Exploratory Data Analysis: Home Loan Default Data

By: Denis O'Byrne

Date: 3/10/2022

IBM Exploratory Data Analysis for Machine Learning Final Project

Introduction

Banks earn a majority of their revenue from lending loans. However, this is often associated with risk. The borrowers may default on the loan. To mitigate this issue, the banks have decided to use Machine Learning to overcome this issue. They have collected past data on the loan borrowers & would like to develop a strong ML Model to classify if any new borrower is likely to default or not.

Data Set Description

For this assignment I will be analyzing the Loan Default Dataset from Kaggle seen here. The data contains 148670 observations with 33 columns of features per observation along with a labeled categorical status (1 = paid, 0 = defaulted). The data specifically seems to cover home loans as many of the features included in the data are only relevant to home loans, so I will be analyzing the data under the assumption that these are home loans although there is no information provided to definitively confirm this.

Note that the dataset does not provide descriptions for the variables so the understanding of the variables and factors is not clear in some cases, but I will still provide a detailed description of the variables based on my understanding of the dataset.

Target:

Status (Categorical)- 1 / 0 - Each datapoint is labeled either 1 for paid or 0 for defaulted

Features:

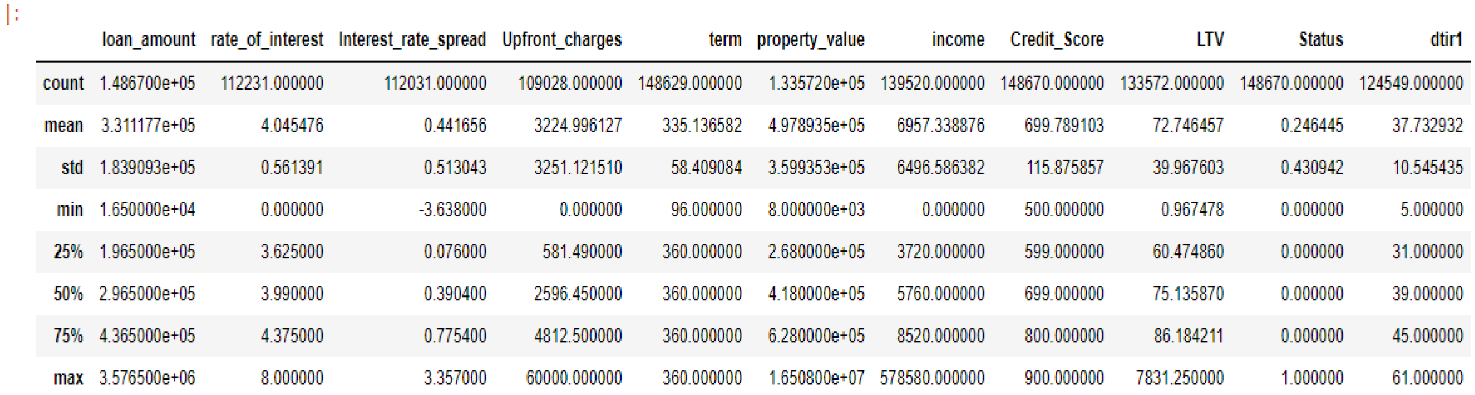
1. ID (Integer) - Row labels for the data from the source. 24890 – 173559 (this column is not important and will be removed)
2. Year (Integer) - Year the Loan payment was due. All loans in this dataset are from 2019 so this column should be removed as it provides no information
3. Loan_Limit (Categorical) - cf /ncf - meaning conforming or nonconforming. A conforming loan is a loan that cannot exceed a specified amount. Therefore, a loan limit exists for a cf loan and there is no limit on a ncf loan. We do not know what the limit is, just that it exists.
4. Gender (Categorical) - Male/Female/Joint/Sex Not Available - A Joint loan is a loan with multiple cosigners not necessarily of mixed gender.
5. Approv_In_Adv (Categorical) - pre/nopre - A preapproved loan is a loan that was given by a bank prior to making a purchase to let the buyer know how much they can afford. A non-preapproved loan is a loan is for a buyer who made a purchase or found a price for an item prior to speaking with a bank or lender.
6. Loan_Type (Categorical) - type1/type2/type3 - no information on what these types mean is provided
7. Loan_Purpose (Categorical) - p1/p2/p3/p4 - no information on what these purposes are
8. Credit_Worthiness (Categorical) – L1/L2 - no information on what this means. I think it refers to habitual defaulters.

9. Open_Credit (Categorical) - nopc/opc - Do they have an open monthly personal credit (opc) card/line with the bank or not?
10. Business_Or_Comercial (Categorical) – nob/c or b/c - Is this a business or personal expense
11. Loan_Amount (Continuous) - Dollar amount of the original loan. Range (16,500 – 32,576,500)
12. Rate_Of_Interest (Continuous) - Percentage of interest accrued annually. Range (0.00 - 8.00)
13. Interest_Rate_Spread (Continuous) - The net interest rate spread is the difference between the interest rate a bank pays to depositors and the interest rate it receives from loans to consumers. Range (-3.638 - +3.357). A negative spread implies the loan holder could deposit the loan in the bank and earn money on the interest faster than the loan grows with interest.
14. Upfront_Charges (Continuous) - Down payments and fees associated with the loan. Range (0.00 - 60,000.00)
15. Term (Integer)- Days Until the Loan is to be paid off in full. Range (96 - 365). Note more than 75% of the loans in this dataset have a loan term of 365. It Is more helpful to consider this as a categorical variable as there are only a few possible values for the loan terms.
16. Neg-Amortization (Categorical) - not_neg / neg_amm - An amortized loan is paid so that you pay more than the accrued interest at the end of each term. Amortized payments are higher but eventually the loan is paid off in full. Negative-amortized loans pay less than the accrued interest on the loan so that the amount owed on the loan is more at the end of each term. A loan will never be paid off if it remains in negative amortized status.
17. Interest_only (Categorical) - not_int / int_only - An interest-only loan is a loan in which the borrower pays only the interest for some or all of the term, with the principal balance unchanged during the interest-only period
18. Lump_Sum_Payment (Categorical) - not_lpsm / lpsm - A lump-sum payment is an often-large sum that is paid in one single payment instead of broken up into installments.
19. Property_Value (Continuous)- Value of the building being purchased without other fees. Range (8,000 - 16,508,000)
20. Construction_Type (Categorical) - sb / mh - I don’t know what these mean
21. Occupancy_Type (Categorical) - pr / sr/ ir - I don’t know what these mean
22. Secured_By (Categorical) - home / land
23. Total_Units (Categorical) - 1U / 2U / 3U / 4U – number of units refers to how many separate families can live in the home
24. Income (Continuous) - How much does the property earn in the term of the loan. If this is a personal home then the value should be 0. If the property is being rented then it should be nonzero. Range (0 – 578,580)
25. Credit_Type (Categorical) - EXP / EQUI / CRIF / CIB – Credit rating type
26. Credit_Score (Integer) - Credit Score of Applicant at the time of the loan. Range (500-900)
27. Co-Applicant-Credit-Type (Categorical) - CIB / EXP
28. Age (Categorical) - < 25 / 25-34 / 35-44 / 45-54 / 55-64 / 65-74 / > 74 – Age range of loan applicant
29. Submission_Of_Application (Categorical) - to_inst / not_inst – to inst means they submitted
30. LTV (Continuous) - The loan-to-value ratio is a financial term used by lenders to express the ratio of a loan to the value of an asset purchased. In Real estate, the term is commonly used by banks and building societies to represent the ratio of the first mortgage line as a percentage of the total appraised value of real property. A higher LTV is better for the buyer since they are getting a better deal. Range (0.967478 - 7831.25)
31. Region (Categorical) - South / North / Central / North-East – Region of the US where the Applicant lives

32. Security_Type (Categorical) - direct / indirect – not sure what this means
33. Dtir1 (Continuous) - Debt to income ratio percentage. Your debt-to-income ratio is all your monthly debt payments divided by your gross monthly income. Range (5.00 - 61.00)

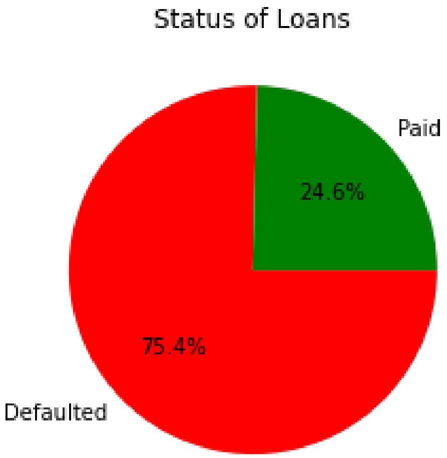
Exploratory Data Analysis and Data Cleaning

For a better understanding of the data set we will begin looking at the data visually. To start we can see the summary statistics for the numeric features omitting the year and ID columns



