

# EDA+ Data Cleaning and Pre-Processing

March 7, 2025

## 1 Introduction

### 1.1 Objective

This report provides an exploratory analysis(EDA) of the Auto Loan Credit Decisioning dataset. It includes data visualization, missing value analysis, and necessary preprocessing steps to prepare the data for modeling.

### 1.2 Data Overview

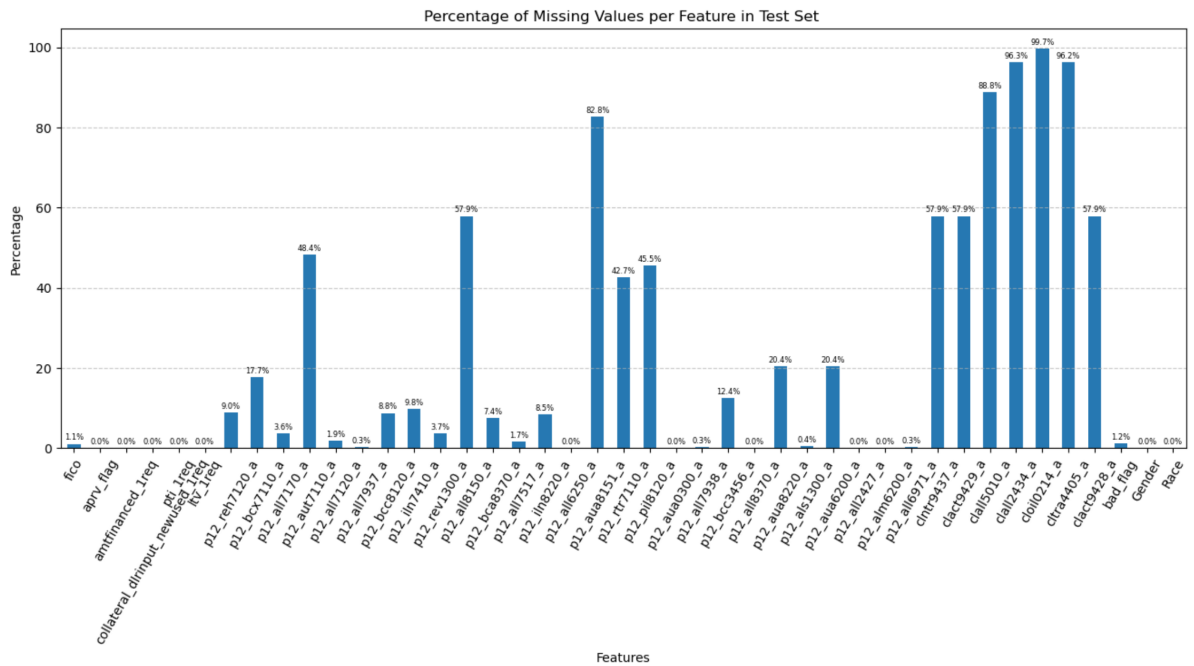
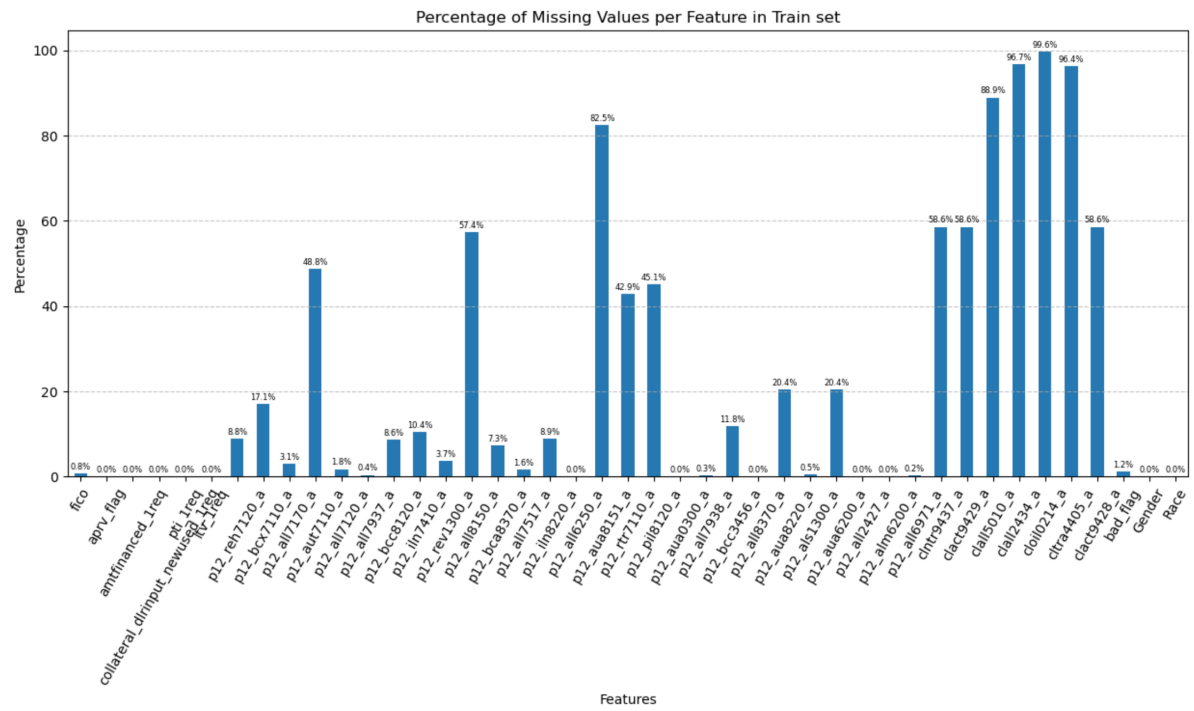
1. Training Dataset:  $\sim 21,000$  records
2. Testing Dataset:  $\sim 5,400$  records
3. Target variable: `apr_v_flag` (loan approval rate)
4. Key Variables:
  - (a) Fico: FICO score
  - (b) `amtfinanced_1req`: Requested loan amount
  - (c) `pti_1req`: Payment-to-income ratio
  - (d) `ltv_1req`: Loan-to-Value ratio

