CAPSTONE PROJECT 1

Mission Statement

Dream Housing is a financial institution that grants loans to people for the purpose acquiring property. There are a number of variables that are taken into account when granting a loan. Some of them are: the type of property being acquired (urban, rural, semi-urban), the applicant's income, the loan term, marital status, number of dependents, level of education, their credit history and various other variables.

I hope we can all agree that there is a certain degree of risk involved when granting loans to applicants. For example, it would definitely be a lot riskier to grant loans to people with no credit history. The company is also taking into account if the applicants are self-employed. Will they be able to continue making payments if they suffer a loss in business? Should applicants with incomes on the lower scale be granted large loan amounts? A 'risk analysis' is performed before a final decision is made.

This is clearly a complex and time consuming process. Hence, I would like to propose the design of a prediction model to automate decisions for loan approvals on behalf of Dream Housing. I have access to two data sets. One is the train file. This file consists of a history of loan approvals. Dream housing have collected information of over 600 applicants. The next dataset is test. This file consists of applicants who are still awaiting a decision. Based on the information in the train file, the model will predict if applicants (whose information is in the test file) are eligible for a loan or not.

Outline of the potential approach

The dataset consists of a total of thirteen columns. The last column in the dataset, the Loan_Status is the dependent variable. Initially, the assumption is that all the other independent variables are critical in determining the final decision. However, it is clear that the 'Loan_ID' column plays no role in determining a decision. Similarly, there could be other columns that are not 'statistically significant' enough in influencing the loan approval. We determine such columns using p-value tests and omit them from our analysis.

Once we have a dataset void of these columns, we will begin our analysis and build a model to determine the loan approval.

Cleaning Steps performed:

