
With parents	13104	1736	14840
Grand Total	**282686**	**24825**	**307511**

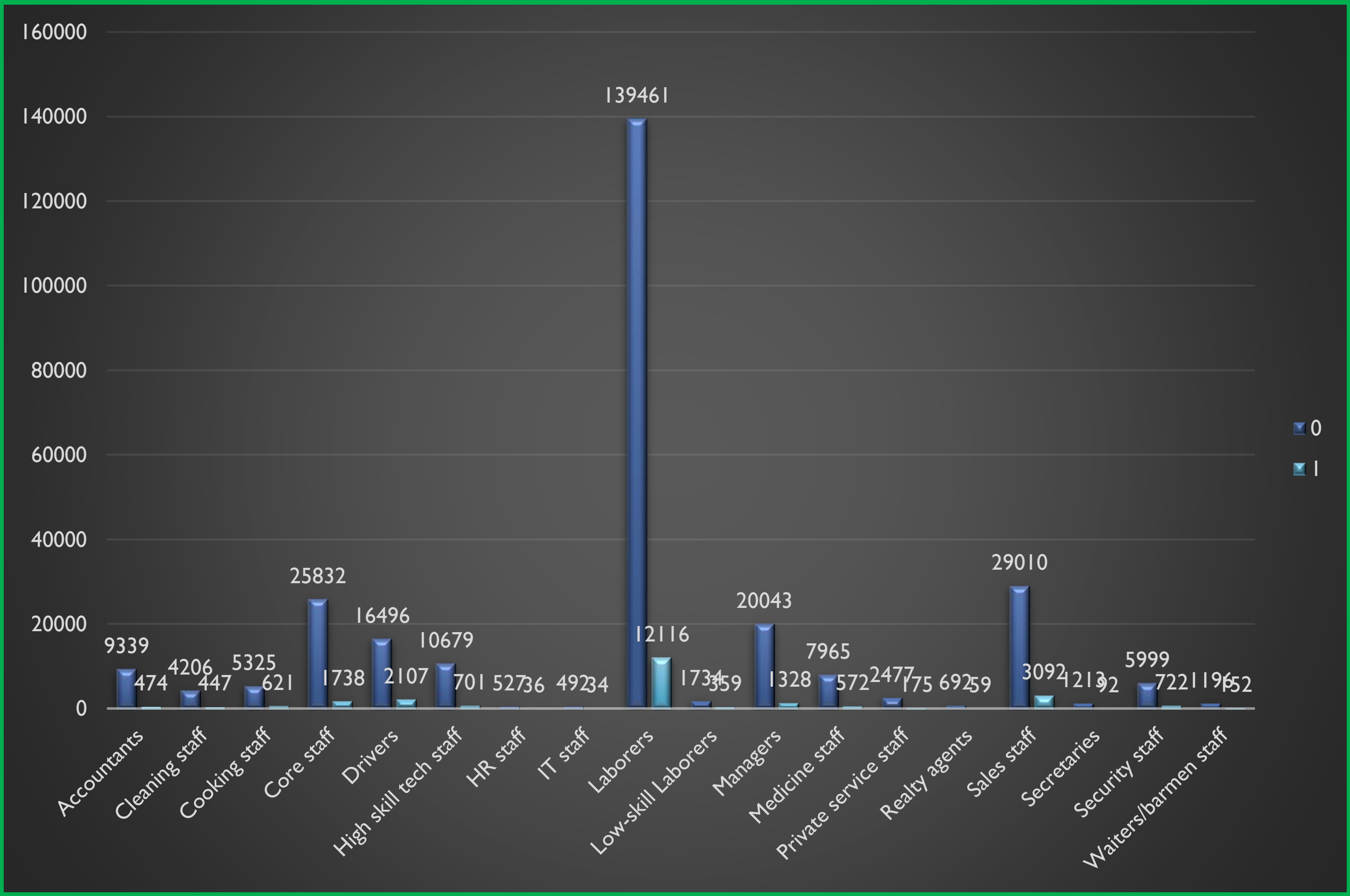
- For this column, the highest count of clients repaying their loans is the "HOUSE/APARTMENT" category.
- And for the highest count of clients not paying their loan also it is the "HOUSE/APARTMENT" category.



- OCCUPATION_TYPE:

Count of TARGET	Column Labels		
Row Labels	0	1	Grand Total
Accountants	9339	474	9813
Cleaning staff	4206	447	4653
Cooking staff	5325	621	5946
Core staff	25832	1738	27570
Drivers	16496	2107	18603
High skill tech staff	10679	701	11380
HR staff	527	36	563
IT staff	492	34	526
Laborers	139461	12116	151577
Low-skill Laborers	1734	359	2093
Managers	20043	1328	21371
Medicine staff	7965	572	8537
Private service staff	2477	175	2652
Realty agents	692	59	751
Sales staff	29010	3092	32102
Secretaries	1213	92	1305
Security staff	5999	722	6721
Waiters/barmen staff	1196	152	1348
Grand Total	**282686**	**24825**	**307511**