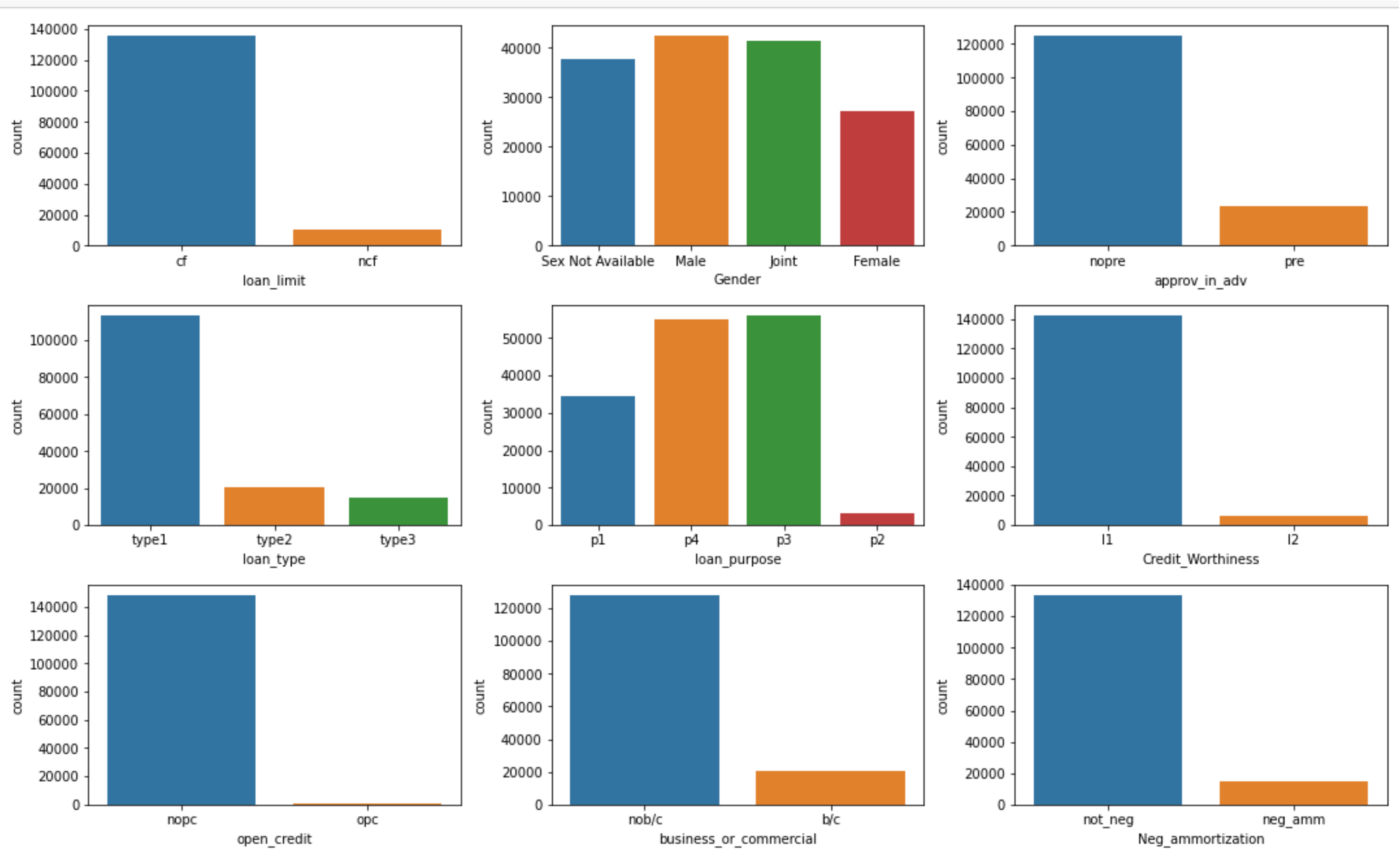
	loan_amount	rate_of_interest	Interest_rate_spread	Upfront_charges	term	property_value	income	Credit_Score	LTV	Status	dtir1
count	1.486700e+05	112231.000000	112031.000000	109028.000000	148629.000000	1.335720e+05	139520.000000	148670.000000	133572.000000	148670.000000	124549.000000
mean	3.311177e+05	4.045476	0.441656	3224.996127	335.136582	4.978935e+05	6957.338876	699.789103	72.746457	0.246445	37.732932
std	1.839093e+05	0.561391	0.513043	3251.121510	58.409084	3.599353e+05	6496.586382	115.875857	39.967603	0.430942	10.545435
min	1.650000e+04	0.000000	-3.638000	0.000000	96.000000	8.000000e+03	0.000000	500.000000	0.967478	0.000000	5.000000
25%	1.965000e+05	3.625000	0.076000	581.490000	360.000000	2.680000e+05	3720.000000	599.000000	60.474860	0.000000	31.000000
50%	2.965000e+05	3.990000	0.390400	2596.450000	360.000000	4.180000e+05	5760.000000	699.000000	75.135870	0.000000	39.000000
75%	4.365000e+05	4.375000	0.775400	4812.500000	360.000000	6.280000e+05	8520.000000	800.000000	86.184211	0.000000	45.000000
max	3.576500e+06	8.000000	3.357000	60000.000000	360.000000	1.650800e+07	578580.000000	900.000000	7831.250000	1.000000	61.000000

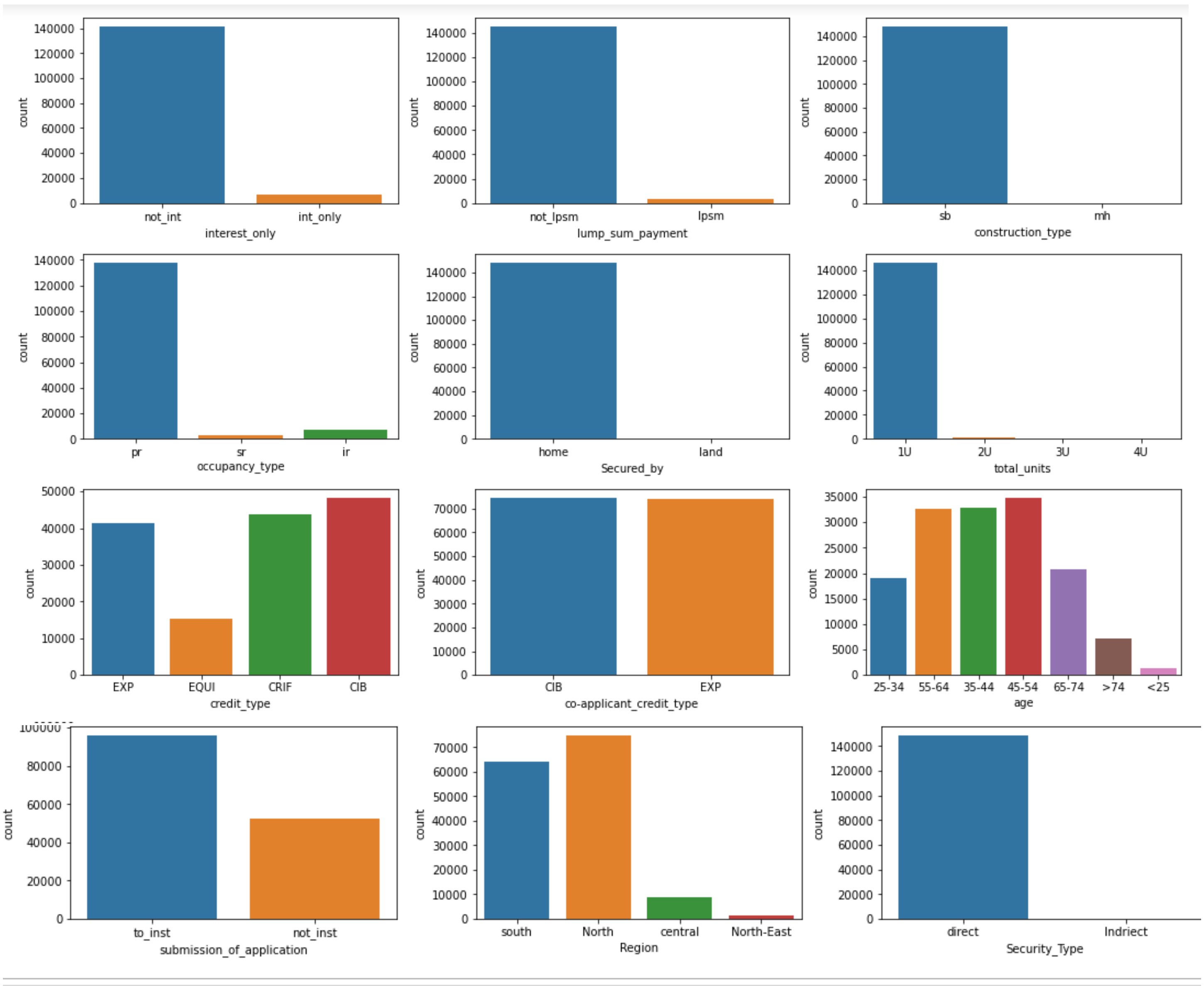
Most of the information here is no more informative than what I had provided in the original data description for the ranges however we can see that the status label for the majority of loans in this data set is defaulted, with only 24.6% of loans being paid. We can see this better in a pie chart.



