Data Science Project

PREDICT LOAN STATUS WITH LOGISTIC REGRESSION

https://github.com/hardiantots/predictloanstatus_LogReg

Hardianto Tandi Seno

hardiantotandiseno@gmail.com

ABOUT PRESENTATION

Contains the creation of a **classification model** to **predict a person's credit risk**. Where credit risk status is divided into **two classifications (Binary Classification)** and **viewed based on loan status (0 for "Charged Off", 1 for "Fully Paid")**. The **label** in this **dataset** is in the **"loan_status"** column, because this column is the final determinant of whether a person's credit risk is low or high based on their loan status.

ABOUT DATASET

```
data.shape
```

```
(466285, 75)
```

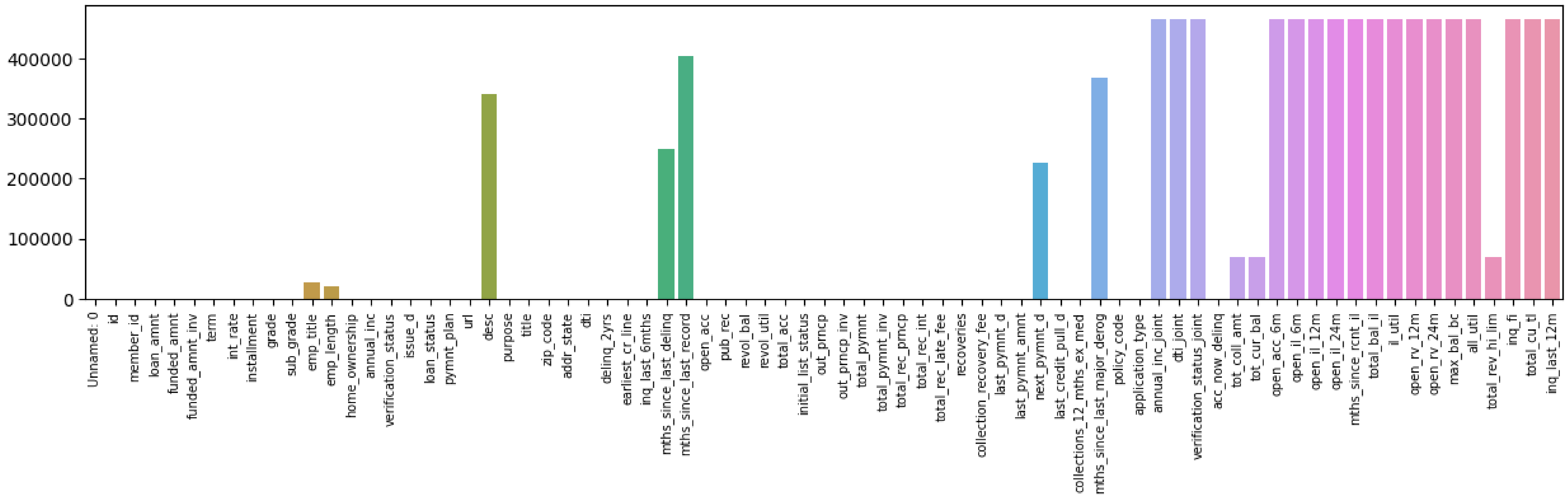
The dataset used is the **loan dataset 2007-2014**, with a total of **75 columns & 466.285 rows.** Of course, not all columns will be used for the modeling process and data pre-processing will be carried out to ensure that the resulting model is accurate for predicting credit risk based on loan status.

DATA PRE-PROCESSING 1

- DROP COLUMN WITH > 50% MISSING VALUE
- DROP SOME UNNEEDED COLUMN
- CHANGE LOAN STATUS FOR ONLY 2 CATEGORIES
- SEPARATE DATA BASED ON CATEGORICAL & NUMERICAL
- FILL NAN VALUE IN COLUMNS

DATA PRE-PROCESSING 1

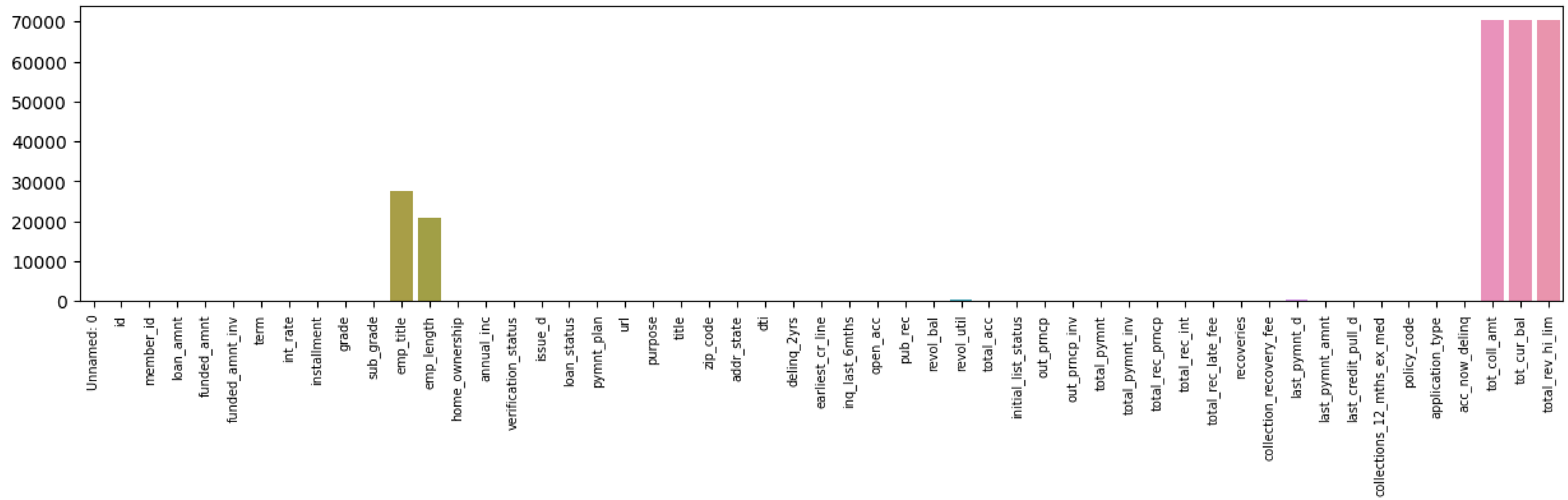
- DROP COLUMN WITH > 50% MISSING VALUE



Here you can see a number of columns with missing values of more than 50%. The benefits of doing this are to improve data quality, simplify datasets, and can improve performance during data analysis and modeling

DATA PRE-PROCESSING 1

- DROP COLUMN WITH > 50% MISSING VALUE



Here you can see the final column from the results of the drop missing value column, where quite a number of columns were dropped.