## 2 EDA

### 2.1 Data Summary

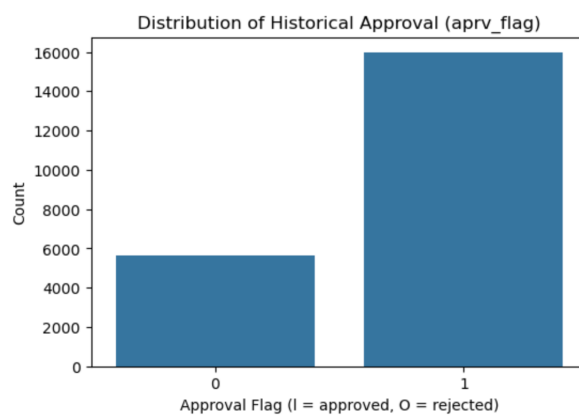
#### 2.1.1 General Information

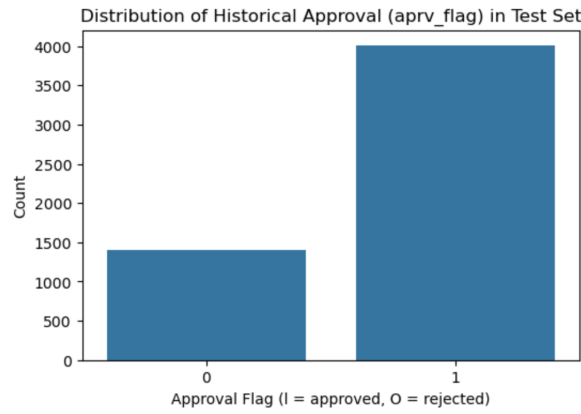
1. Number of features: 43
2. Number of Numerical features: 38
3. Number of Categorical features: 3 which are Race, Gender and `collateral_dlrinput_newused_1req` (if the vehicle is used or not)
4. Number of Missing Values per column:



### 2.1.2 Target Distribution

The target is the `aprv_flag` (approval rate):





### 2.1.3 Statistical Summary

Statistics for some columns in the train set:

	fico	aprv_flag	amtfinanced_1req	pti_1req	\
count	21431.000000	21606.000000	21606.000000	21603.000000	
mean	703.643087	0.738313	29870.867118	9.025000	
std	82.786470	0.439563	15311.300550	4.803567	
min	372.000000	0.000000	0.000000	0.000000	
25%	644.000000	0.000000	19370.000000	5.930000	
50%	701.000000	1.000000	26806.000000	8.590000	
75%	766.000000	1.000000	36931.250000	11.580000	
max	894.000000	1.000000	189729.000000	207.090000	