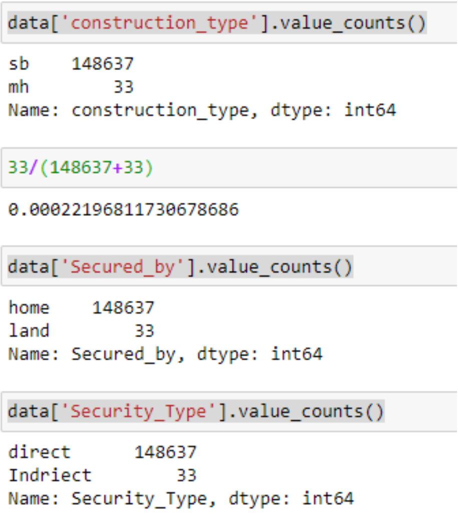
Although in most cases this imbalance of classes would seem bad for building a predictive model, we need to recognize that in the case of loan defaults we would like to only lend out to customers who we know will pay back the loan, so by oversampling the default class, we can help any model we build in the future spot more indicators of a defaulter. Any model built on this data should be optimized to improve sensitivity to default instead of specificity of paid loans or total accuracy, so that the bank does not make risky loans.

We can view bar charts to see the distributions of the other categorical variables in the data set as well.





Looking at the splits in the data here we can see that we have extreme class imbalances in many of the features meaning they will almost surely be unhelpful in developing a predictive model as the information gain from using these features such as Security_Type, Construction_Type, or Secured_By to develop a model will be minute and any information gleaned from these features will not extrapolate to new data as our sample size is too small. For now I will leave them in, but we can get even more information on these 3 features specifically by looking at the exact counts in the groups for these data types.



```
data['construction_type'].value_counts()
```

```
sb    148637
mh        33
Name: construction_type, dtype: int64
```

```
33/(148637+33)
```

```
0.00022196811730678686
```

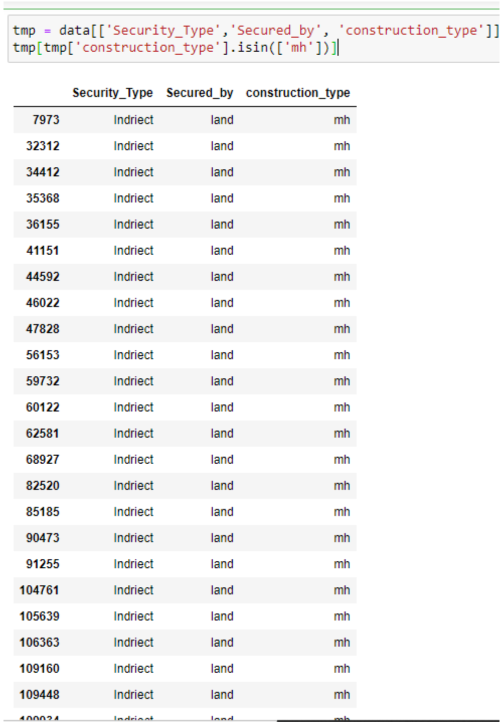
```
data['Secured_by'].value_counts()
```

```
home    148637
land        33
Name: Secured_by, dtype: int64
```

```
data['Security_Type'].value_counts()
```

```
direct      148637
Indriect        33
Name: Security_Type, dtype: int64
```

As we can see above, the three variables seem to be indicators of the same observations in the data. In other words, if a loan is construction type mh, then it will be secured by land and security type indirect and if it is construction type sb, then it will be secured by home and security type will be direct.

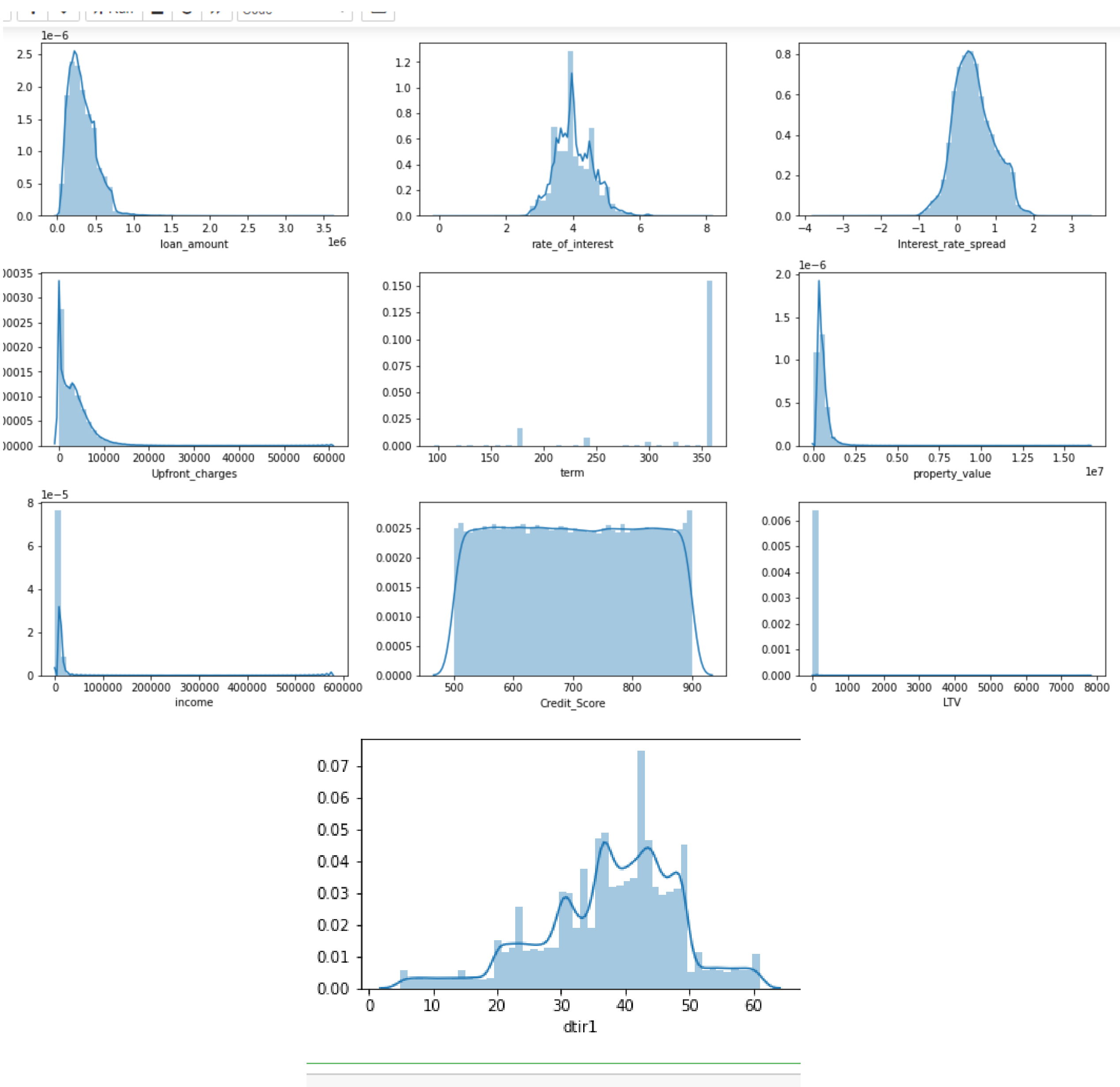


```
tmp = data[['Security_Type','Secured_by', 'construction_type']]
tmp[tmp['construction_type'].isin(['mh'])]
```