DATA PRE-PROCESSING 1

- DROP SOME UNNEEDED COLUMN

The next step is to drop some unneeded column. This is done because there are several columns that will not play an important role in the modeling process later. The list of columns that are not needed are:

'UNNAMED: 0'
'ID'
'MEMBER_ID'
'FUNDED_AMNT'
'FUNDED_AMNT_INV'
'INT_RATE'
'SUB_GRADE'
'EMP_TITLE'
'issue_d'
'pymnt_plan'
'url'
'zip_code'
'OUT_PRNCP'
'OUT_PRNCP_INV'
'TOTAL_PYMNT'
'TOTAL_PYMNT_INV'
'TOTAL_REC_PRNCP'
'TOTAL_REC_INT'
'TOTAL_REC_LATE_FEE'
'RECOVERIES'
'COLLECTION_RECOVERY_FEE'
'LAST_PYMNT_D'
'LAST_PYMNT_AMNT'

DATA PRE-PROCESSING 1

- CHANGE LOAN STATUS FOR ONLY 2 CATEGORIES

The reason i change loan status for only 2 categories is because I see only 2 categories that can be a benchmark for whether a credit loan is accepted or not

The following is a list of loan statuses in the dataset. At this moment **I only took 2 categories, namely "Fully Paid" & "Charged Off"**. The reason is that these 2 categories already have certainty regarding loan status and it is unlikely that any changes will occur

```
Current                                                224226
Fully Paid                                             184739
Charged Off                                             42475
Late (31-120 days)                                       6900
In Grace Period                                          3146
Does not meet the credit policy. Status:Fully Paid       1988
Late (16-30 days)                                        1218
Default                                                   832
Does not meet the credit policy. Status:Charged Off       761
Name: loan_status, dtype: int64
```

DATA PRE-PROCESSING 1

- SEPARATE DATA BASED ON CATEGORICAL & NUMERICAL

The process of separating data based on categorical and numerical is useful when entering the data transformation process (encoding categorical data). Because we will find it easy to find out which column contains categorical data so that the encoding process can take place more quickly.

To see the complete process of separated data, you can look at the ipynb file in the repository

DATA PRE-PROCESSING 1

- FILL NAN VALUE IN COLUMNS

```
loan_amnt                     0
term                          0
installment                   0
grade                         0
emp_length                    0
home_ownership                0
annual_inc                    0
verification_status           0
loan_status                   0
purpose                       0
title                         0
addr_state                    0
dti                           0
delinq_2yrs                   0
earliest_cr_line              0
inq_last_6mths                0
open_acc                      0
pub_rec                       0
revol_bal                     0
revol_util                    0
total_acc                     0
initial_list_status           0
last_credit_pull_d            0
collections_12_mths_ex_med    0
policy_code                   0
application_type              0
acc_now_delinq                0
tot_coll_amt                  0
tot_cur_bal                   0
total_rev_hi_lim              0
dtype: int64
```

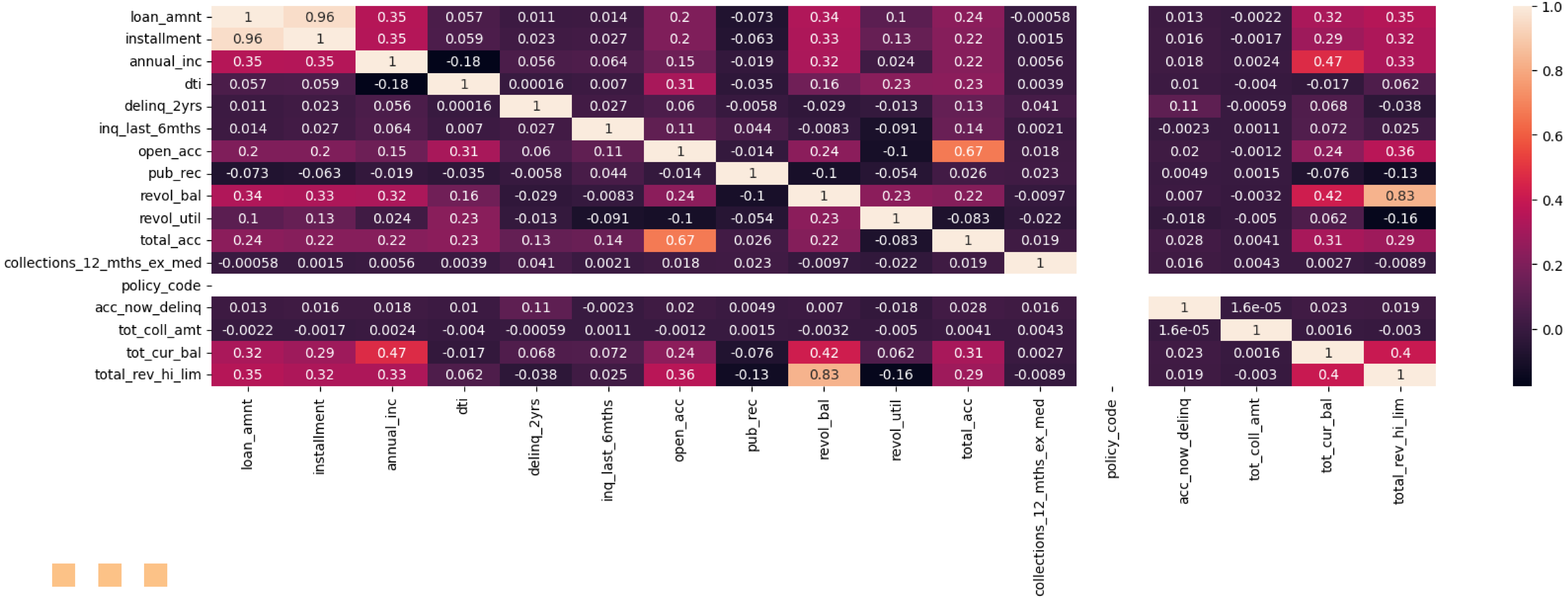
For the fill nan value process, in **numerical data**, the fill process is carried out by giving the **mean value** of the **column that contains the NaN value.** For **categorical data**, the fill process is carried out by **changing the NaN value** to the **string 'unknown'**.