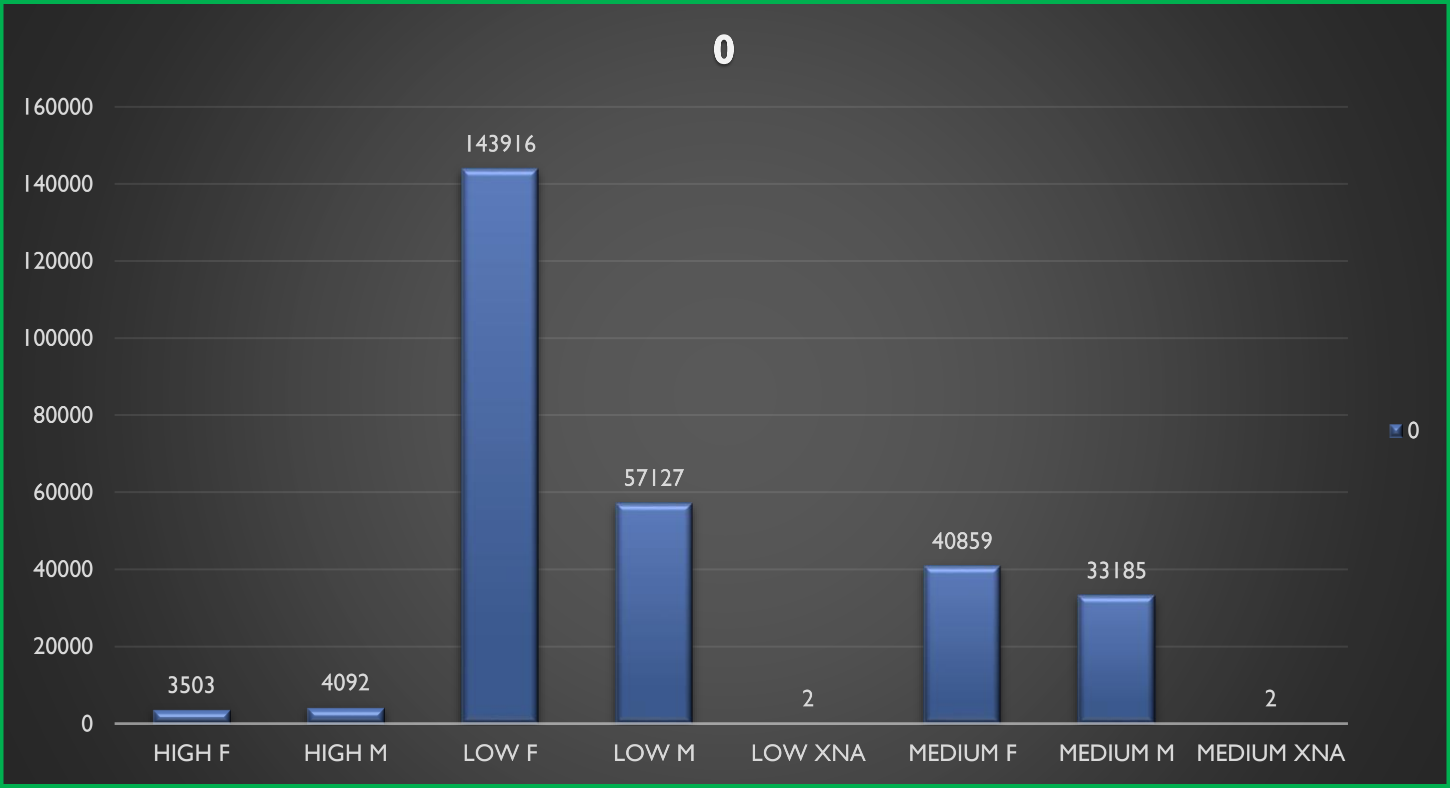
- Here we can see 'Laborers' category dominating both the columns of clients paying and not paying of loans.
- Laborers have 139,461 clients paying their loans on time and 12,116 clients not paying their loans.
- This will help the bank to decide which occupation type is more likely to repay their loans.