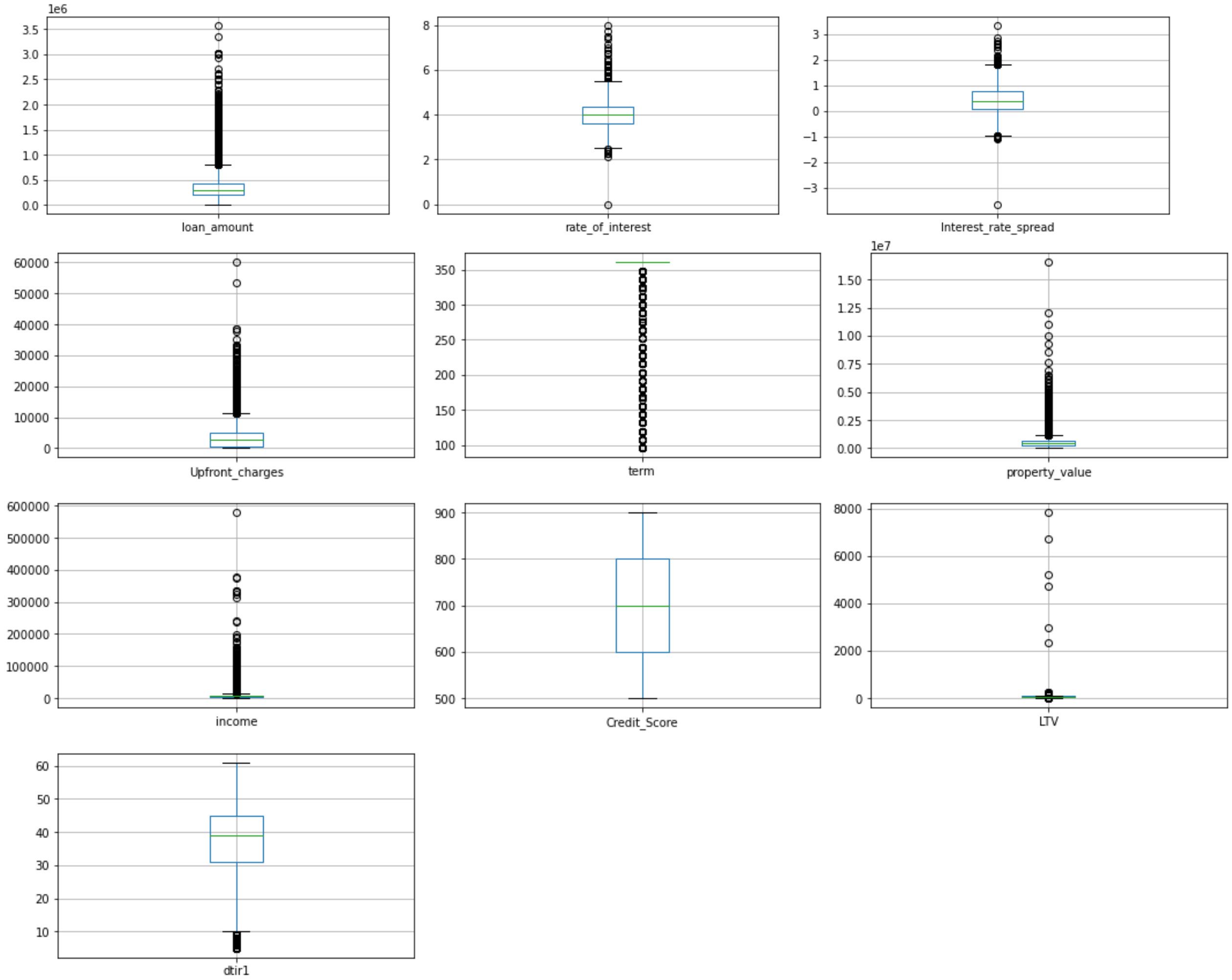
	Security_Type	Secured_by	construction_type
7973	Indriect	land	mh
32312	Indriect	land	mh
34412	Indriect	land	mh
35368	Indriect	land	mh
36155	Indriect	land	mh
41151	Indriect	land	mh
44592	Indriect	land	mh
46022	Indriect	land	mh
47828	Indriect	land	mh
56153	Indriect	land	mh
59732	Indriect	land	mh
60122	Indriect	land	mh
62581	Indriect	land	mh
68927	Indriect	land	mh
82520	Indriect	land	mh
85185	Indriect	land	mh
90473	Indriect	land	mh
91255	Indriect	land	mh
104761	Indriect	land	mh
105639	Indriect	land	mh
106363	Indriect	land	mh
109160	Indriect	land	mh
109448	Indriect	land	mh
109934	Indriect	land	mh

Checking this we see this assumption is true and so we can at least drop two of these columns. I will leave the secured_by column as it is the easiest column to understand as a variable.

Lastly, we can investigate the numeric features using plots of the distribution functions of the data



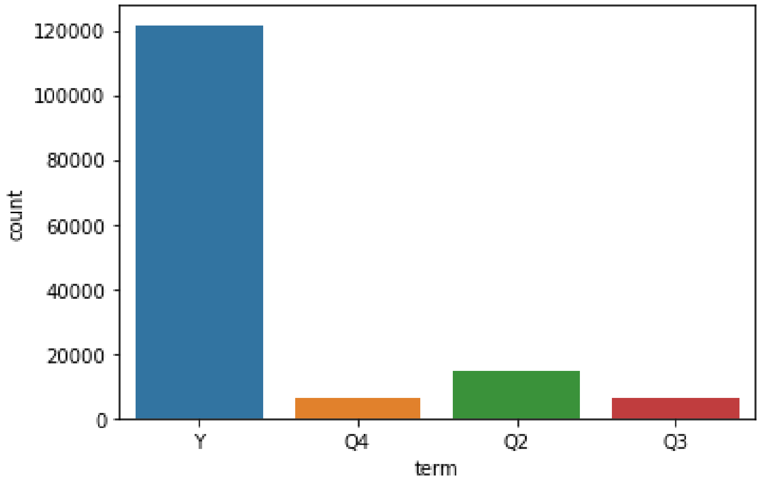
Here we can see the sample probability mass functions with the overlaid maximum likelihood density of each of the numeric variables. Here we can see that none of these variables follow a normal distribution except possibly the rate of interest and the interest rate spread. Credit score appears to follow a uniform distribution between 500 and 900. Dtir1 appears to follow a right skewed heavy tailed distribution. The Term looks to be discrete however inspecting further there do appear to be continuous possible values, just a majority fall on certain dates. The rest of the variables follow some form of exponential distributions. We can observe box plots to see if any outliers exist.



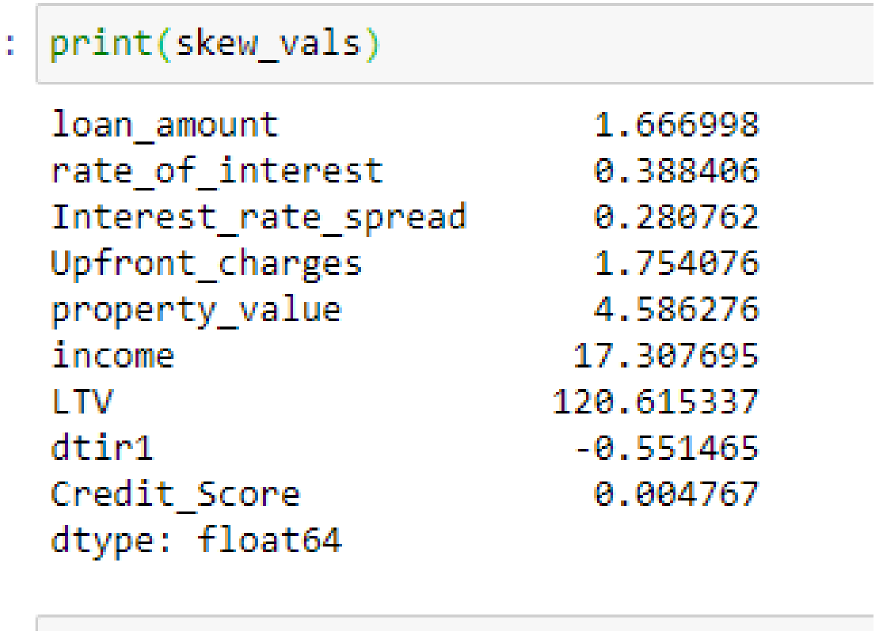
From these plots we can see that each variable besides Credit score is plagued with outliers and it is more obvious now that most of these variables are skewed to the right or left meaning they would be well suited for a log transform.

Data Cleaning

Also, I would like to restructure the term variable into a categorical variable, splitting the data into 5 groups, separating the year into quarters such that terms between 0-90 days are in group Q1, 91-180 days are in group Q2, 181-240 days are in group Q3, 241-359 days are in group Q4, and 360 days exclusively are in group Y for Year. Below is the histogram of the term data after this transformation.