	ltv_1req	p12_reh7120_a	p12_bcx7110_a	p12_all7170_a	\
count	21601.000000	19694.000000	17917.000000	20943.000000	
mean	101.188938	51.866406	35.863370	3.597288	
std	23.245966	37.352331	33.225946	14.888786	
min	10.350000	0.000000	0.000000	0.000000	
25%	90.520000	16.000000	6.000000	0.000000	
50%	103.470000	52.000000	26.000000	0.000000	
75%	113.800000	88.000000	63.000000	0.000000	
max	955.260000	415.000000	290.000000	100.000000	

	p12_aut7110_a	p12_all7120_a	...	p12_alm6200_a	p12_all6971_a	\
count	11070.000000	21226.000000	...	21606.000000	21562.000000	
mean	66.256459	85.351362	...	149.763445	58.454271	
std	24.470168	37.630013	...	181.349594	133.394966	
min	0.000000	0.000000	...	1.000000	0.000000	
25%	50.000000	73.000000	...	1.000000	1.000000	
50%	72.000000	94.000000	...	30.000000	1.000000	
75%	86.000000	100.000000	...	400.000000	1.000000	
max	152.000000	711.000000	...	400.000000	400.000000	

	clntr9437_a	clact9429_a	clall5010_a	clall2434_a	cloil0214_a	\
count	8952.000000	8952.000000	2390.000000	719.000000	78.000000	
mean	1.391309	2.264187	2427.658996	0.020862	0.217949	
std	3.433233	6.929704	4421.540329	0.143023	0.415525	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	0.000000	
75%	1.000000	2.000000	2798.250000	0.000000	0.000000	
max	73.000000	178.000000	33549.000000	1.000000	1.000000	

Statistics for some columns in the test set:

	fico	aprv_flag	amtfinanced_1req	pti_1req	ltv_1req	\
count	5343.000000	5400.000000	5400.000000	5398.000000	5398.000000	
mean	702.706719	0.741481	29782.550317	9.114783	101.516121	
std	82.291798	0.437861	14843.983207	4.375332	22.239436	
min	416.000000	0.000000	3786.000000	0.260000	12.350000	
25%	644.000000	0.000000	19638.000000	5.990000	91.170000	
50%	701.000000	1.000000	26894.000000	8.710000	103.675000	
75%	765.000000	1.000000	36946.500000	11.687500	113.705000	
max	893.000000	1.000000	134500.000000	79.270000	304.230000	

	p12_reh7120_a	p12_bcx7110_a	p12_all7170_a	p12_aut7110_a	\
count	4916.000000	4442.000000	5206.000000	2788.000000	
mean	51.010985	35.400495	3.892816	66.228121	
std	37.081701	33.378023	15.486130	24.474930	
min	0.000000	0.000000	0.000000	0.000000	
25%	15.000000	5.000000	0.000000	51.000000	
50%	51.000000	24.000000	0.000000	71.000000	
75%	88.000000	63.000000	0.000000	86.000000	
max	279.000000	154.000000	100.000000	104.000000	

	p12_all7120_a	...	p12_alm6200_a	p12_all6971_a	clntr9437_a	\
count	5295.000000	...	5400.000000	5385.000000	2272.000000	
mean	85.011331	...	150.206481	59.554875	1.428257	
std	39.007500	...	181.471627	134.574791	3.305008	
min	0.000000	...	1.000000	0.000000	0.000000	
25%	71.000000	...	1.000000	1.000000	0.000000	
50%	94.000000	...	30.000000	1.000000	0.000000	
75%	100.000000	...	400.000000	1.000000	2.000000	
max	603.000000	...	400.000000	400.000000	61.000000	