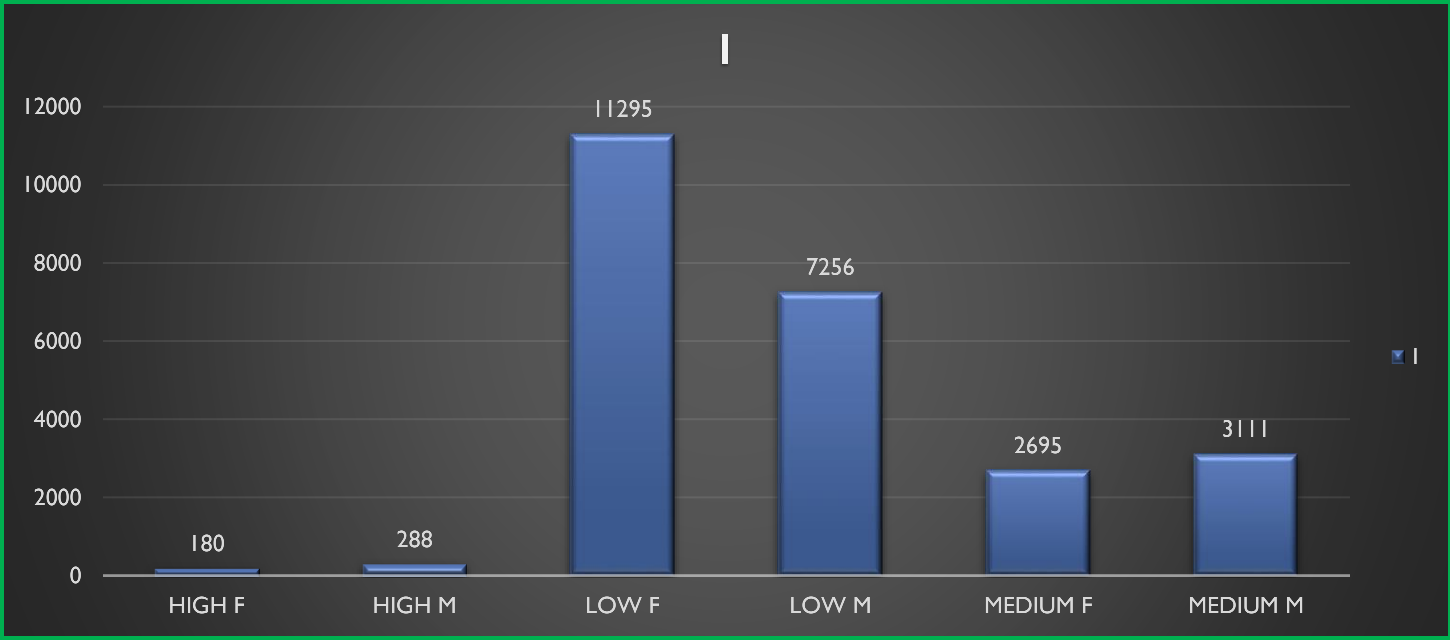


BIVARIATE ANALYSIS

- Bivariate analysis means selecting up to two variables and analyzing them from business point of view.
- I have done different bivariate analysis by taking sets of 2 variables and analyzing them together to find correlation between them.
- First two columns are TOTAL_INCOME_RANGE VS CODE_GENDER
- The process is almost the same compared to univariate analysis which is using pivot table in excel determining the distribution and correlation between any two variables or columns.

Count of CODE_GENDER	Column Labels		
Row Labels	0	1	Grand Total
HIGH	**7595**	**468**	**8063**
F	3503	180	3683
M	4092	288	4380
LOW	**201045**	**18551**	**219596**
F	143916	11295	155211
M	57127	7256	64383
XNA	2		2
MEDIUM	**74046**	**5806**	**79852**
F	40859	2695	43554
M	33185	3111	36296
XNA	2		2
Grand Total	**282686**	**24825**	**307511**