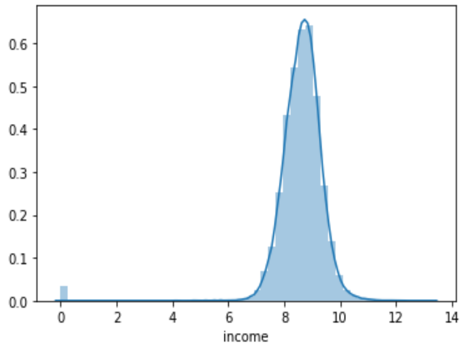
To see which variables need the log transform, we can check the skewness of the numeric variables to find the following skewness values:

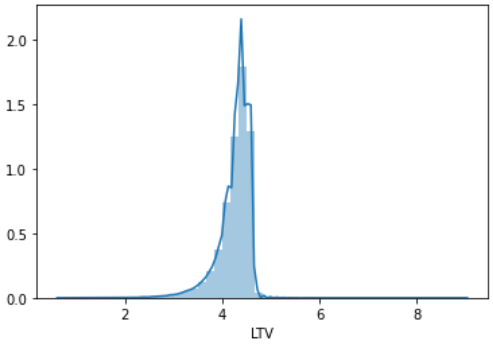


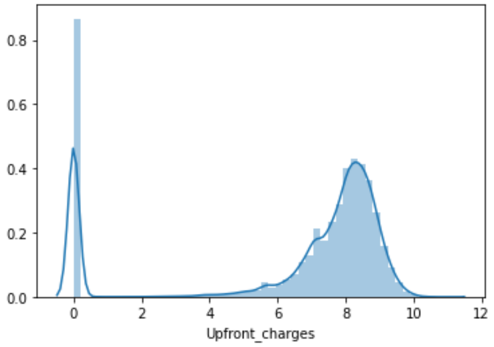
```
: print(skew_vals)

  loan_amount                1.666998
  rate_of_interest           0.388406
  Interest_rate_spread       0.280762
  Upfront_charges            1.754076
  property_value             4.586276
  income                    17.307695
  LTV                      120.615337
  dtir1                     -0.551465
  Credit_Score               0.004767
  dtype: float64
```

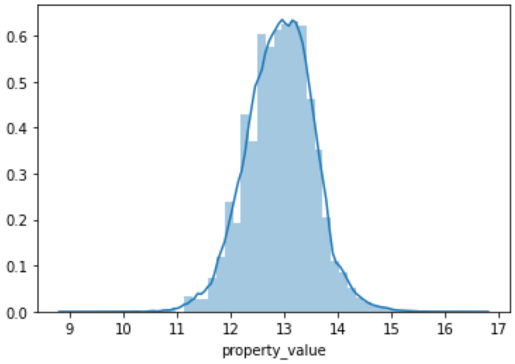
Following the recommendations of this course I only applied the log transformation to the features with a skewness above 0.75, which applies to loan_amount, Upfront_charges, property_value, income, and LTV.

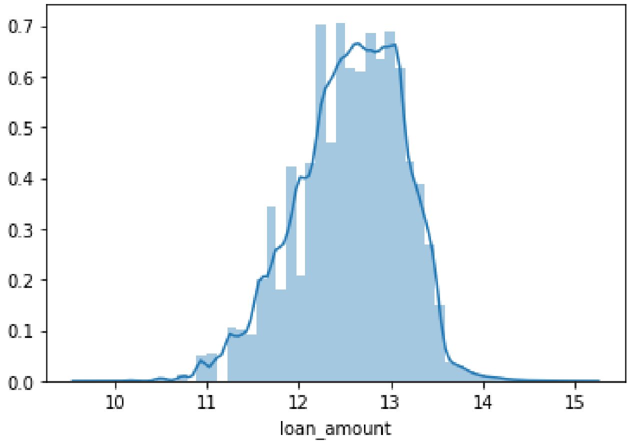
Following the transformation, we find that the data looks much more normally distributed with the possible exception of zero inflation.





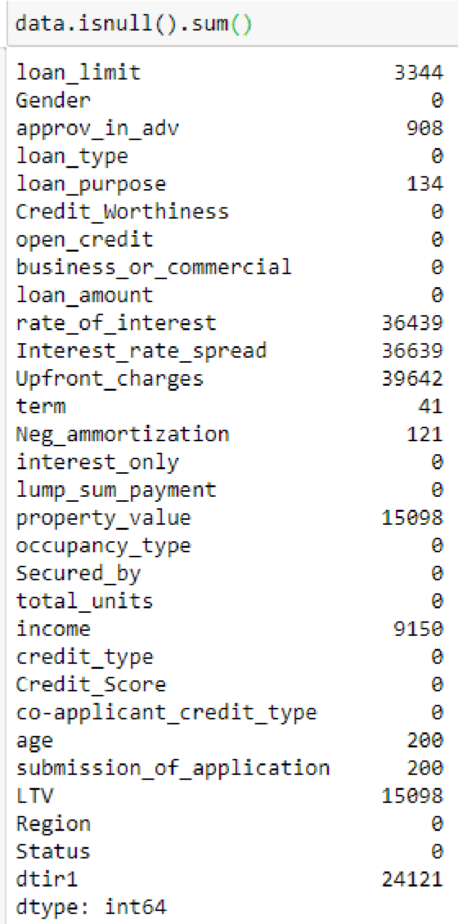






Imputing Missing Values

Now that the data is properly transformed, we can impute missing values. First, we can look at the number of missing values for each variable presented below:



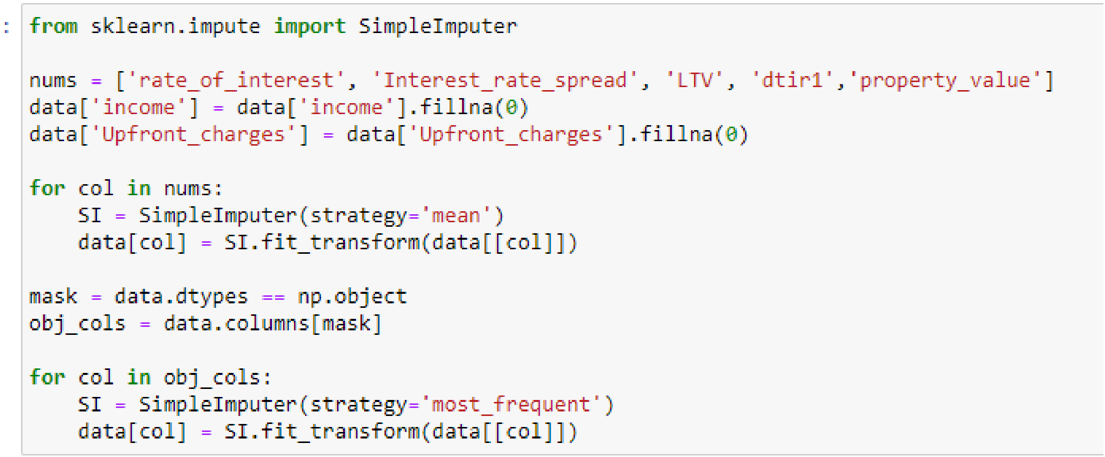
```
data.isnull().sum()
```

```
loan_limit                    3344
Gender                           0
approv_in_adv                  908
loan_type                        0
loan_purpose                   134
Credit_Worthiness                0
open_credit                      0
business_or_commercial           0
loan_amount                      0
rate_of_interest             36439
Interest_rate_spread         36639
Upfront_charges              39642
term                            41
Neg_ammortization              121
interest_only                    0
lump_sum_payment                 0
property_value               15098
occupancy_type                   0
Secured_by                       0
total_units                      0
income                        9150
credit_type                      0
Credit_Score                     0
co-applicant_credit_type         0
age                            200
submission_of_application      200
LTV                          15098
Region                           0
Status                           0
dtir1                        24121
dtype: int64
```

We see that for the categorical variables we have at most 3344 missing variables which accounts for 2% of total observations but most are much less than this, so for convenience I will impute the most frequent value. We also see that the target value, Status has no missing values. Lastly, we see that the numeric variables have at minimum 9150 missing values or 6.15% of the observations for the income feature all the way up to 39642 missing values or 26.67% of the observations for the Upfront_charges feature. For these numeric features we should be more cautious of what value we impute.

For the variables income and Upfront_charges, I will impute the value 0 for missing data, as it is the most frequent value for the Upfront_charges and I am making the assumption that anyone who did not fill out the income for a property left it blank for the purpose that the property will not be used for income. Meanwhile for the rate_of_interest, Interest_rate_spread, LTV, dtir1, and property_value features I will impute the mean of the data as we have transformed the data to be somewhat normalized for these variables, making the mean an acceptable average observation for the sample date on these features.