In the picture beside, shows that there are no more NaN values in each existing column

EXPLORATORY DATA ANALYSIS

USING POWER BI & SEABORN

FOR VISUALIZATION DATA



CORRELATION BETWEEN COLUMN USING SEABORN

(JUST NUMERICAL DATA)

THERE ARE 3 COLUMNS IN THE NUMERICAL DATA THAT HAVE A FAIRLY STRONG CORRELATION, NAMELY :

01

LOAN_AMNT & INSTALLMENT

The amount of the loan given & The monthly payment owed by the borrower

02

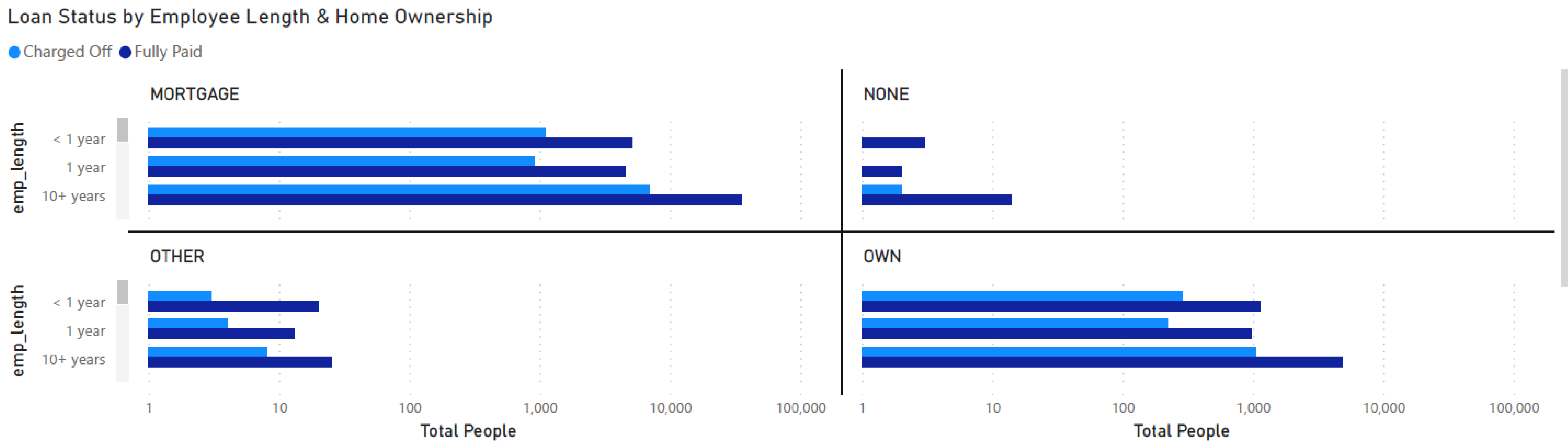
REVOL_BAL & TOTAL_REV_HI_LIM

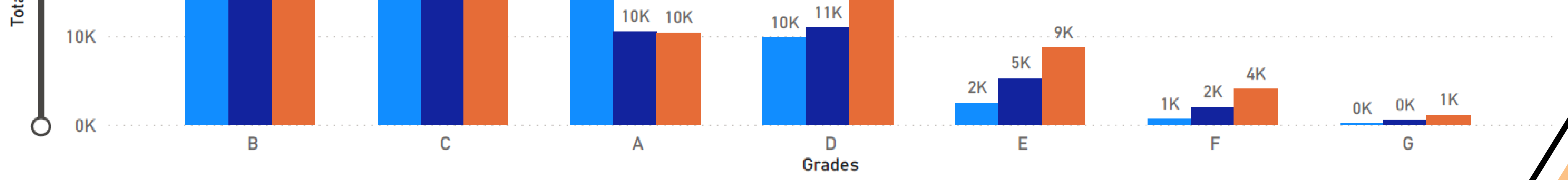
Total credit revolving balance & Total revolving high credit/credit limit

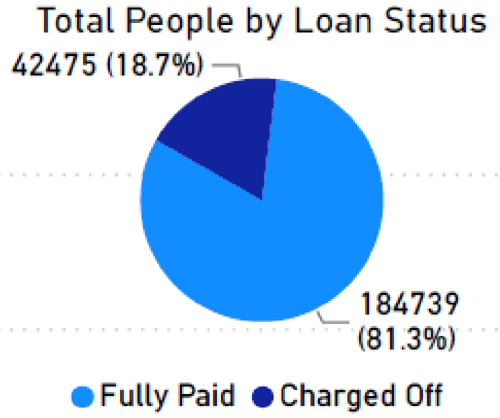
03

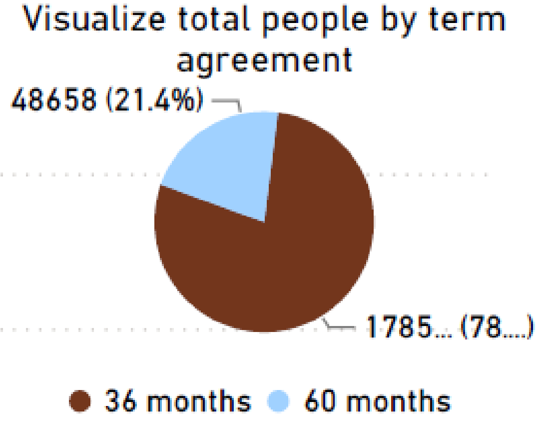
TOTAL_ACC & OPEN ACC

The total number of credit lines currently in the borrower's credit file & The number of open credit lines in the borrower's credit file.



