INSY 5339 - Data mining

Project Proposal

Team -12

Darshanik Mekapati
Jalam Yashwanth
Sai Krishna Teja Adarsh Nadella
Bala Manikanta Ummaneni

Business Problem:

A banking Client offers loan to the eligible customers and denies the offer to customers who have not met certain criterion.

However, there are many customers who were not offered loan and they are eligible for loan approval.

Objective:

This Project is taken up by our Orion Analytics and will help in finding such type of customers through exhaustive data mining techniques.

With this the banking client can classify the customers based on data driven decision and can offer the loans more precisely.

Source: Kaggle's Loan Prediction and Approval Data sets.

Data Description:

- There are 2 datasets that were downloaded from Kaggle.
  1. Train.csv
  2. Test.csv
- Train data set has 614 rows and 13 columns. The data set has 13 Independent variables. Their names and description are given in Table1.

Variable Name	**Type**	**Description**
Gender	Independent	Gender of applicant male/female
Married	Independent	Yes/No
Dependents	Independent	0,1, 2, 3 and 3+
Education	Independent	Graduate or not graduated
Self- Employed	Independent	Is the applicant having business or job (Yes/No)
Applicant Income	Independent	A continuous variable depicting customer's income
Co Applicant Income	Independent	A continuous variable

		depicting customer's dependent income
Loan Amount	Independent	A continuous variable depicting loan amount
Loan Amount Term	Independent	Time period of loan repayment in days
Credit History	Independent	A customer having good credit history, yes/No
Property Area	Independent	Applicant's collateral property locality – urban, semiurban, or rural
Loan Status	Dependent	Sanctioned(yes) or Denied (No)
Loan ID	Independent	ID of applicant

Table 1. variables Legend

- On the other hand, Test Data set consists of different data points and does not contain Loan Status variable as we need to predict the values for it. It has same independent variables.

Data Visualization:

- Matrix graph is plotted between 4 Independent variables (Loan Amount, Applicant Income, Co-Applicant Income and Credit History) to visualize if there is any linearity among them.
- The other variables are of classification type.
- From **Fig.1**, there is visual evidence of linearity between Applicant Income and Loan Income. However, the credit history is of Binomial type and linear trend is absent. The best method to incorporate credit history is **maximum likelihood estimate (MLE)**
- To further understand the relation among the Loan Amount, Applicant Income and Co-Applicant Income variables, we use probability distribution plot of Seaborn shown in **Fig.2**.
- From **Fig.2** the Distribution is positively skewed.
- Boxplots are used to visualize outliers and the corresponding distribution for the variables. The plots are showed in **Fig.3.**
- Additionally, in Applicant Income and loan Status scatter plot (top left) in Fig.4, there is an outlier and is denoted after the vertical red line. **It says the applicant has high income (80000$) but the loan was not approved**.