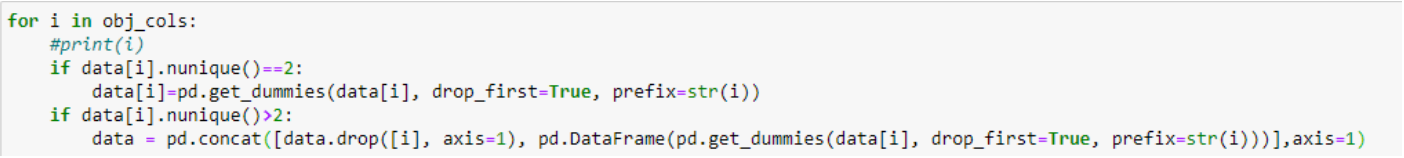
We can achieve these imputation specifications using the following code:



```
: from sklearn.impute import SimpleImputer

  nums = ['rate_of_interest', 'Interest_rate_spread', 'LTV', 'dtir1','property_value']
  data['income'] = data['income'].fillna(0)
  data['Upfront_charges'] = data['Upfront_charges'].fillna(0)

  for col in nums:
      SI = SimpleImputer(strategy='mean')
      data[col] = SI.fit_transform(data[[col]])

  mask = data.dtypes == np.object
  obj_cols = data.columns[mask]

  for col in obj_cols:
      SI = SimpleImputer(strategy='most_frequent')
      data[col] = SI.fit_transform(data[[col]])
```

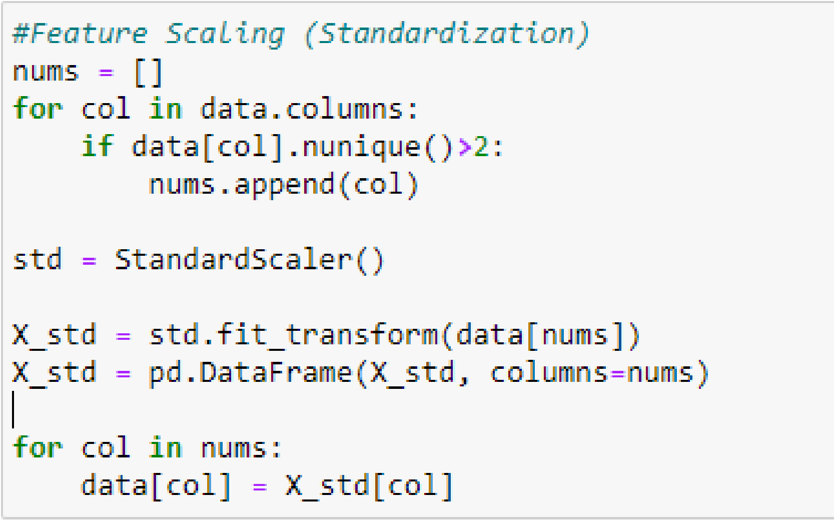
Standard Scaling and One Hot Encoding:

For one hot encoding we convert all factors for each feature to their own dummy variable with values of 1 or 0 to indicate if the observation belongs to the group. We drop 1 factor from each feature as it will be redundant since we can equally say an observation is not in all other groups for that feature. We do this with the following lines of code:



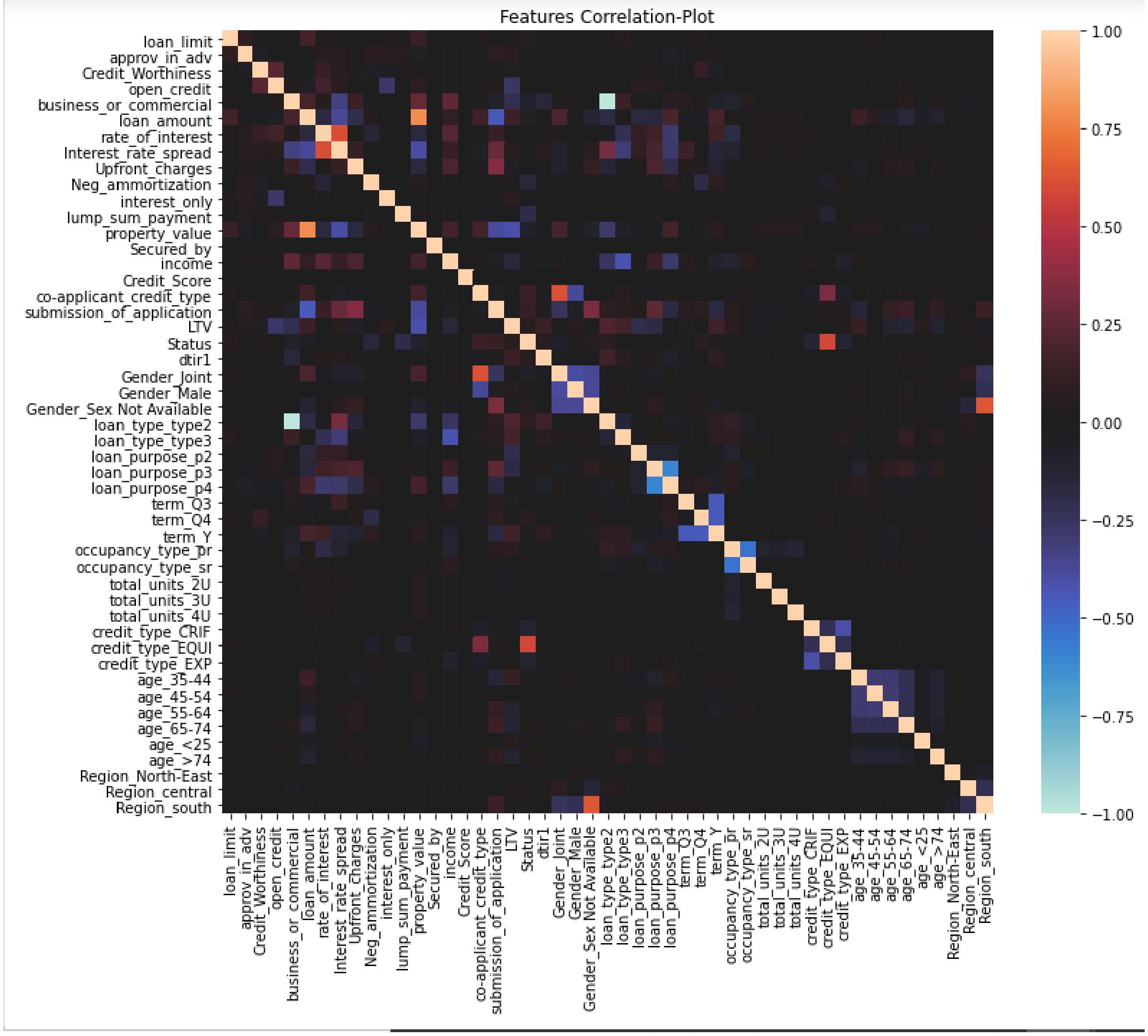
```
for i in obj_cols:
    #print(i)
    if data[i].nunique()==2:
        data[i]=pd.get_dummies(data[i], drop_first=True, prefix=str(i))
    if data[i].nunique()>2:
        data = pd.concat([data.drop([i], axis=1), pd.DataFrame(pd.get_dummies(data[i], drop_first=True, prefix=str(i)))],axis=1)
```

Lastly for feature engineering we apply the Standard Scalar to the data to standardize the scales on the data for future modeling. We do this only to features which are not binary. This can be accomplished with the following code:



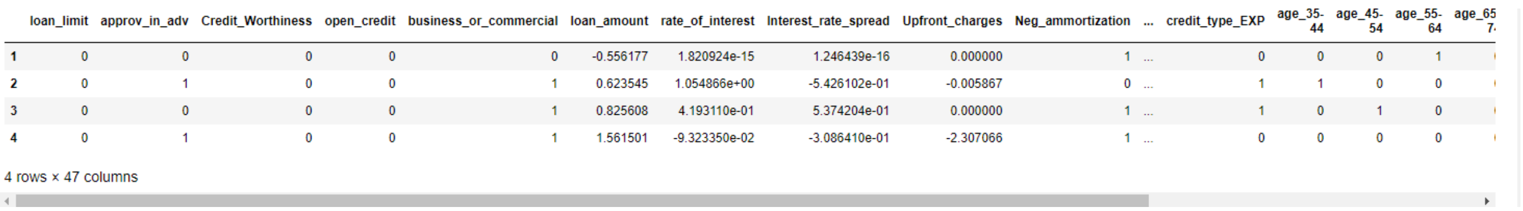
```
#Feature Scaling (Standardization)
nums = []
for col in data.columns:
    if data[col].nunique()>2:
        nums.append(col)

std = StandardScaler()

X_std = std.fit_transform(data[nums])
X_std = pd.DataFrame(X_std, columns=nums)

for col in nums:
    data[col] = X_std[col]
```

We can now look at the correlation heatmap of the data below.



Here we can see that there is some multi-correlation in our features and we also find that the feature buisness_or_commercial is the exact opposite indicator as a loan_type of type 2 meaning we can drop one of these two variables to avoid redundancy. We also see that rate_of_interest and interest_rate_spread are highly corelated with an exact correlation of 0.6143209863866121. This makes sense that they would be correlated as the bank is more likely to have a strong or weak spread if they give out a strong or weak interest rate, but it is possible these two variables can have different effects on loan repayment so we should keep them in. Lastly the loan amount and the property value are highly correlated as the loan is meant to pay for the property, with an exact correlation of 0.7919049032238942. With this level of correlation, it is probably best practice to drop one of the variables. I will drop the property value for this as we are trying to detect what makes people not pay back their loans after all, so the loan amount is more important for understanding the data in this case.