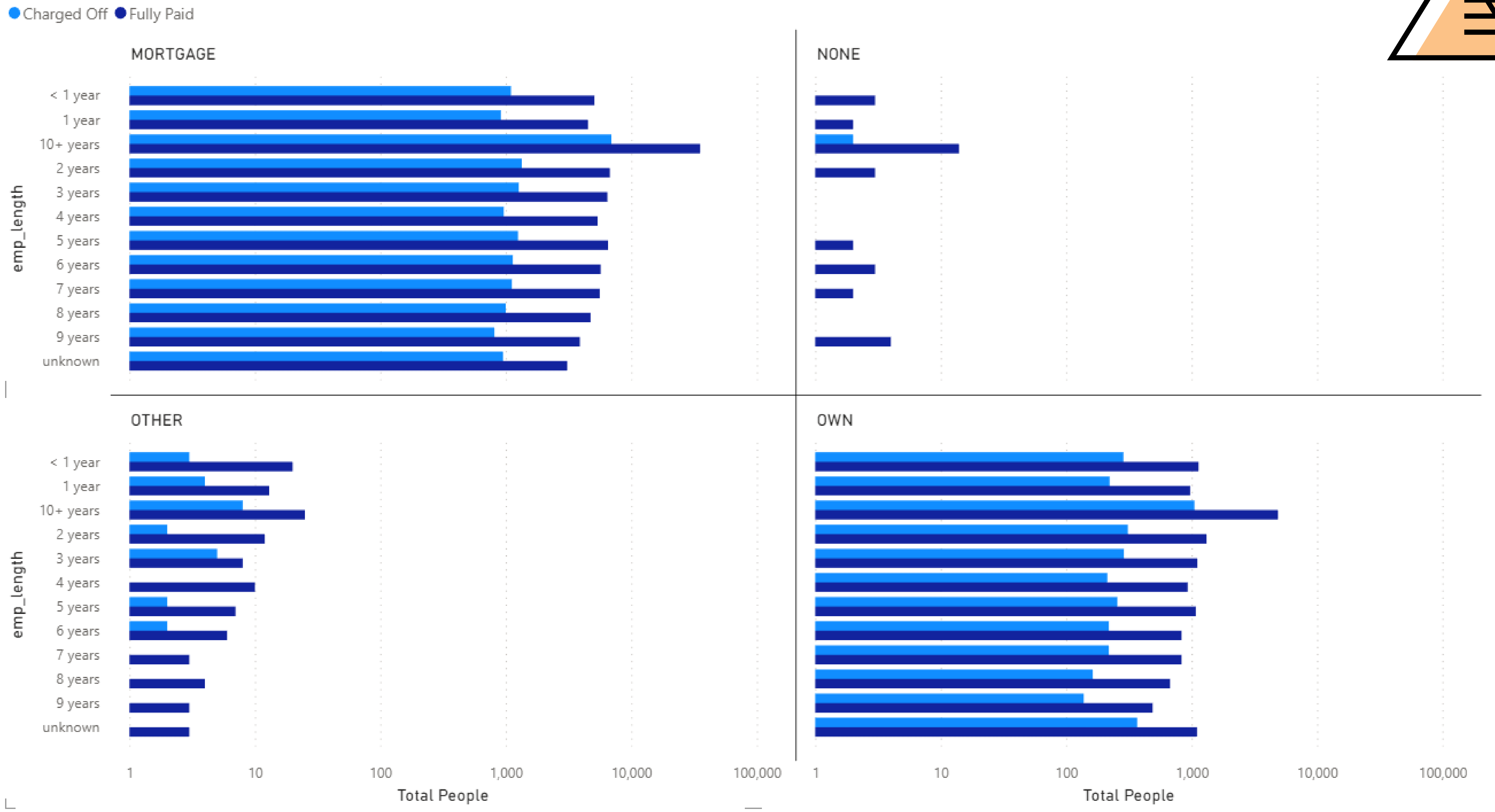




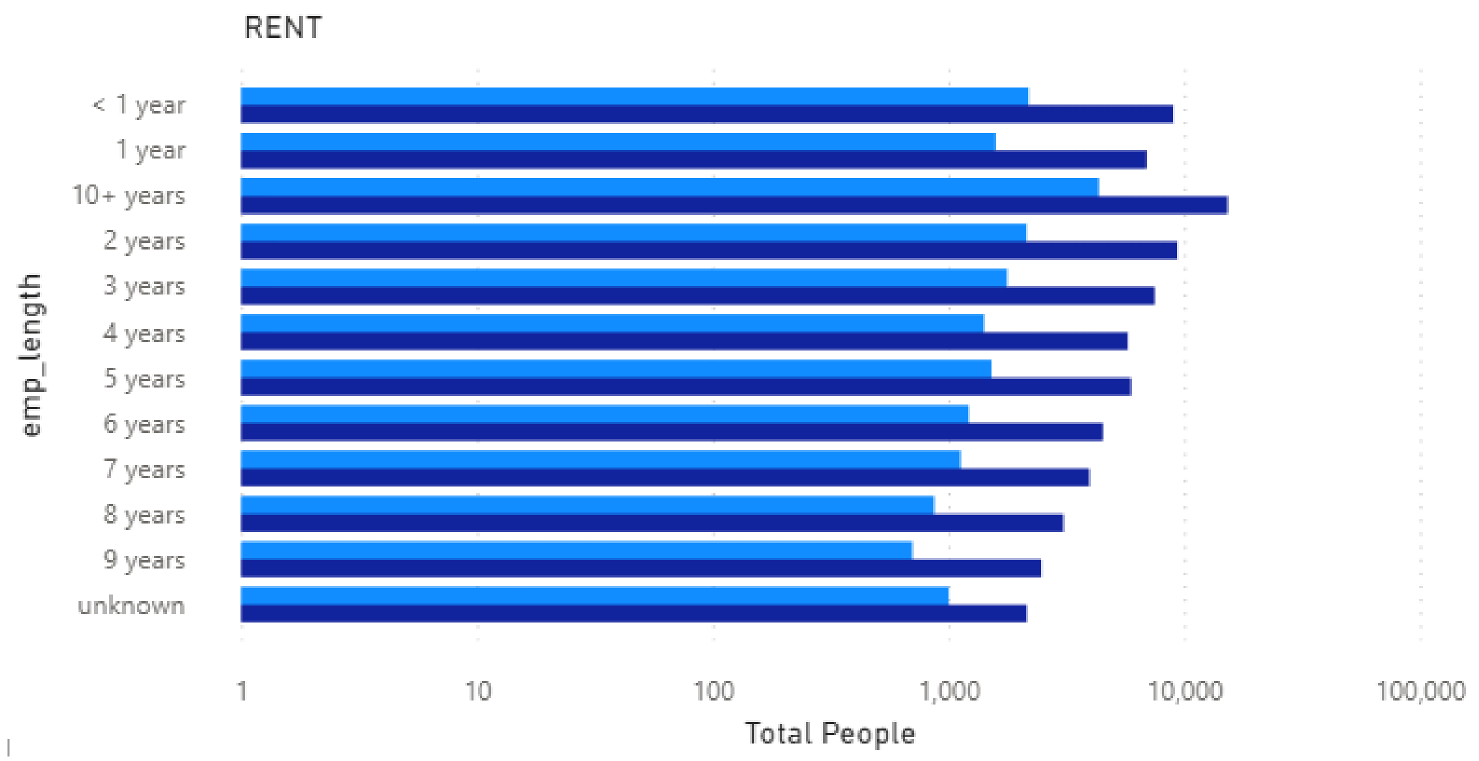


VISUALIZATION DATA ON POWER BI

LOAN STATUS BY EMPLOYEE LENGTH & HOME OWNERSHIP



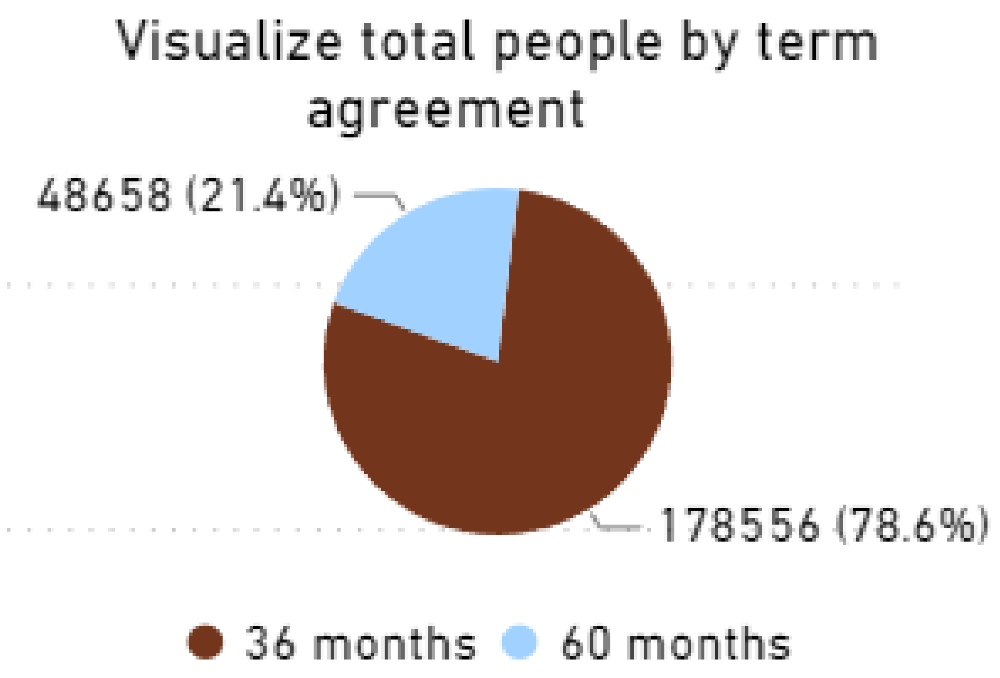
LOAN STATUS BY EMPLOYEE LENGTH & HOME OWNERSHIP



From the visualization of loan status based on length of work & home ownership, there are several summaries that we can put forward, such as:

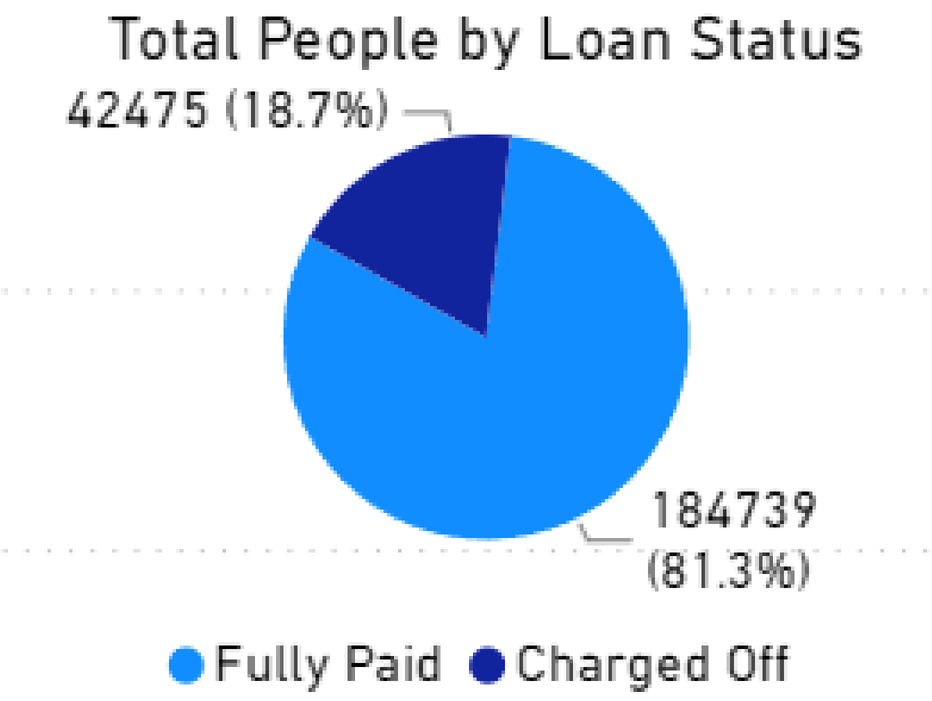
- The **3 categories of home ownership** that most borrowers are **"Mortgage", "Own", & "Rent"**
- In the chart of **loan status based on employee length and home ownership**, it can be seen that each category, both **based on employee length (< 1 year - 10+ years)** and **based on home ownership**, has **almost the same graph**, where **most of the loan status is "Fully Paid"**, even though there are also **quite a lot of loans with "Charged off" status**.
- **Conclusion** that can be drawn is that **employee length and home ownership** have **less influence on loan status** because they **have similar tendencies**.