Out[13]: <seaborn.axisgrid.PairGrid at 0x192e91bd4f0>

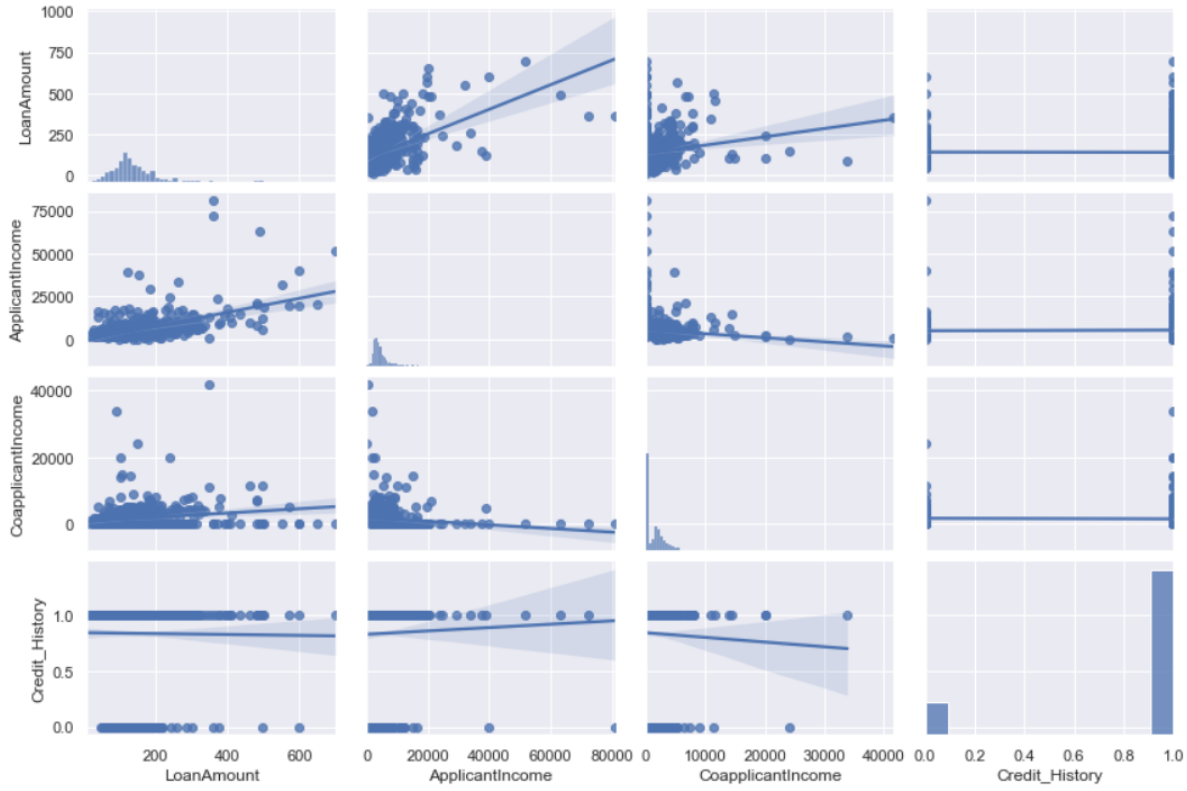

Fig.1. Matrix plot of independent variables

In [34]:
```
def dist(variable_1,variable_2,variable_3):
    sns.displot(comb_data[variable_1])
    sns.displot(comb_data[variable_2])
    sns.displot(comb_data[variable_3])
dist('LoanAmount','ApplicantIncome','CoapplicantIncome')
```