At this point the data cleaning is complete, so let's take a look at the head of the dataset



	loan_limit	approv_in_adv	Credit_Worthiness	open_credit	business_or_commercial	loan_amount	rate_of_interest	Interest_rate_spread	Upfront_charges	Neg_ammortization	...	credit_type_EXP	age_35-44	age_45-54	age_55-64	age_65-7
1	0	0	0	0	0	-0.556177	1.820924e-15	1.246439e-16	0.000000	1	...	0	0	0	1	
2	0	1	0	0	1	0.623545	1.054866e+00	-5.426102e-01	-0.005867	0	...	1	1	0	0	
3	0	0	0	0	1	0.825608	4.193110e-01	5.374204e-01	0.000000	1	...	1	0	1	0	
4	0	1	0	0	1	1.561501	-9.323350e-02	-3.086410e-01	-2.307066	1	...	0	0	0	0	

4 rows × 47 columns

We can see that we have a total of 46 features left and we never removed any observations so we still have 148670 rows.

Testing Hypotheses About The Data

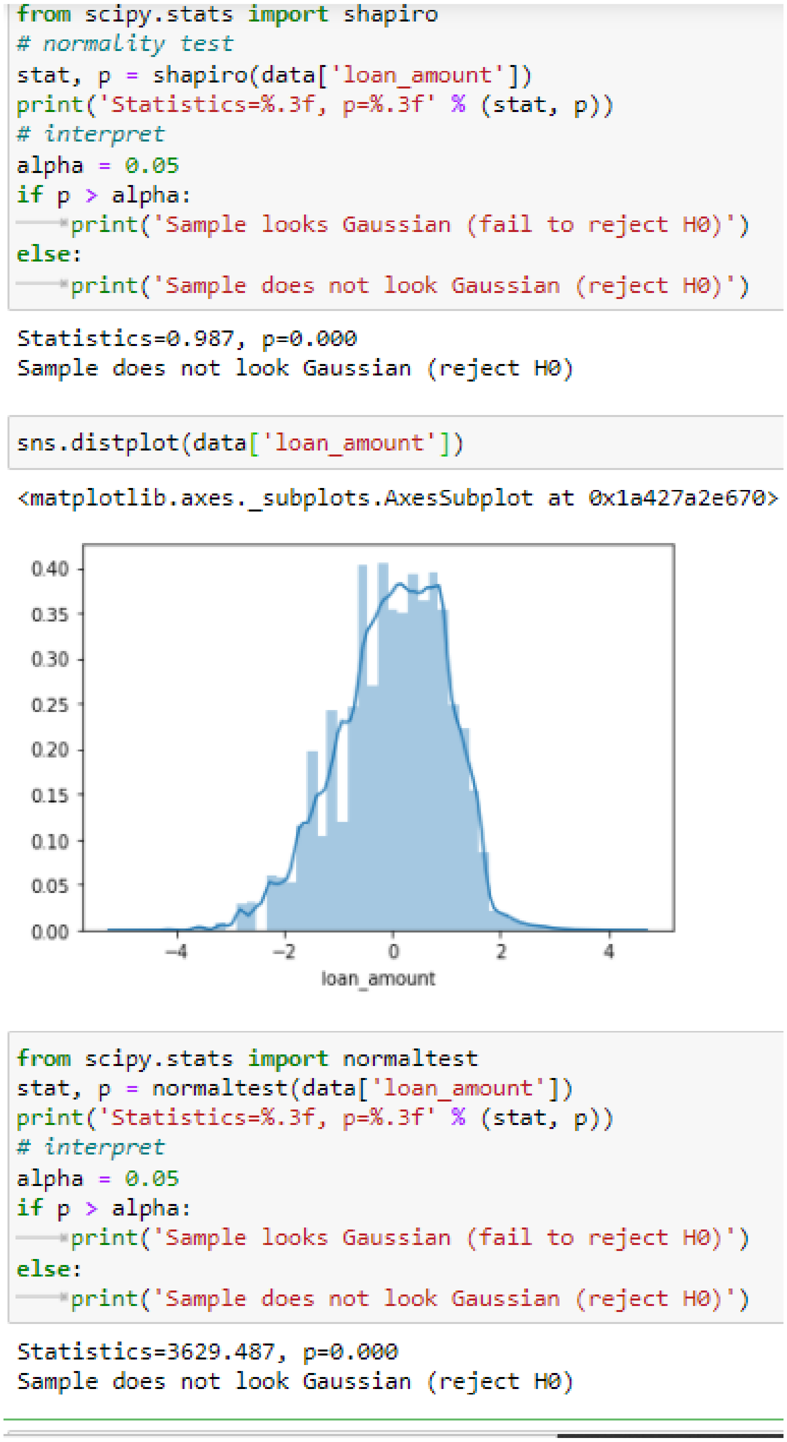
Hypothesis #1

We have been making some assumptions on the distributions of the features in the data set, so we should test these assumptions to ensure that the transformations made earlier actually normalized the data.

H_o: The loan_amount variable follows a Standard normal distribution after standard scaling and log transforming the data

H_a: The loan_amount variable does not follow a Standard normal distribution after standard scaling and log transforming the data

We can test this assumption using a Shapiro-Wilks test for normality or the D'Agostino's K^2 test. We will use an alpha = 0.05 level of significance for both tests.



```
from scipy.stats import shapiro
# normality test
stat, p = shapiro(data['loan_amount'])
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
    print('Sample looks Gaussian (fail to reject H0)')
else:
    print('Sample does not look Gaussian (reject H0)')
```

```
Statistics=0.987, p=0.000
Sample does not look Gaussian (reject H0)
```

```
sns.distplot(data['loan_amount'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a427a2e670>
```

```
from scipy.stats import normaltest
stat, p = normaltest(data['loan_amount'])
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
    print('Sample looks Gaussian (fail to reject H0)')
else:
    print('Sample does not look Gaussian (reject H0)')
```

```
Statistics=3629.487, p=0.000
Sample does not look Gaussian (reject H0)
```

We can see that the test statistic is too large in both tests leading to a p-value below 0.001. Since our p-value is below 0.05, we reject that the data for loan amounts is normally distributed at the alpha = 0.05 level of significance.

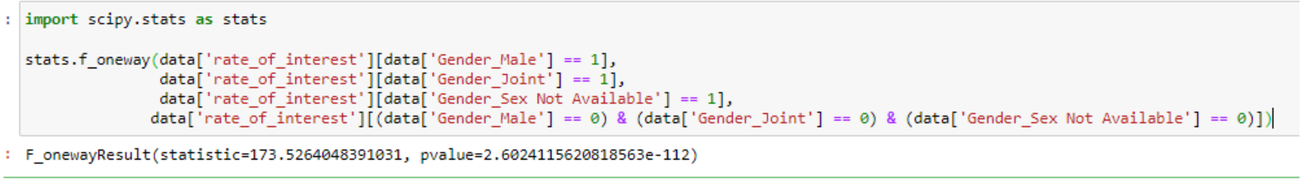
Hypothesis #2

We can test with the data if the gender of an applicant has an effect on the rate of interest for the loan. We may want to know if men or women are better at bargaining for terms in their loan agreements, or is it possible that couples/joint buyers receive better rates. For the specific test we can use a one-way Anova F test to see if the mean interest rate is equal for all groups. We can structure the test as follows:

H$_o$: $\mu_{Male} = \mu_{Female} = \mu_{Joint} = \mu_{Unspecified}$

H$_a$: At least one of the mean values is different from the others.