TOTAL PEOPLE BY TERM AGREEMENT



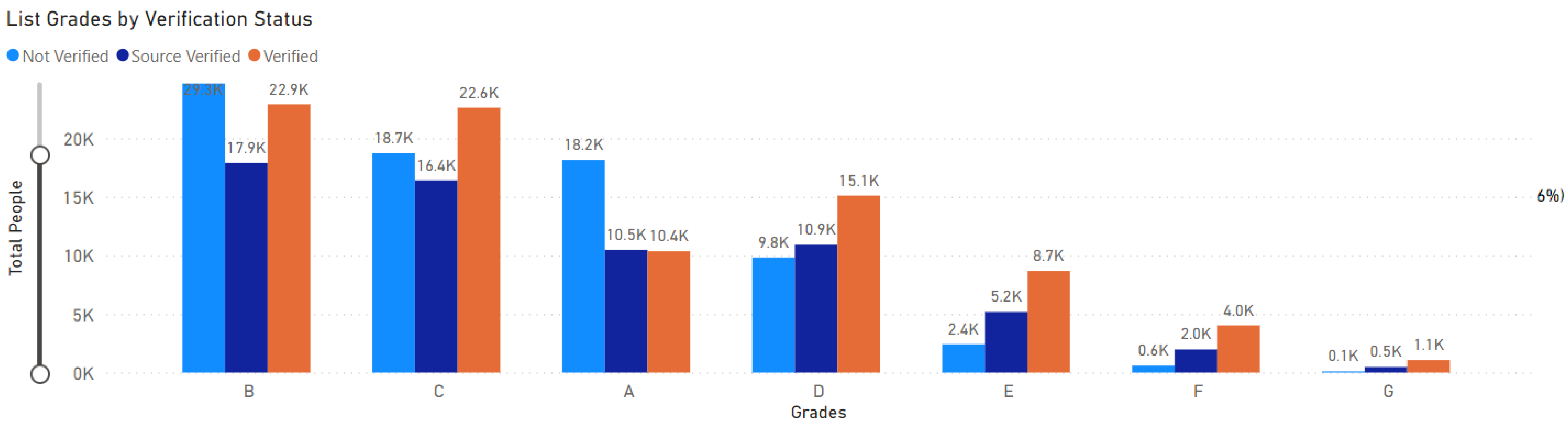
From this visualization, we can conclude that around **78.6% (178556) people** agree to a term agreement of **36 months** and around **21.4% (48658) of people** agree to a term agreement of **64 months.**

TOTAL PEOPLE BY LOAN STATUS



From this visualization it can be concluded that around **81.3% (184739) people** have a **loan status "Fully Paid"** or have paid off a loan and around **18.7% (42475) people** have a **loan status "Charged Off"** or the borrower is unable to repay the loan so that it is considered a loss for the lender.

LIST GRADES BY VERIFICATION STATUS



Grade when applying for a loan shows the **range of loans applied for**. From the existing graph, it can be seen that **grade B is the grade with the most borrowers**, followed by **grade C & grade A**. This means that the **average range of loans proposed is quite high. Verification status is very important** to ensure that the **lender gives the option not to submit any collateral for the loan** to the right borrower.

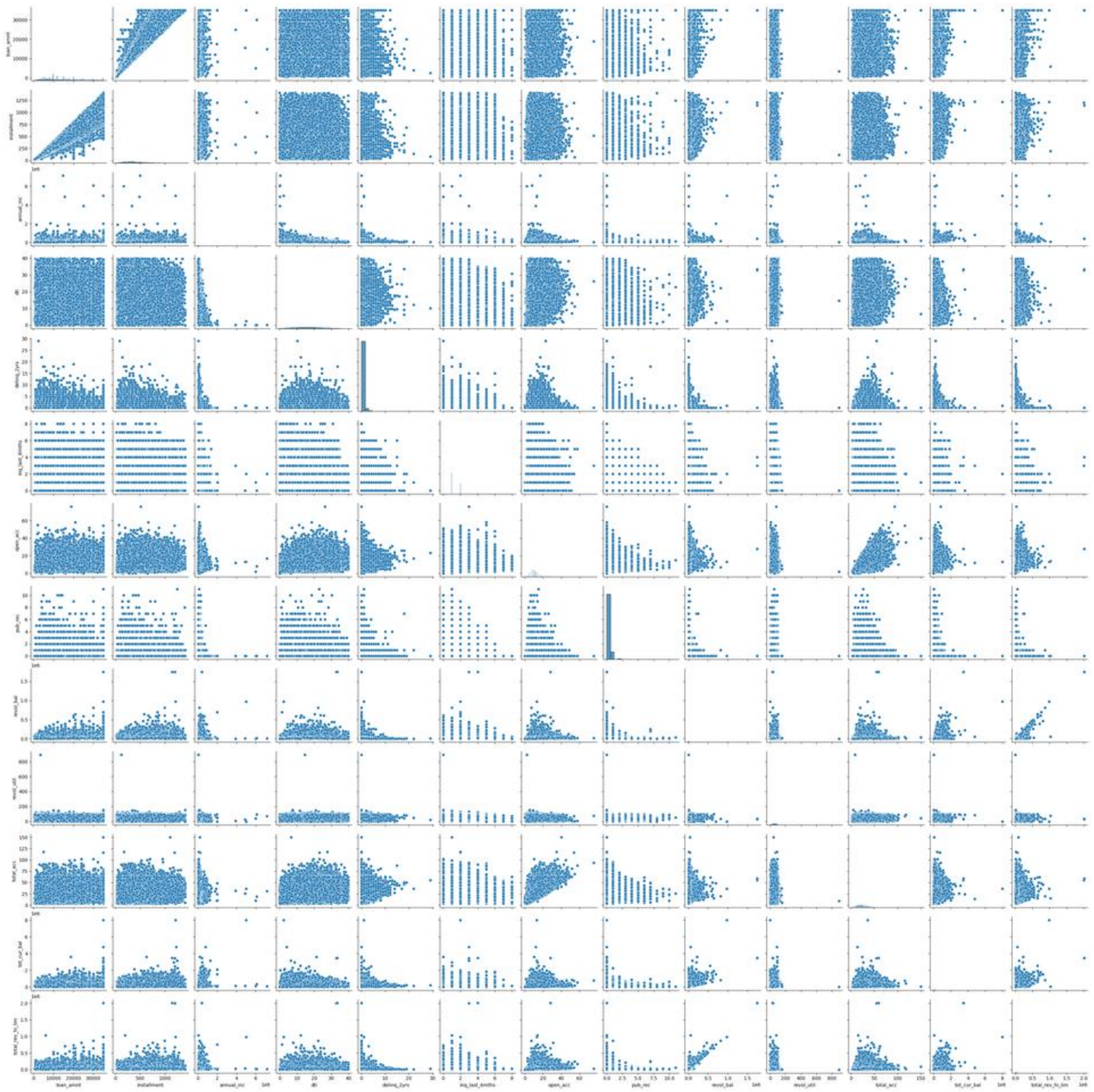
Then from these grades, it can be seen that the 3 highest grades, there are still **many verification status "Not Verified"**.
- **In Grade B, "Not Verified" status verification dominates**, although "Verified" status is also high. For "Source Verified" status, it is the lowest among the three categories.
- **In Grade C, it has the reverse condition with Grade B**, where **"Verified" has the most status** followed by "Not Verified".
- **In Grade A, the conditions are almost the same as Grade B**, but the **distance between "Not Verified" status & "Verified" status is quite far**.
- **For the order of 4 and so on**, it can be seen that **"Verified" status is the most among the others**.