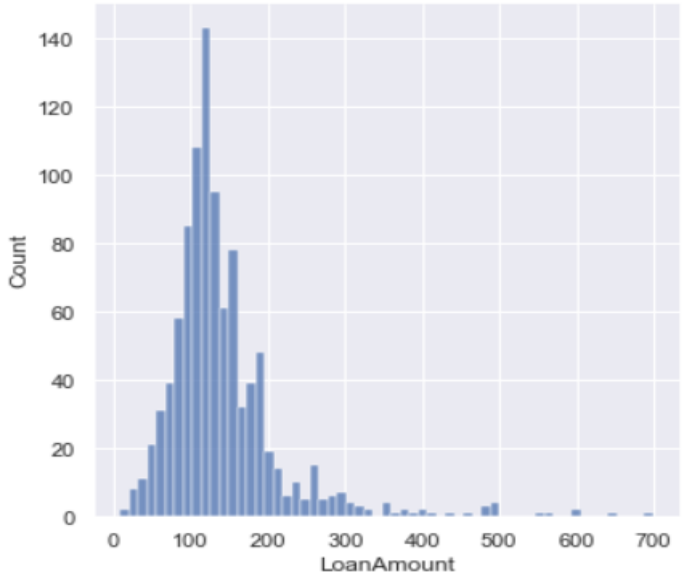

Fig.2. Probability plot of loan amount

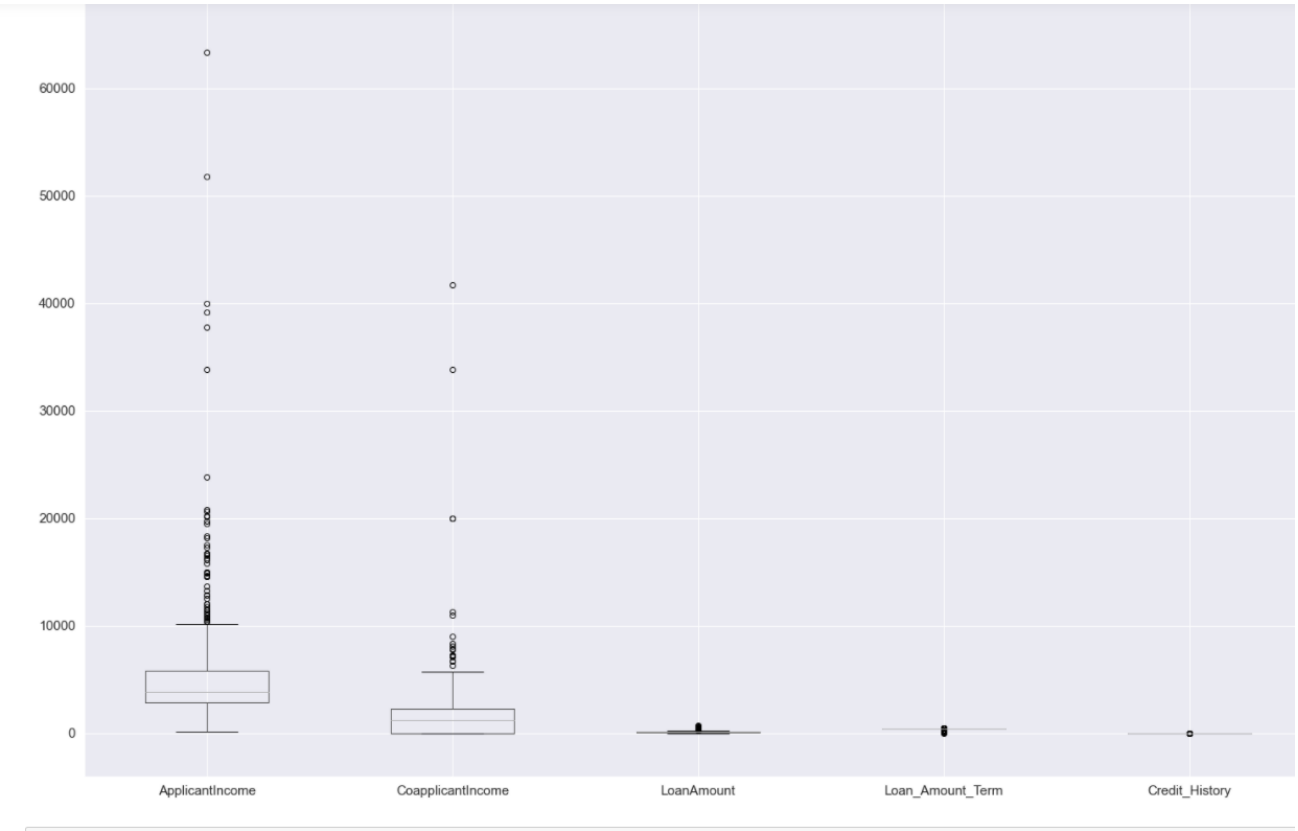

Fig.3. Box Plots for continuous independent variables

- Similarly, for Co-applicant and Loan status plot (top right) of Fig.4, there are **2 outliers where the co-applicant income is above 30000$ and the loan was not approved for their applicants**.
- These were the main scenarios that this project will address.