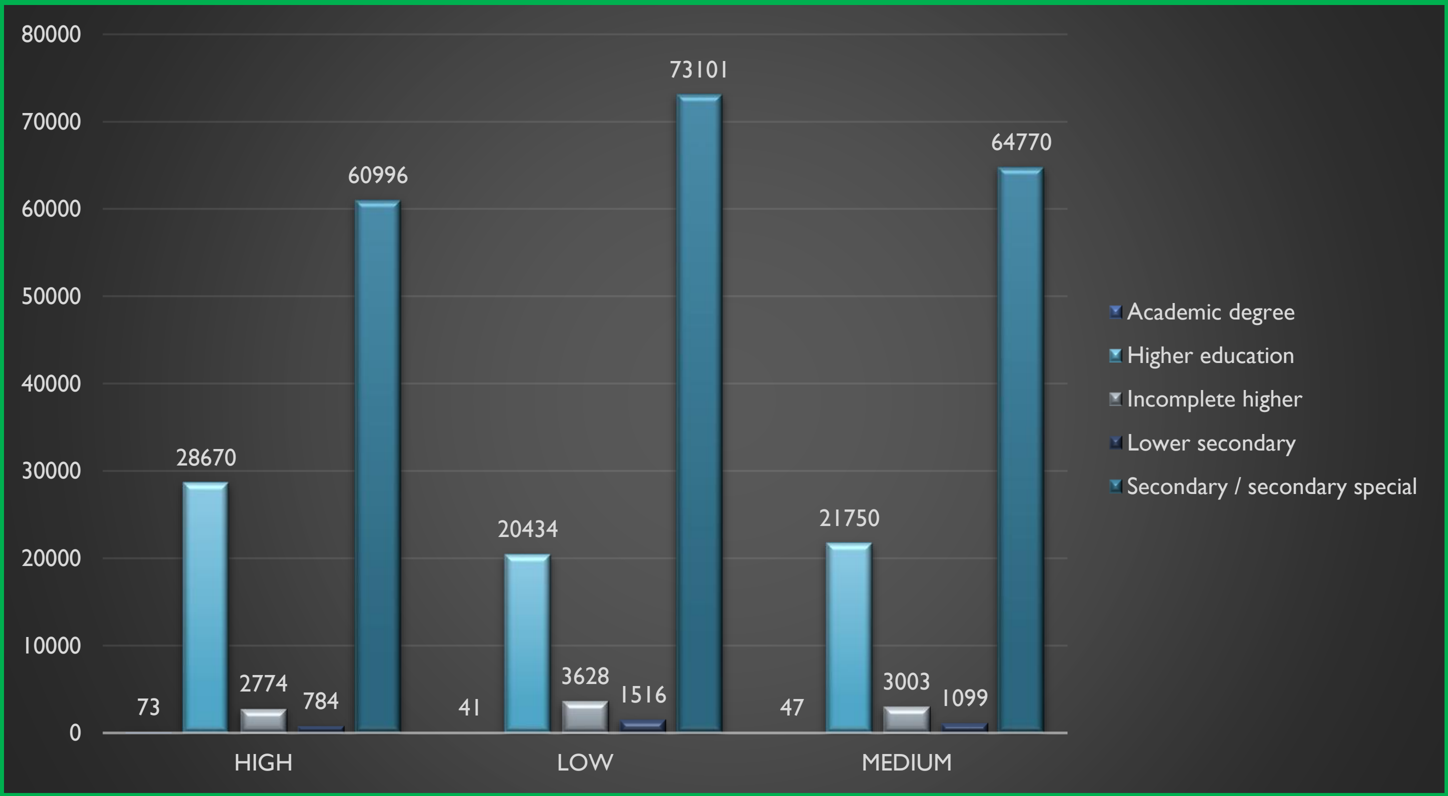




- From the charts above we can infer that women belonging to Low income group have highest number of clients with no payment issues and also with payment issues simultaneously.

- CREDIT AMOUNT VS EDUCATION STATUS:

TARGET	0					
Count of NAME_EDUCATION_TYPE	Column Labels					
Row Labels	Academic degree	Higher education	Incomplete higher	Lower secondary	Secondary / secondary special	Grand Total
HIGH	73	28670	2774	784	60996	93297
LOW	41	20434	3628	1516	73101	98720
MEDIUM	47	21750	3003	1099	64770	90669
Grand Total	**161**	**70854**	**9405**	**3399**	**198867**	**282686**

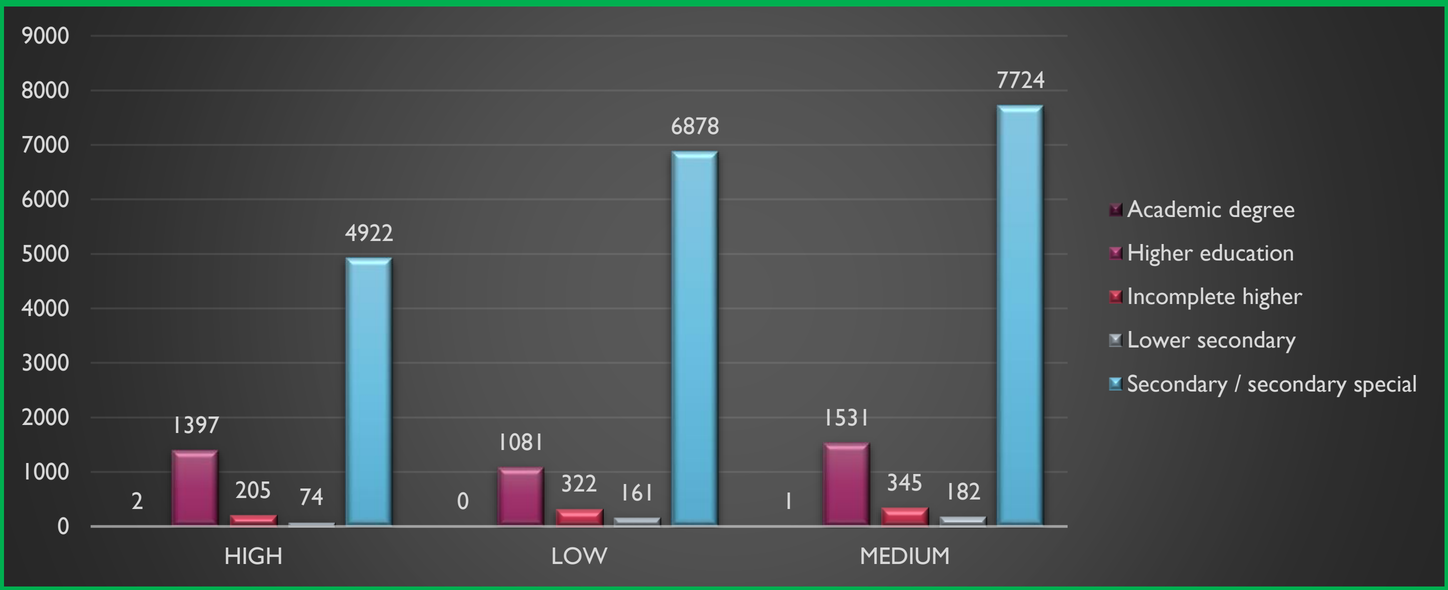


- From the charts above we can infer that the "LOW" category from AMT_CREDIT and "SECONDARY/SECONDARY SPECIAL" have the highest count of repaying their loans back to the bank.