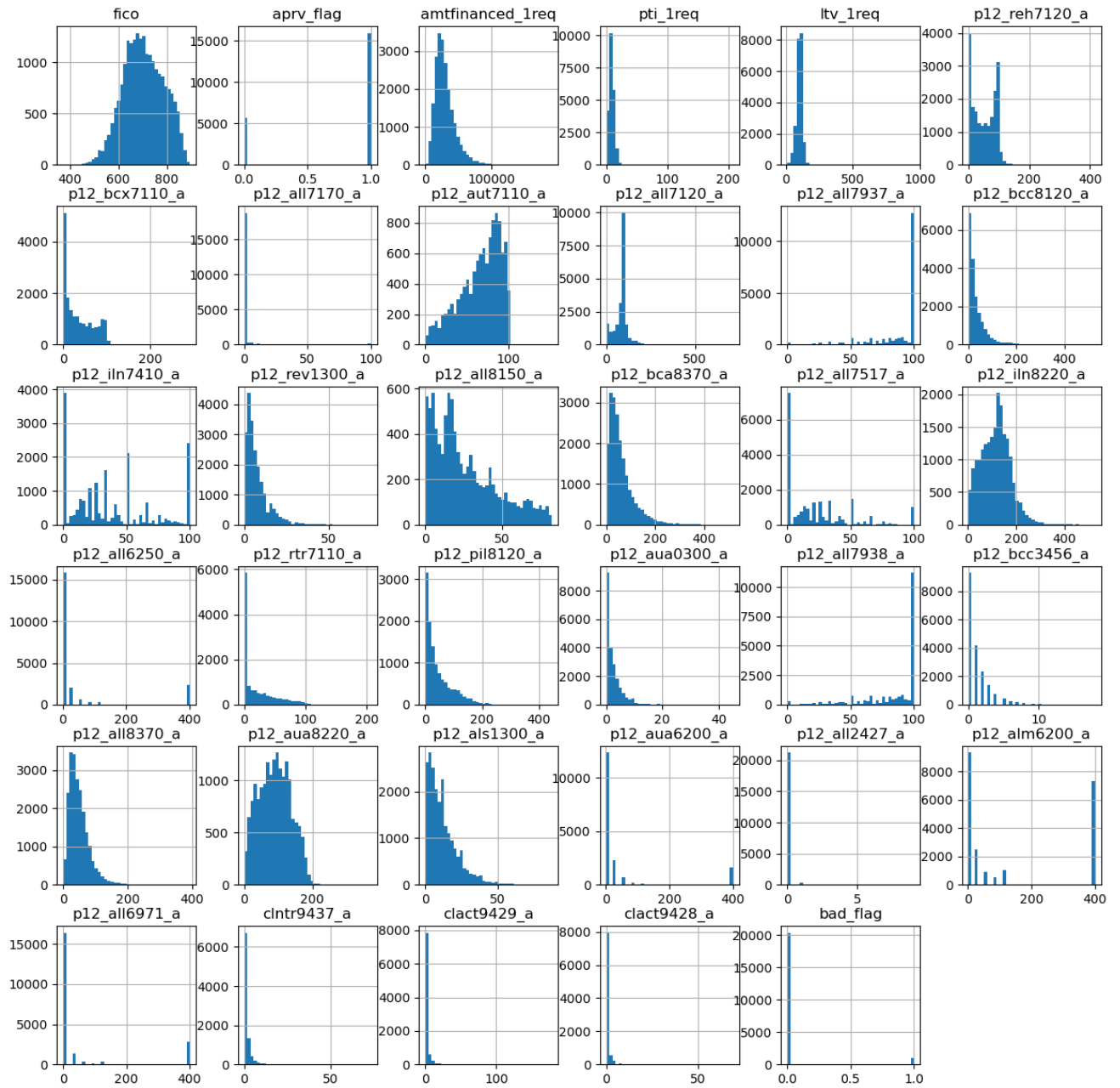
  

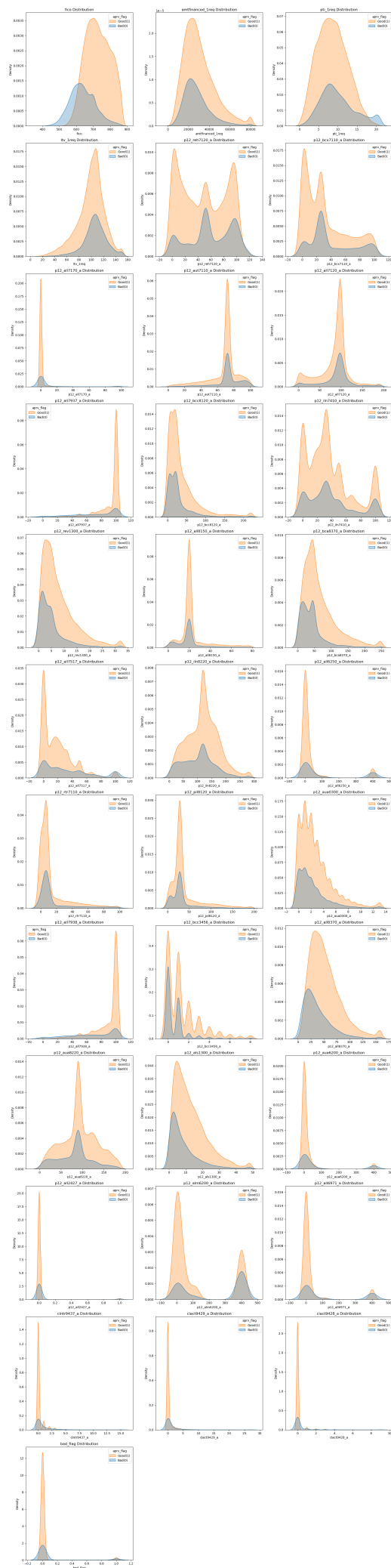
	clact9429_a	clall5010_a	clall2434_a	cloil0214_a	cltra4405_a	\
count	2272.000000	607.000000	200.000000	18.000000	203.000000	
mean	2.311620	2902.642504	0.030000	0.333333	0.004926	
std	6.601955	5030.618640	0.198233	0.485071	0.070186	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	0.000000	
75%	2.000000	3885.000000	0.000000	1.000000	0.000000	
max	117.000000	38755.000000	2.000000	1.000000	1.000000	