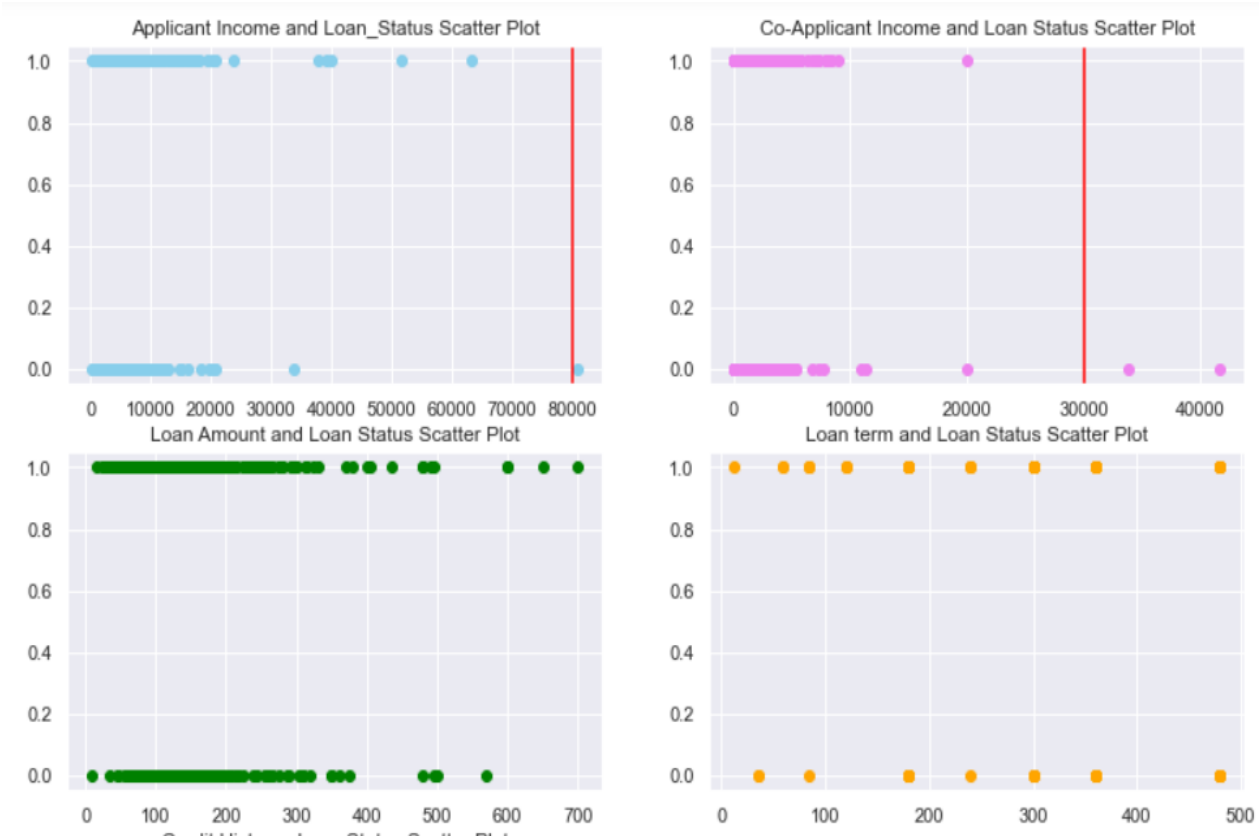

Fig.4. Potential Outliers

Data Pre-Processing:

- The train and test data sets are having missing values. They can be either dropped from the data sets or can be replaced with appropriate measure which represents the entire variable like Mean, Median or Mode (in case of Categorical data).
- Both the train and test data sets are concatenated so that it becomes easy to treat the missing values than treating them separately.
- Once the combined data set is free from missing values and potential outliers, they are sliced back into their original train and test data sets.
- Fig.5 shows the missing values when the 2 data sets are merged.
- Fig.6 shows the missing values after they are replaced with median and mode of the corresponding variable
- Dummy values are created for non-numeric and non-continuous variables. The method used to generate dummy values is One-Hot Encoding. This is the reason for increased dimension of the train and test data sets.

```
In [23]: Percent_missing = (comb_data.isnull().sum()/comb_data.isnull().count())*100
         print('\n Percent of Missing values')
         print('\n', Percent_missing)
         print('\n Missing values')
         print('\n', comb_data.isnull().sum())
```

```
 Percent of Missing values

 Gender               2.446483
Married              0.305810
Dependents           2.548420
Education            0.000000
Self_Employed        5.606524
ApplicantIncome      0.000000
CoapplicantIncome    0.000000
LoanAmount           2.752294
Loan_Amount_Term     2.038736
Credit_History       8.053007
Property_Area        0.000000
dtype: float64

 Missing values

 Gender               24
Married               3
Dependents           25
Education             0
Self_Employed        55
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount           27
Loan_Amount_Term     20
Credit_History       79
Property_Area         0
dtype: int64
```

