# Delinquency Analysis Report

# Python Data Preprocessing and Analysis Report

# 1. Introduction

This report outlines the steps taken to preprocess and analyze a loan dataset using Python. The dataset, stored in a `.csv` file named `loanexpert.csv`, was cleaned and processed to prepare it for further analysis.

# 2. Importing Libraries

Pandas and NumPy were imported to handle the data manipulation and numerical operations, respectively.

import pandas as pd  
import numpy as np

# 3. Reading the Data

The `.psv` file `loanexpert.csv` was read into a Pandas DataFrame. The `info` and `shape` methods were used to inspect the structure and size of the dataset.

df = pd.read\_csv('loanexpert.csv')  
print(df.info())  
print(df.shape)

# 4. Dropping Unwanted Columns

Unnecessary columns were identified and removed using a for loop. The columns dropped were:

- payment date, maturity date, MSA, occupancy, channel, product type, loan, sequence sum, number, seller name, postal code, service name, property state, PPM, property type, loan purpose, loan borrower.

columns\_to\_drop = ['payment date', 'maturity date', 'MSA', 'occupancy', 'channel',   
 'producttype', 'loanseqnum', 'seller name','postal code', 'service name', 'property state', 'PPM’,'property type', 'loan purpose', 'loan borrower']  
df.drop(columns=columns\_to\_drop, inplace=True)

# 5. Handling Missing Values

Null values were removed using the custom `dropna` function.

# 6. Checking and Replacing Unique Values

Unique values for each column were checked. For the `CS` (credit score) column, any `0` values were replaced with `850`.

df['CS'].replace(0, 850, inplace=True)

In the `first time homebuyer` column, the unique values were `no`, `n`, `x`, and `y`. The `x` value was dropped, and the remaining values (`n` and `y`) were counted.

df = df[df['first time homebuyer'] != 'x']  
print(df['first time homebuyer'].value\_counts())

# 7. Creating Partitions and Labels

- \*\*Credit Score Range\*\*: Ranges: 0-650 (poor), 650-700 (fair), 700-750 (good), 750-900 (excellent). Checked for unique values and their counts.

def credit\_score\_range(cs):  
 if cs <= 650:  
 return 'poor'  
 elif cs <= 700:  
 return 'fair'  
 elif cs <= 750:  
 return 'good'  
 else:  
 return 'excellent'  
  
df['credit\_score\_range'] = df['CS'].apply(credit\_score\_range)  
print(df['credit\_score\_range'].value\_counts())

- LTV Range: Ranges: 0-25 (low), 25-50 (medium), 50-1000 (high). Checked for value counts.

def ltv\_range(ltv):  
 if ltv <= 25:  
 return 'low'  
 elif ltv <= 50:  
 return 'medium'  
 else:  
 return 'high'  
  
df['LTV\_range'] = df['LTV'].apply(ltv\_range)  
print(df['LTV\_range'].value\_counts())

- Repay Range: Labels: 0-4, 4-8, 8-12, 12-16, 16-20. Applied to the ranges: 0-48, 48-96, 96-144, 144-192, 192-240. Checked for value counts.

df['repay\_range'] = pd.cut(df['repay'], bins=[0, 48, 96, 144, 192, 240], labels=['0-4', '4-8', '8-12', '12-16', '16-20'])  
print(df['repay\_range'].value\_counts())

# 8. Final Data Checks

The DataFrame was checked for any remaining null values. Grouped the data by `LTV\_range` and `repay\_range`, and described the grouped data.

# 9. Extracting the Final File

The cleaned and processed DataFrame was saved back to a file.

df.to\_csv('msb.csv', index=False)

# Dashboard Process

# 1. Key Performance Indicators (KPIs)

* **Average Months Delinquent**: 1.8 months
* **Percentage of Loans Ever Delinquent**: 19.40%

# 2. Distribution of Months Delinquent

* **Chart Type**: Bar Chart
* **Details**: This chart shows the count of months delinquent versus the months delinquent. The highest number of delinquencies is in the range of 0 to 10 months is 202691.

# 3. Percentage of Loans Ever Delinquent by Credit Range

* **Chart Type**: Pie Chart
* **Details**:
  + **Credit Range 1**: 279,482 loans
  + **Credit Range 2**: 54,762 loans
  + **Credit Range 3**: 72,678 loans (highest percentage of loans ever delinquent)
  + **Credit Range 4**: 57,181 loans

# 4. Trend of Delinquency Rates Over Time

* **Chart Type**: Line Graph
* **Details**: This chart shows the trend of delinquency rates over time, with the percentage of loans ever delinquent versus the first payment date.
  + **Year 2447**: 19.40%
  + **Year 2448**: 19.19%
  + **Year 2450**: 24.92%(Highest)
  + **Year 2451**: 23.69%