DATA PRE-PROCESSING 2

- CLEAR OUTLIER
- TRANSFORMATION DATA
- CHOOSE FEATURE & TARGET
- SET TRAINING & TEST SET FROM DATA

DATA PRE-PROCESSING 2

- CLEAR OUTLIER



First, we have to visualize the data (only numeric data, because categorical data cannot be visualized before starting the encoding operation) to see the outliers in each column.

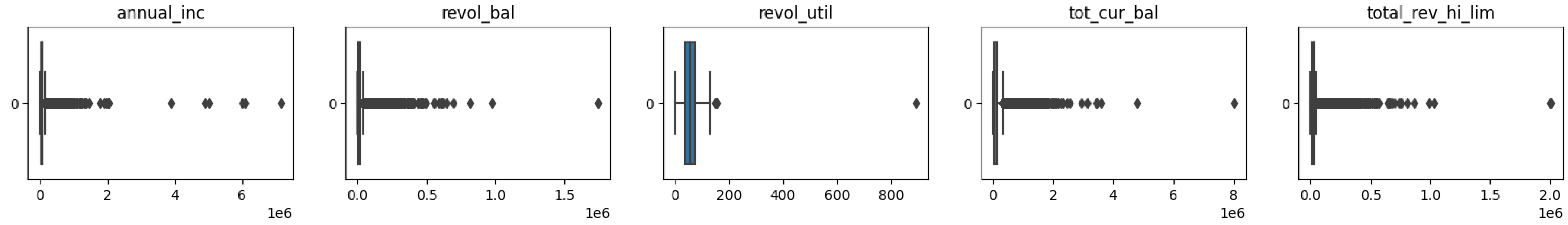
For a more detailed visualization, you can see the ipynb file in the repository because the image width is very large.

DATA PRE-PROCESSING 2

- CLEAR OUTLIER

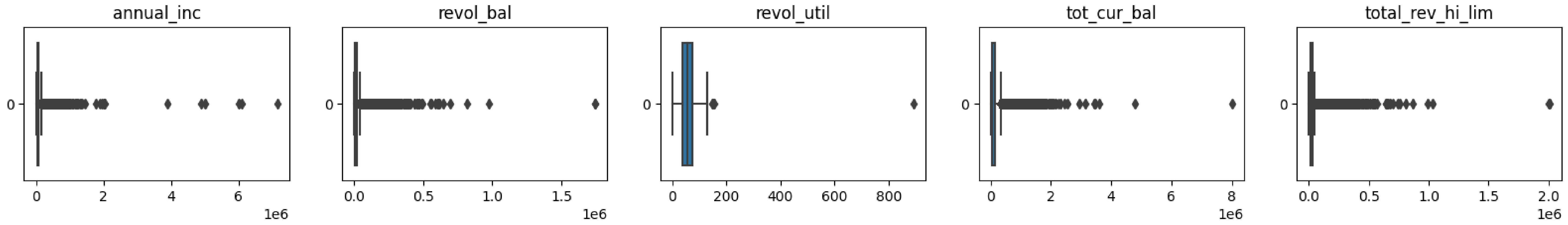
We can directly perform operations on certain columns (the range of values on the chart is the same as those in the column) that have **outliers by performing a comparison operation**, then the **value of the operation is updated in the selected column.**

For **a range of values on the chart that are not the same as those in the column**, we can see further **outliers with boxplots**



DATA PRE-PROCESSING 2

- CLEAR OUTLIER

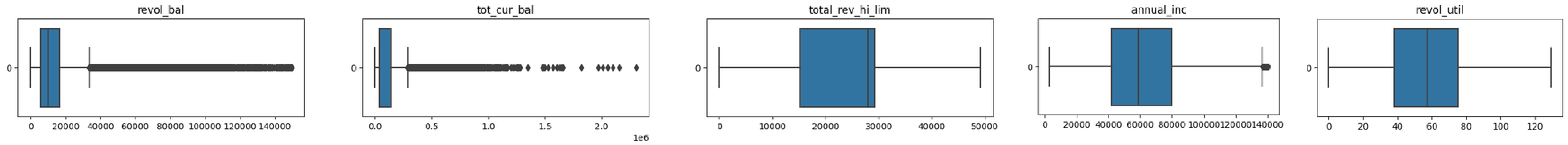


There are still **5 columns whose outliers will be removed**, what can be done is by applying the **IQR (Interquartile Range)**. Using the **IQR method**, **outliers** that are **well below or above the typical range of values in a data set** can be **detected and potentially removed**.

DATA PRE-PROCESSING 2

- CLEAR OUTLIER

The following is the boxplot after removing the outliers in the five columns:



After the process of removing outliers is carried out, there is a reduction in the data that was:

227.214 rows of data to **196.415 rows of data**

DATA PRE-PROCESSING 2

- TRANSFORMATION DATA

After cleaning the outliers, the next step is to perform data **transformation**, where the **numerical data will be normalized for better modelling (using MinMaxScaler)** and the **categorical data will be encoded** to **change the data in the column into numbers (using LabelEncoder).**
To do this transformation, can use scikit-learn library

DATA PRE-PROCESSING 2

- CHOOSE FEATURE & TARGET

After that, we must choose **feature column & target column** for modelling process later.