TARGET 1 : AMT_CREDIT_RANGE VS NAME_EDUCATION TYPE

TARGET 1

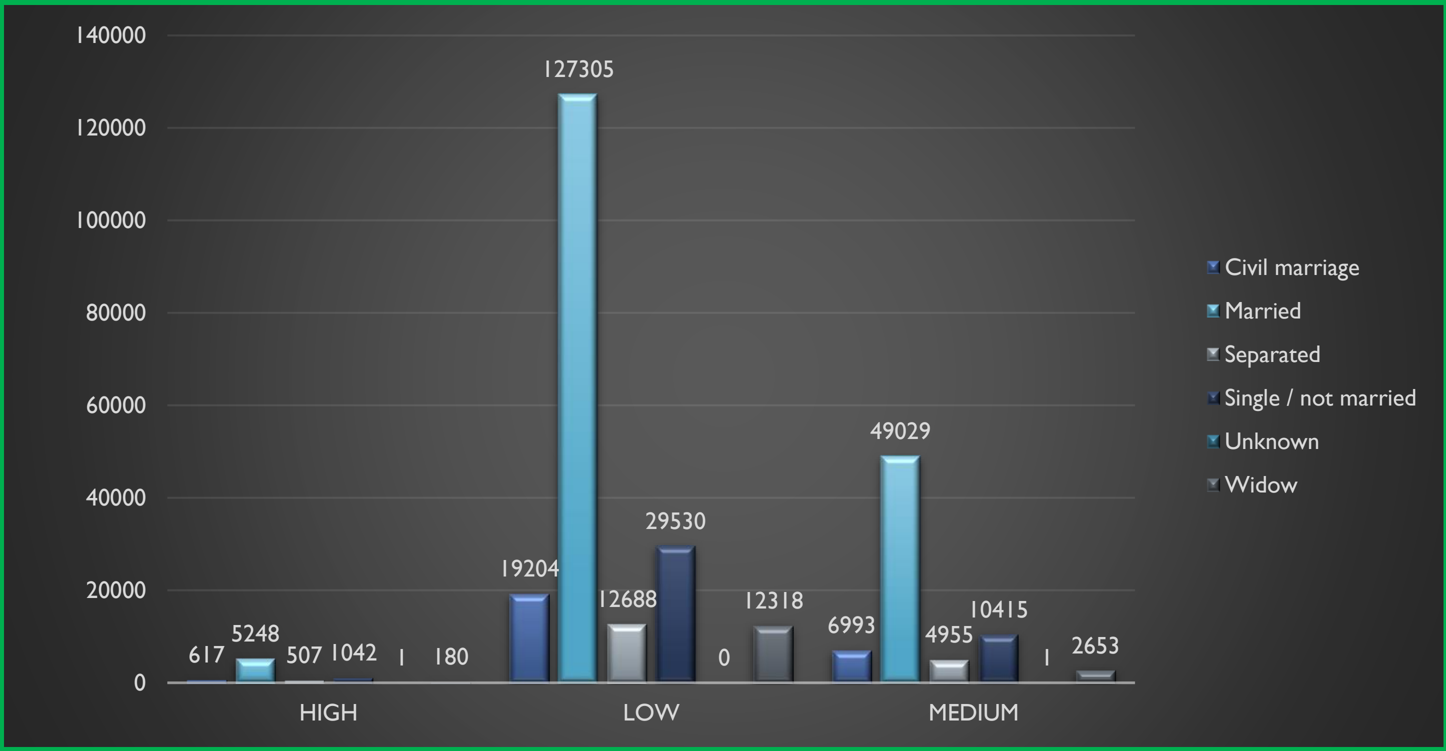
Count of NAME_EDUCATION_TYPE	Column Labels					
Row Labels	Academic degree	Higher education	Incomplete higher	Lower secondary	Secondary / secondary special	Grand Total
HIGH	2	1397	205	74	4922	6600
LOW		1081	322	161	6878	8442
MEDIUM	1	1531	345	182	7724	9783
Grand Total	**3**	**4009**	**872**	**417**	**19524**	**24825**



- From the above charts we can say that "MEDIUM" category from AMT_CREDIT and "SECONDARY/SECONDARY SPECIAL" have the highest count of clients not paying their loans or have some issues repaying the loans.
- By this analysis, bank will take a decision regarding which client with what education background is trustable and who is not.