Fig.5. Missing values Before Pre-processing

```
 train Data set missing values after pre-processing
ApplicantIncome            0
CoapplicantIncome          0
LoanAmount                 0
Loan_Amount_Term           0
Credit_History             0
Gender_Female              0
Gender_Male                0
Married_No                 0
Married_Yes                0
Dependents_0               0
Dependents_1               0
Dependents_2               0
Dependents_3+              0
Education_Graduate         0
Education_Not Graduate     0
Self_Employed_No           0
Self_Employed_Yes          0
Property_Area_Rural        0
Property_Area_Semiurban    0
Property_Area_Urban        0
Loan_Status                0
Loan_ID                    0
dtype: int64

 test Data set missing values after pre-processing
ApplicantIncome            0
CoapplicantIncome          0
LoanAmount                 0
Loan_Amount_Term           0
Credit_History             0
Gender_Female              0
Gender_Male                0
Married_No                 0
Married_Yes                0
Dependents_0               0
Dependents_1               0
Dependents_2               0
```

Fig.6. Missing values after Pre-processing

Prediction techniques and Initial results.

- Since the Dependent variable is not continuous and is binomial in nature with 2 possible outcomes (Loan approved = 1.0 or denied = 0.0), The Logit model is used for predictions and the method by which the **Logit or Logistic regression** is solved is called **Maximum likelihood estimate (MLE).** The model's performance is evaluated by **Accuracy, precision**

and Recall metrics.

- Additionally, Principal component Analysis will be used to minimize the independent variables. As mentioned in Data Pre-processing step that dummy values are created which contributed to hike in dimensions.
- Applying PCA will reduce the dimensionality by explaining as much variability as possible as it uses Eigen vectors which always points to the highest variance direction.
- Apart from Logistic regression, **Classification boosted trees** will also be used as **Random Forrest classifier** is best at solving almost all types of classification models. The **Tree's performance will be tuned or pruned with Gini Index.**
- **Fig.7** shows the initial MLE model's summary run in Python with Stats model's Logit API.

```
In [30]: results_log.summary()
```

Out[30]:

Logit Regression Results

Dep. Variable:	Loan_Status	**No. Observations:**	614
Model:	Logit	**Df Residuals:**	608
Method:	MLE	**Df Model:**	5
Date:	Sat, 23 Oct 2021	**Pseudo R-squ.:**	0.2345
Time:	22:27:16	**Log-Likelihood:**	-292.00
converged:	True	**LL-Null:**	-381.45
Covariance Type:	nonrobust	**LLR p-value:**	9.250e-37

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.9135	0.722	-2.651	0.008	-3.328	-0.499
ApplicantIncome	8.807e-06	2.29e-05	0.384	0.701	-3.61e-05	5.37e-05
CoapplicantIncome	-4.755e-05	3.24e-05	-1.468	0.142	-0.000	1.59e-05
LoanAmount	-0.0011	0.002	-0.721	0.471	-0.004	0.002
Loan_Amount_Term	-0.0011	0.002	-0.633	0.527	-0.004	0.002
Credit_History	3.8152	0.409	9.317	0.000	3.013	4.618

Fig.7 MLE result's summary

How many observations in the dataset?	**614**
How many binary/categorical variables?	**9**
How many continuous variables?	**3**
What is the outcome / target variable?	**Loan Status**
If binary or categorical: What percentage of the variables belong to each class.	**Loan Approved (1) = 422** **Loan Approval rate = 69%** **Loan Denied (0) = 192** **Loan Denial rate = 31%** **male population = 80%** **Female population = 18%** **Other population = 2%** **Graduates = 78%** **Non-graduates = 22%** **Self-Employed = 13%** **Non-Self-Employed = 81%** **Others = 6%** **Property Area Urban = 33%** **Property Area semi-urban = 38%** **Property Area Rural = 290%** **“0” Dependents = 56%** **“1” Dependent = 17%** **“2” Dependents = 16%** **“3+” Dependents = 8%**

If continuous: What is the mean value of the target variable?	**Target variable is not continuous**
Before doing any further processing, what would your prediction of the target variable be?	**Loan Approvals (1) for Denied customers**

Table 2. Summary of Data set and Report