Featue Columns : All columns without a loan status column
Target Column : Just loan status column

DATA PRE-PROCESSING 2

- SET TRAINING & TEST SET FROM DATA

Set training & test set from data with using scikit-learn library with :

- **Test set data** set to **0.2 (20%)**
- **Training set data** set to **0.8 (80%)**

MODELLING PROCESS

MODEL THAT USED

Model that will used for this scenario

EVALUATION METRICS

Evaluate model performance using some metrics like precision, recall, f1-score, and ROC Curve

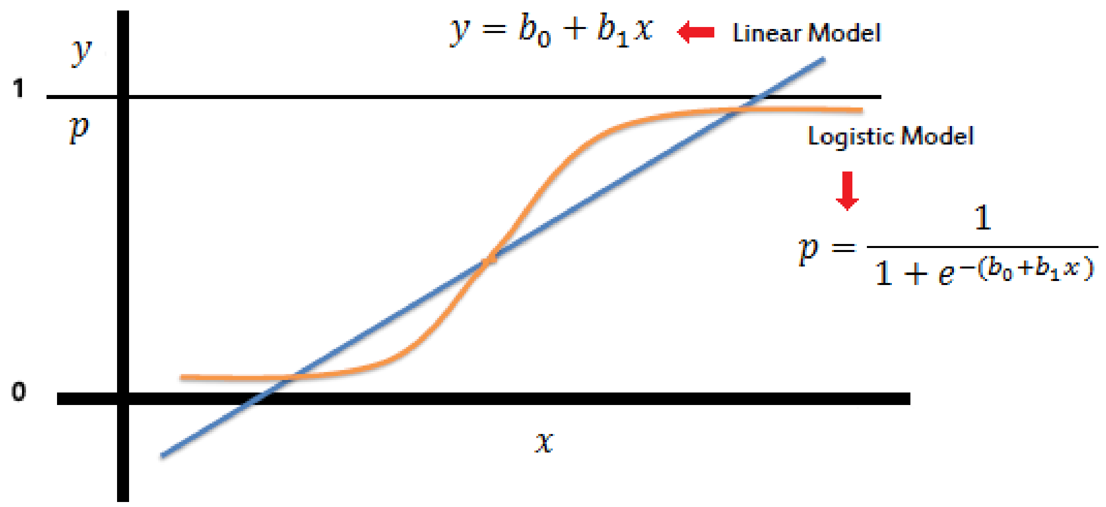
SAVE MODEL

Model can be save for using in application that need this model

MODELLING PROCESS

MODEL THAT USED

Model that will used for this scenario is **Logistic Regression (Classfication Model)**. **Logistic regression** is a **data analysis technique** that **uses mathematics to find the relationship between two data factors**. It then uses this relationship to predict the value of one of these factors based on the other factors.



MODELLING PROCESS

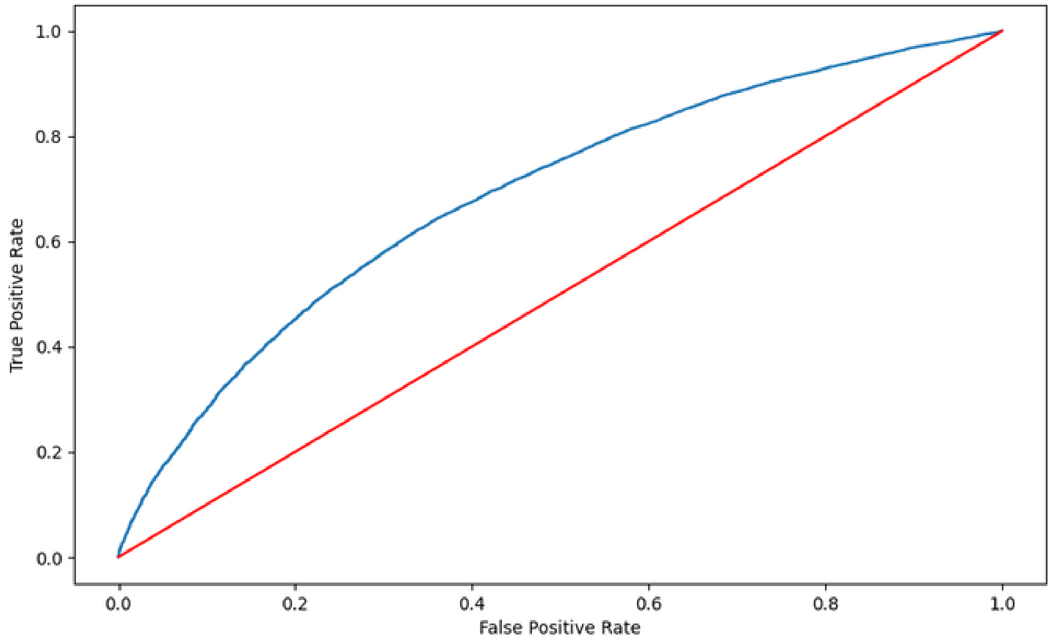
EVALUATION METRICS

Evaluate model performance using some metrics like **precision, recall, f1-score, and ROC Curve.**

Result of Precision, Recall, & f1 Score from model

```
Precision Score :  80.7%
Recall Score :  100.0%
f1 Score :  89.3%
```

Result of ROC Curve :



*Conclusion for detailed information about evaluation

MODELLING PROCESS

SAVE MODEL

Model can be save for using in application that need this model. We can save the model with using library called Pickle.

```python
import pickle

with open('model.pkl', 'wb') as f:
    pickle.dump(model, f)
```

python

CONCLUSION

- The purpose of this project is to **predict the loan status of prospective borrowers** so that **lenders can find out** the **eligibility status of granting loans**.
- Many **columns in the dataset are dropped**, either because there are many missing values in these columns or these columns are less useful for the process of extracting information later.
- From the existing dataset, it turns out that **more than 70% of the loan status is "Fully Paid"**
- In this project, **two data pre-processing and one Exploratory Data Analysis (EDA)** were carried out. **For the EDA process itself, a more in-depth analysis can still be carried out regarding the relationships in the column.**
- The modeling process uses the **Logistic Regression model**, because it uses mathematics to find the relationship between two data factors. It then uses this relationship to predict the value of one factor based on the other factor. I feel this model is suitable to be applied in this case.
- In the **evaluation metrics section**, I feel that the **performance of this model can still be improved**, this is because **no processing has been carried out on categorical data** (such as removing outliers, etc.). Besides that, I **haven't used hyperparameter tuning** (using GridSearchCV) which can be **used to find the best parameters for the modeling process.**

THAT'S ALL FROM ME
THANK YOU

LINKEDIN

Hardianto Tandi Seno

E-MAIL

hardiantotandiseno@gmail.com