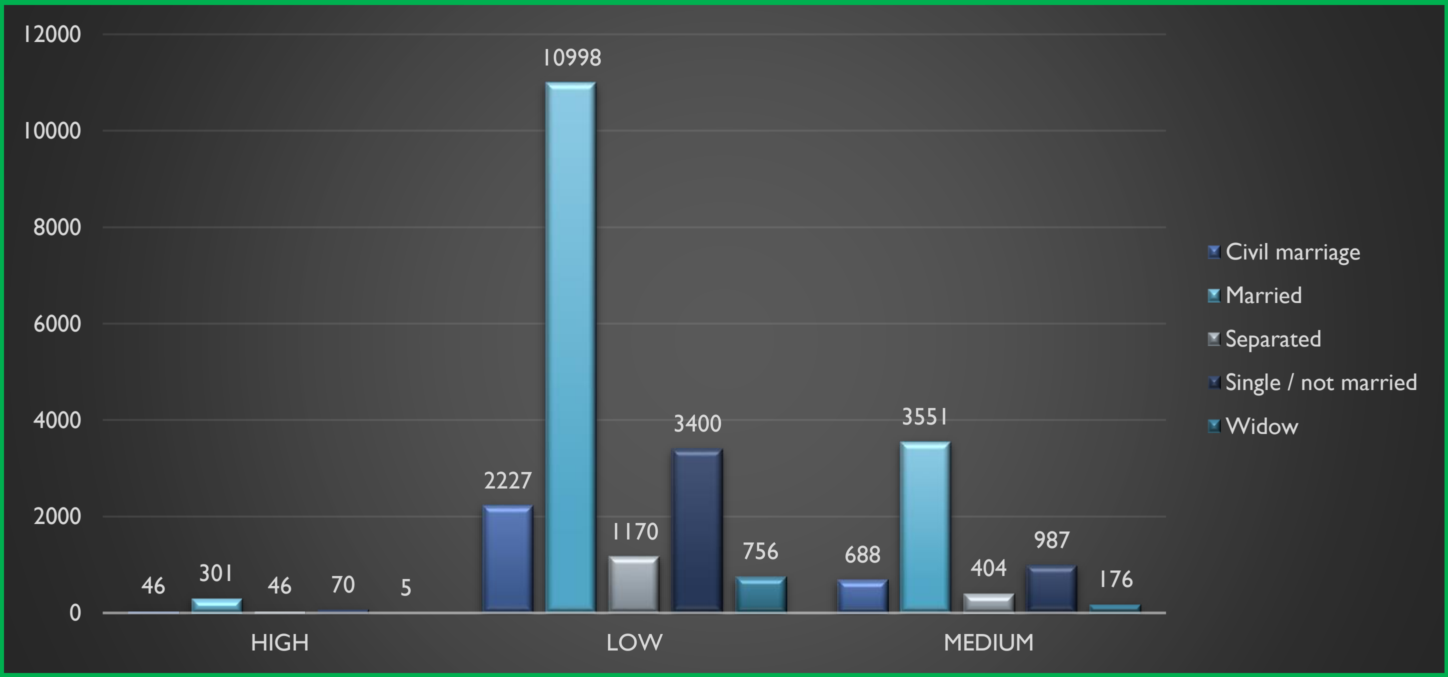
- TOTAL_INCOME VS FAMILY STATUS:

TARGET 0 : TOTAL_INCOME VS FAMILY STATUS							
TARGET	0						
Count of NAME_FAMILY_STATUS	Column Labels						
Row Labels	**Civil marriage**	**Married**	**Separated**	**Single / not married**	**Unknown**	**Widow**	**Grand Total**
HIGH	617	5248	507	1042	1	180	7595
LOW	19204	127305	12688	29530		12318	201045
MEDIUM	6993	49029	4955	10415	1	2653	74046
Grand Total	**26814**	**181582**	**18150**	**40987**	**2**	**15151**	**282686**



- From the charts above we can say that client who is MARRIED and has a LOW income has higher chances of repaying the loan.

TARGET	1					
Count of NAME_FAMIL	Column Labels					
Row Labels	Civil marriage	Married	Separated	Single / not married	Widow	Grand Total
HIGH	46	301	46	70	5	468
LOW	2227	10998	1170	3400	756	18551
MEDIUM	688	3551	404	987	176	5806
Grand Total	2961	14850	1620	4457	937	24825