Before I get started, I will attach a snapshot of all the columns in the dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
Loan_ID              614 non-null object
Gender               601 non-null object
Married              611 non-null object
Dependents           599 non-null object
Education            614 non-null object
Self_Employed        582 non-null object
ApplicantIncome      614 non-null int64
CoapplicantIncome    614 non-null float64
LoanAmount           592 non-null float64
Loan_Amount_Term     600 non-null float64
Credit_History       564 non-null float64
Property_Area        614 non-null object
Loan_Status          614 non-null object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.4+ KB
```

There are 614 observation in total. Some of the columns have missing values. These values had 'NaN' assigned for missing values. I will now outline the steps I took to clean the data:

Gender – To fill the missing values of this column, I computed the mode value (most repeated value) of the Gender column. In this case, it was 'Male'

Married – Just as the gender column, missing values for the married column were filled with the mode value. It was 'married' for this column.

Dependents – Again, the mode value was used. It was '0'.

Self Employed – The mode value was computed once again.

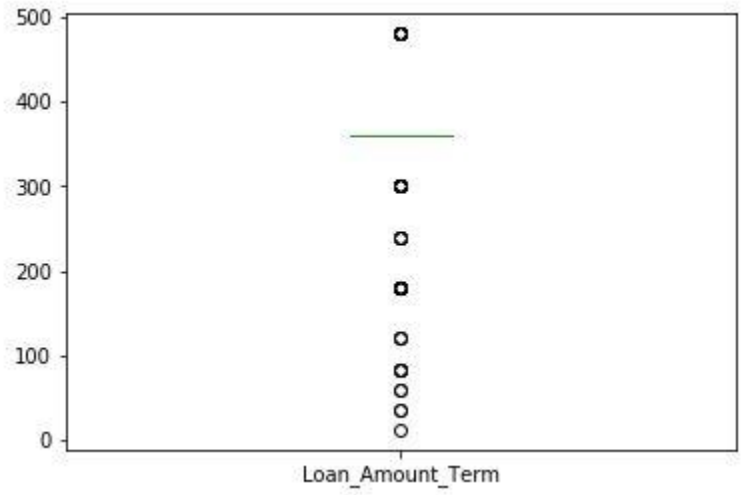
LoanAmount – The mean of Loan amounts was computed and assigned to the NaN values.

Loan_Amount_Term – The mode was computed and assigned to the missing values. It was 360.

Credit History – The mode was computed once again.

Outliers

Loan_Amount_Term – From the box plot, there is one data point > 360. On pulling values from the dataframe for Loan_Amount_term > 360, I observed that there quite a few applicants for loan_amount term = 480 and considering the relatively small size of this dataset, these data points will be taken into consideration for analysis. However, there are a few data points below 100.

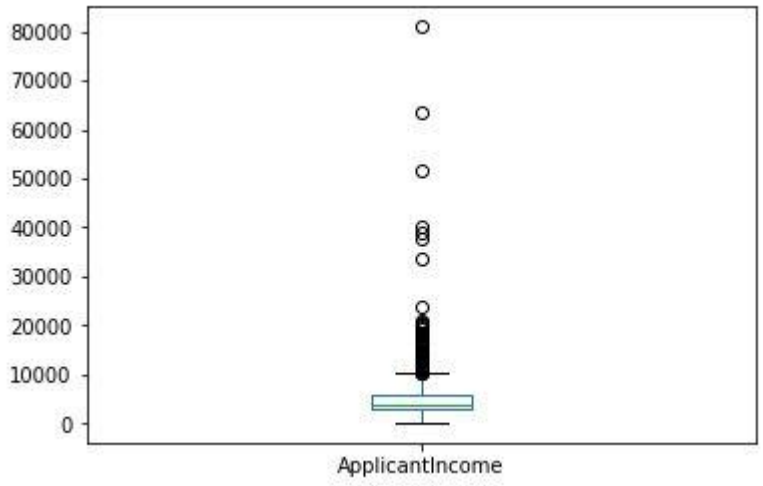


So when I take a peek at that data, here's what I get: A total of 9 data points below 100. However, we can see that there are just 2 values below 50: 12 and 24. Therefore, it seems highly unlikely that applicants would apply for a 12 or 36 term housing loan.

Proposed course of action: Omit data points below 50.

Loan_Amount_Term
60.0
60.0
36.0
84.0
84.0
12.0
36.0
84.0
84.0

ApplicantIncome - By observing the box plot for applicant incomes, we again see quite a few



outliers.

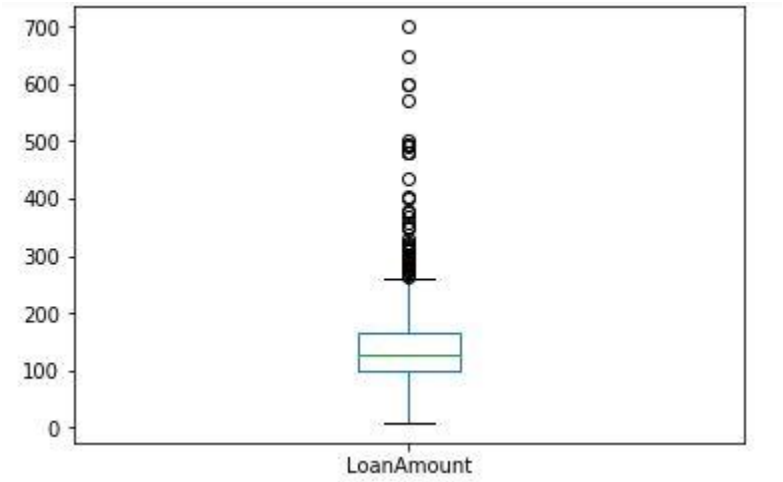
ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
51763	0.0	700.0	300.0	1.0	Urban	Y
63337	0.0	490.0	180.0	1.0	Urban	Y
81000	0.0	360.0	360.0	0.0	Rural	N

All the values in the boxplot from 50000 and above seem pretty far away and we see that there are indeed just three data points.

Proposed course of action: *Even if we do assume these rows to be legitimate, I think perhaps we could drop these rows for analysis? Simply because the % of people with an income > 50000 probably won't apply to Dream Housing*.

LoanAmount - The box plot shows quite a few outliers. We see that some outliers over 500 that are pretty isolated. We find 3 data points in the 600s and one at 700.

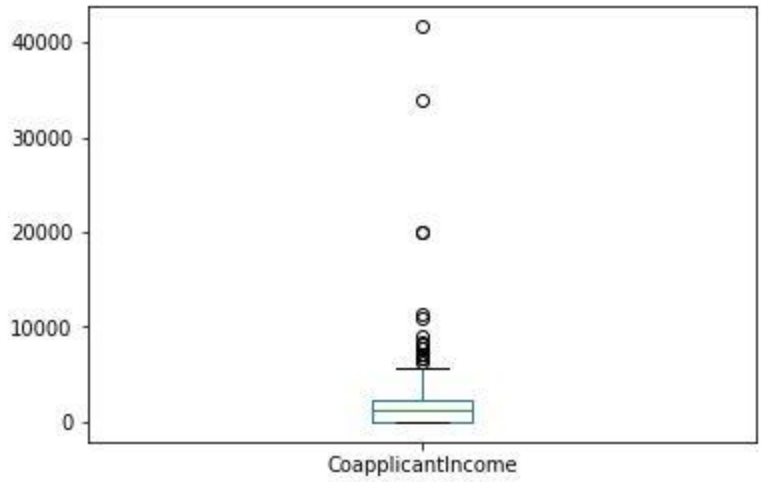
Proposed course of action: *Omit the data point at 700.*



LoanAmount
650.0
600.0
700.0
570.0
600.0

CoapplicantIncome - On observing the box plot, we again see quite a few outliers but most of them are located close to the box plot. However, there are three data points over the 20000 mark that appear to be pretty isolated.

Proposed course of action: There are only 4 data points in the data set. Unsure what to do. Need to discuss during our call.



CoapplicantIncome
20000.0
20000.0
33837.0
41667.0