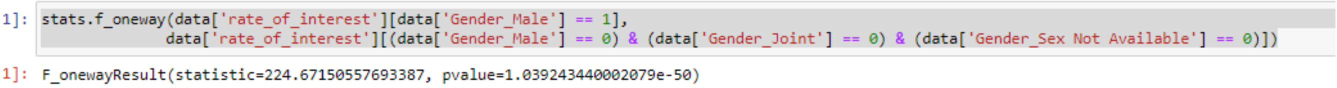
We again will run the test at the alpha = 0.05 level of significance



```
: import scipy.stats as stats

  stats.f_oneway(data['rate_of_interest'][data['Gender_Male'] == 1],
                 data['rate_of_interest'][data['Gender_Joint'] == 1],
                 data['rate_of_interest'][data['Gender_Sex Not Available'] == 1],
                data['rate_of_interest'][(data['Gender_Male'] == 0) & (data['Gender_Joint'] == 0) & (data['Gender_Sex Not Available'] == 0)])

: F_onewayResult(statistic=173.5264048391031, pvalue=2.6024115620818563e-112)
```

Here we see that that gender has a significant effect on the mean rate of interest for home loans with a p-value of 2*10^{-112} thus we reject the null hypothesis at any rational alpha level including our choice of 0.05

I decided to run the same test for just males and females as well below:



```
1]: stats.f_oneway(data['rate_of_interest'][data['Gender_Male'] == 1],
                   data['rate_of_interest'][(data['Gender_Male'] == 0) & (data['Gender_Joint'] == 0) & (data['Gender_Sex Not Available'] == 0)])

1]: F_onewayResult(statistic=224.67150557693387, pvalue=1.039243440002079e-50)
```

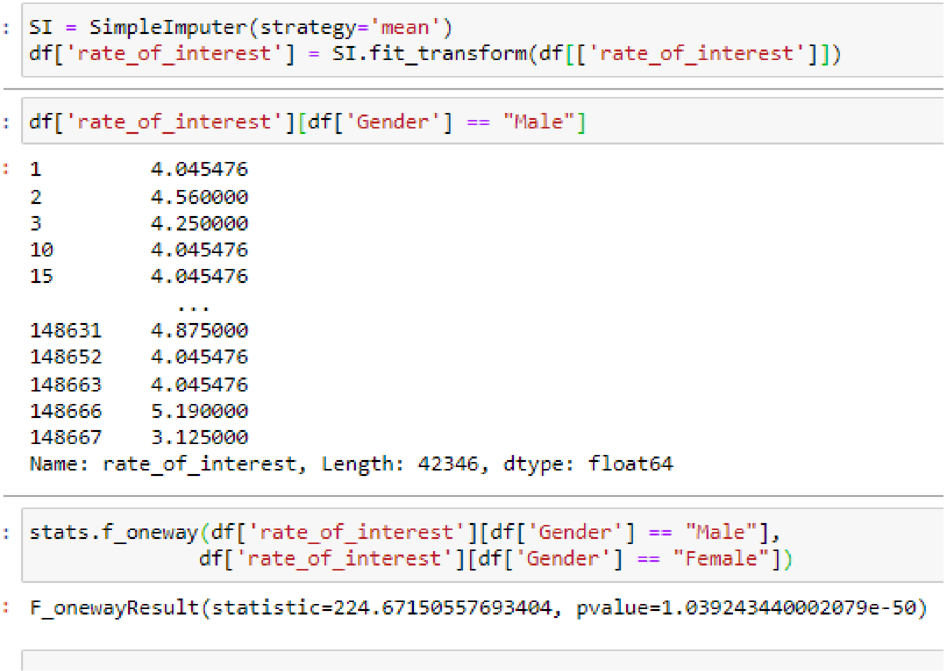
We see again that we reject the hypothesis that men and women receive equal loan interest rates with a p value of 1* 10^{-50} thus again we reject the null hypothesis at any rational alpha level including our choice of 0.05

We can see the sample mean and standard deviation for the standardized rate of interest on loans for males and females below. Recall that the data has been standardized prior to these calculations so the data is measuring the average Z score, not the average value.

Loan Interest Rate Z Score Statistics by Gender			
Gender	Sample Size	Sample Mean Z score	Sample Z score Standard Deviation
Male	42346	-0.010968703013856218	0.9917286229656858
Female	27266	0.1026491404400715	0.9516309894572063

We see that men receive loan interest rates relatively close to the total sample average while women usually get interest rates about 0.1 standard deviation higher than the average loan.

If we run the same test on the original data we get the same p-value as the standard scaling has a 1 to 1 correspondence as shown below, but we can look at the exact loan rates.



```
SI = SimpleImputer(strategy='mean')
df['rate_of_interest'] = SI.fit_transform(df[['rate_of_interest']])
```

```
df['rate_of_interest'][df['Gender'] == "Male"]
```

```
1         4.045476
2         4.560000
3         4.250000
10        4.045476
15        4.045476
            ...
148631    4.875000
148652    4.045476
148663    4.045476
148666    5.190000
148667    3.125000
Name: rate_of_interest, Length: 42346, dtype: float64
```

```
stats.f_oneway(df['rate_of_interest'][df['Gender'] == "Male"],
               df['rate_of_interest'][df['Gender'] == "Female"])
```

```
F_onewayResult(statistic=224.67150557693404, pvalue=1.039243440002079e-50)
```

We find the table for loan rates becomes as follows:

Loan Interest Rate Z Score Statistics by Gender			
Gender	Sample Size	Sample Mean Interest Rate	Sample Standard Deviation
Male	42346	4.040125682383571	0.48372803062761205
Female	27266	4.095544205542036	0.464169908737542

We see that the difference in interest rates between the sexes is very miniscule but the evidence from the One-way Anova test is overwhelmingly significant due to the sample sizes that there is a difference between the interest rates for men and women with women receiving slightly higher interest rates.

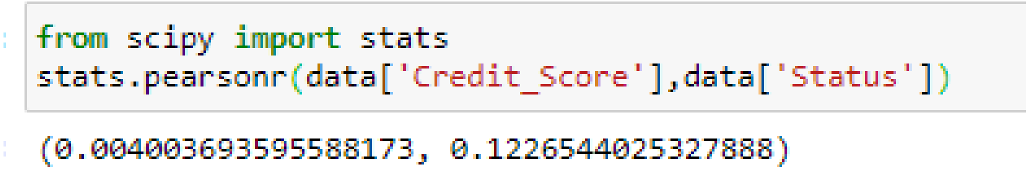
Hypothesis #3

Lastly, we know that a high credit rating indicates a person who pays their debts and bills on time regularly, so the credit score should be correlated with the status of loan so we should check using a Pearson correlation test we test this under the original assumption that there is no correlation.

H_o: The $r^2 = 0$ between Credit Score and Status

H_a: r^2 != 0 and there is a correlation between Credit Score and Status

We test this hypothesis with a Pearson Correlation Test at alpha = 0.05 level significance



```
from scipy import stats
stats.pearsonr(data['Credit_Score'],data['Status'])
```

```
(0.004003693595588173, 0.1226544025327888)
```

We see from the above test that there is a very miniscule correlation in this sample of 0.004 between the Credit Score and the default Status of the loan. From our intuition this seems nonsensical as the whole point of a credit score is to show if a person will pay their bills or loans back so that the banks know who they should lend to, but we need to remember that the purpose of this data set is to find reasons why people who passed loan acceptance decided not to pay. Therefore, our data includes tons of people with great credit scores who still defaulted which the bank did not expect and the bank wants to know what caused this. Thus, credit scores alone in this dataset should not be a good indicator of default since anyone we loaned money to is someone who we thought would pay their bills to begin with. That being said, we see that the p-value for this test is 0.1227 which is greater than 0.05 and thus we fail to reject the null hypothesis and conclude that there is no statistically significant correlation between Credit Score and Default Status.

Conclusion

After running through the exploratory data analysis, it was easy to understand the features provided even without a proper description provided by the source. Along with this we found the distribution of variables visually and transformed the data to be more normal through log transforms to reduce the number of outliers. We were also able to make appropriate assumptions and imputations of missing data for use in future predictive modeling. Also, we found and removed highly correlated and even duplicate features. Lastly, we were able to test some assumptions about the relation between variables to see how the data in this set goes against our initial hypotheses. We proved that our log transformation was not enough to normalize the dataset, and found that women receive loans with higher interest rates, and surprisingly Credit Scores are not a useful indicator of default status for this given data set.

Next Steps

In the future it would be useful to apply some feature reduction techniques such as Principal component Analysis to the data as well as removing multicollinearity in the data through analysis of the Variance Inflation Factors for the features. Furthermore, it would be nice to improve the class imbalance of the Status target by collecting more data on homeowners who paid their loan payment on time, as a 3:1 split in the data currently is a little excessive and hard improve a model on. This is especially strange considering that the bank should have more paying customers than delinquent customers, otherwise the bank would be out of business by now.

I will be using this data set for the future assignment for the Supervised Machine Learning: Classification

Course later in this certification. There I will test machine learning methods to build a predictive model for the Default status of loans. Note that the data will need to be split into testing and training prior to scaling and transforming the data in this case but the methods applied will still be the same.

Dataset Quality Assessment

As stated above the class imbalance is the most prevalent issue with this dataset, however it is not the only problem. As seen in the imputation analysis section the continuous variables had excessive amounts of missing data, above 25% of the observations were missing on important factors like the interest rate, which if the bank does not keep track of for its loan agreements, I think the IRS might need to get involved. In any case the volume of the data is still plentiful and if needed we could drop some of the rows with missing data as we would still have over 100,000 observations if we dropped 25% of the data. If record keeping improves at this bank than the data will be perfect for successful modeling of loan defaults.