### 3 Visualization

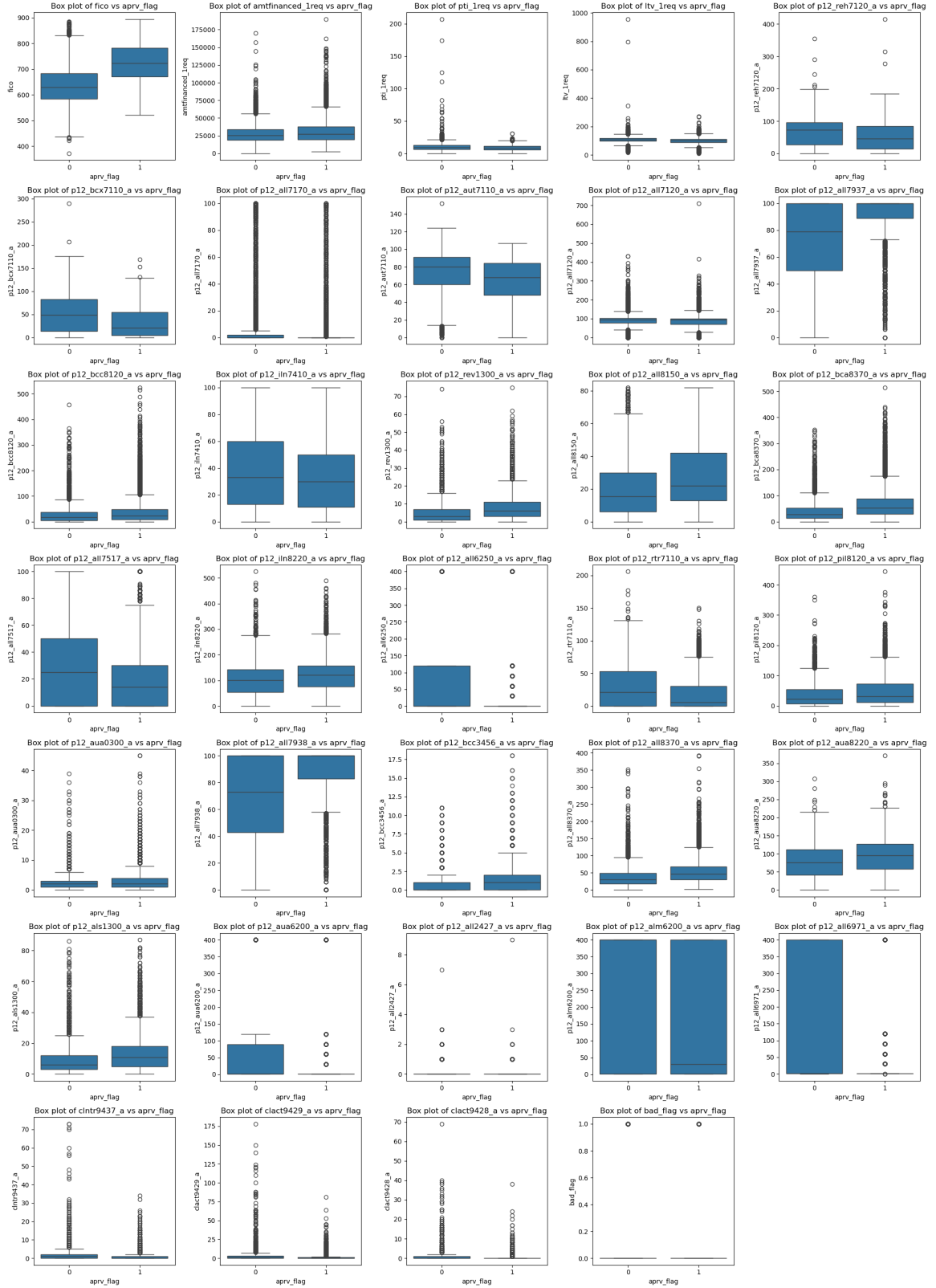
#### 3.1 Univariate Analysis

##### 1. Histogram and KDE plots





## 2. Box plots

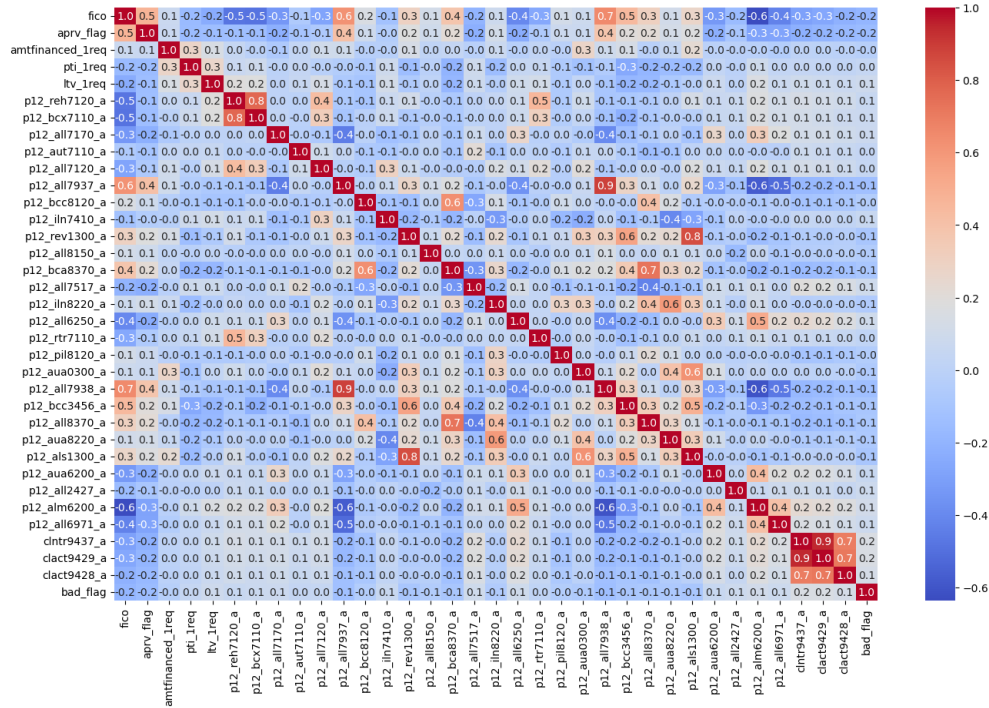


## 3. Log Transformation of Skewed Features:

We aimed to address features with significant skewness in their distribution. Skewed features can negatively impact the performance of certain machine learning models, which assume more symmetric, normal distributions.

## 3.2 Bivaiaate Analysis

### 1. Correlation Heatmap



We removed the features that have more than 0.8 correlation value.

### 3.3 Outliers Detection

Using the above boxplots, we saw that there is a lot of outliers. To handle them the values in numerical features were capped at the 99th percentile.

## 4 Data Cleaning and Pre-Processing

### 4.1 Handling Missing Values

1. Drop features that have more than 80% missing data.
2. For categorical features, we used SimpleImputer using the most frequent strategy.
3. For numerical features, we used SimpleImputer using the median strategy.

### 4.2 Scaling

Scaling Method: RobustScaler was chosen to transform numerical features, which is effective in handling outliers by using the median and interquartile range (IQR) instead of mean and standard deviation.

### 4.3 Encoding Categorical Variables

1. One-hot encoding was applied using `pd.get_dummies()`, creating binary indicator variables for each category in these columns.
2. The parameter `drop_first=False` ensures that all categories are retained, preventing loss of information.
3. Newly generated dummy variables were explicitly converted to integers to ensure compatibility with machine learning models.

### 4.4 Train\_Test\_Split

The dataset was split into 80% training data and 20% testing data