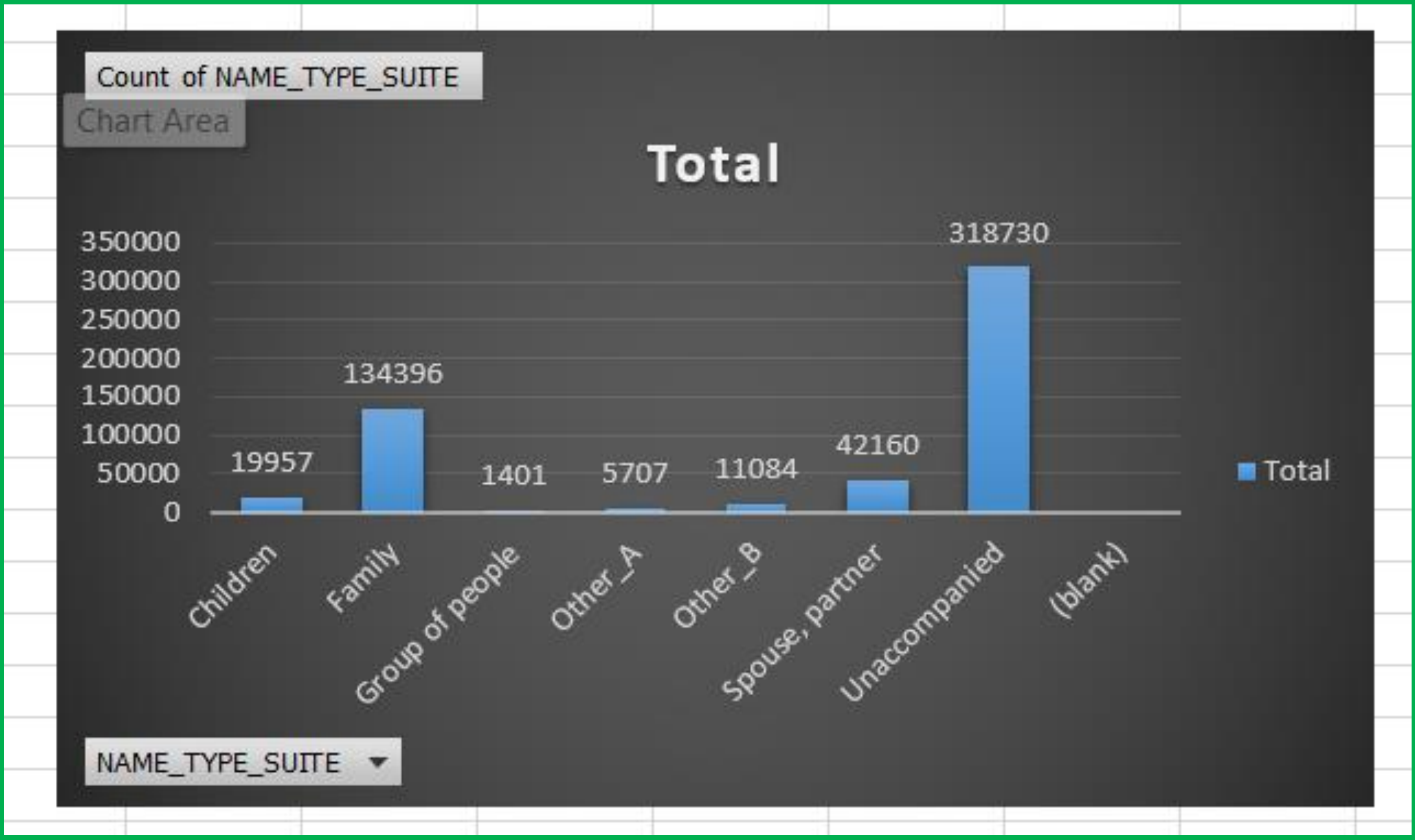
- From the charts above we can infer that a MARRIED client with a low income has highest count for not paying their loan back to the banks.

- With this we complete all the analysis on APPLICATION_DESCRIPTION.XLSX file.

PREVIOUS APPLICATION

- Again we will first drop all the columns which are not required for our analysis or which are irrelevant to our analysis.
- Names of columns I have dropped are:
    1. 'HOUR_APPR_PROCESS_START'
    2. 'WEEKDAY_APPR_PROCESS_START'
    3. 'FLAG_LAST_APPL_PER_CONTRACT'
    4. 'NFLAG_LAST_APPL_IN_DAY'
    5. 'SK_ID_CURR'
- Next we move on to imputing and analyzing null values across the dataset.
- Starting from NAME_TYPE_SUITE column:
- We start with counting each value from the column

Row Labels	Count of NAME_TYPE_SUITE
Children	19957
Family	134396
Group of people	1401
Other_A	5707
Other_B	11084
Spouse, partner	42160
Unaccompanied	318730
(blank)	
Grand Total	**533435**

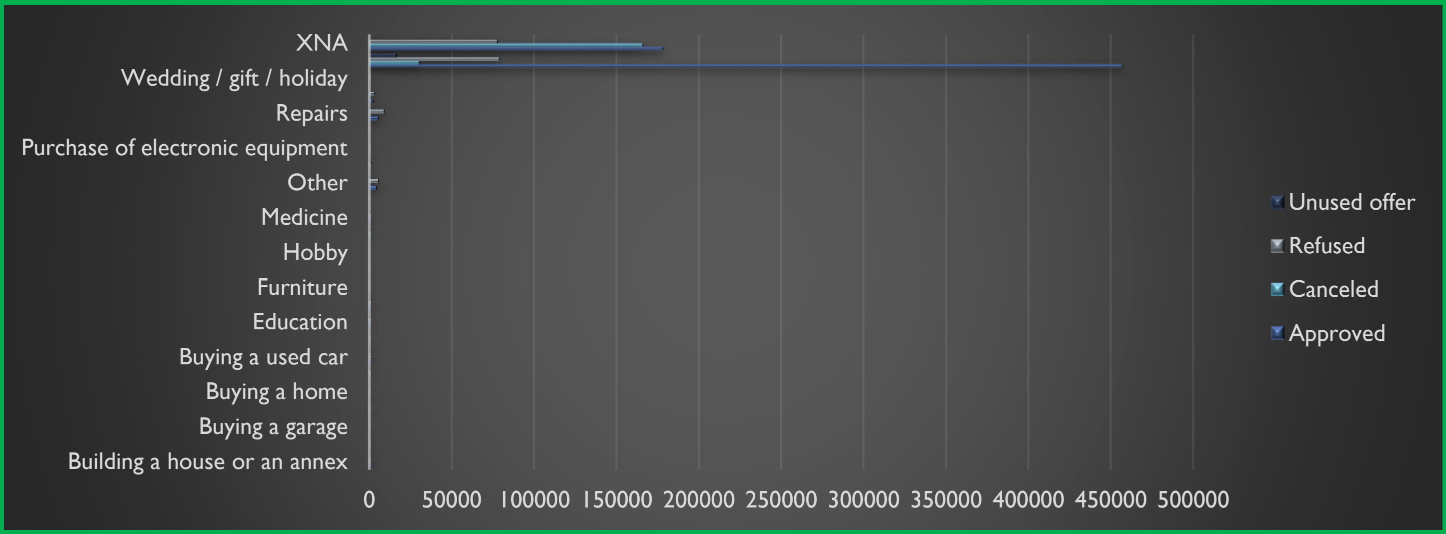


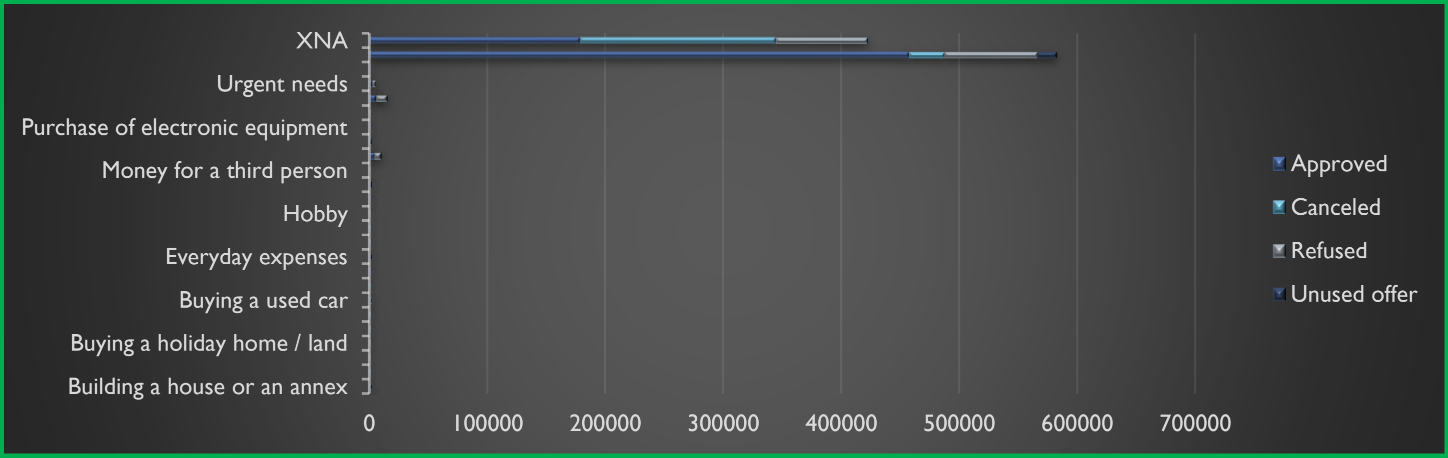
- Here I have found out the count of values first, then plotted them using bar plot.
- Once I have count of each value, I will replace blanks with most frequent value from the column.
- In this case it is "UNACCOMPANIED".

- DISTRIBUTION OF NAME CONTRACT STATUS:

Count of NAME_CONTRACT_STATUS	Column Labels				
Row Labels	Approved	Canceled	Refused	Unused offer	Grand Total
Building a house or an annex	434	60	1188		1682
Business development	78	12	164		254
Buying a garage	28	5	51		84
Buying a holiday home / land	91	13	230		334
Buying a home	130	23	393		546
Buying a new car	139	29	465	4	637
Buying a used car	552	57	1166	9	1784
Car repairs	223	14	256		493
Education	481	14	476	4	975
Everyday expenses	732	8	740	7	1487
Furniture	210	15	250		475
Gasification / water supply	75	3	125		203
Hobby	11		20		31
Journey	329	10	404	2	745
Medicine	676	25	696	5	1402
Money for a third person	10		6		16
Other	4106	186	5310	62	9664
Payments on other loans	189	45	973	3	1210
Purchase of electronic equipment	357	4	280	3	644
Refusal to name the goal	1		7		8
Repairs	5385	381	8973	28	14767
Urgent needs	2228	83	2998		5309
Wedding / gift / holiday	248	10	336		594
XAP	457147	30216	78886	16465	582714
XNA	178626	166018	77690	183	422517
Grand Total	**652486**	**197231**	**182083**	**16775**	**1048575**

- Here we can see apart from XNA and XAP; REPAIR category has the highest count of approved loans.





CONCLUSION

From the following analysis we can draw few conclusions:

- The percentage of Non-payer i.e. target =1 is around 8% and for Payer it is 92%.
- Bank normally approves more loan to female clients as compared to male.
- Clients who belong to working class have a tendency to pay their loans on time.
- Clients with education status Secondary/secondary special or more have higher chances of paying loans on time.
- Clients who fall in age group of 31-40 have higher chances of paying off their loans.
- Clients with LOW credit amount range have a tendency to repay their loans on time than HIGH or MEDIUM credit range holders.
- Clients living with their parents have higher chances of paying off their loans on or before time compared to other housing types.
- Clients applying for Home Loans, Car Loan with income type as State Servant tend to pay their loans on time.
- Bank should be careful before lending out money to clients with Repairs as purpose as they have high count of Non-payers.
- This was my analysis of BANK LOAN CASE STUDY.

THANK

YOU