TEMPORARY SLIDES

FINAL PROJECT

P R E S E N T A T I O N

CONTENT

PROBLEM OVERVIEW

ANALYSIS DATA

DATA MODELING

CONCLUSION & IDEAL

Hypothetical problem

LOAN PREDICTION

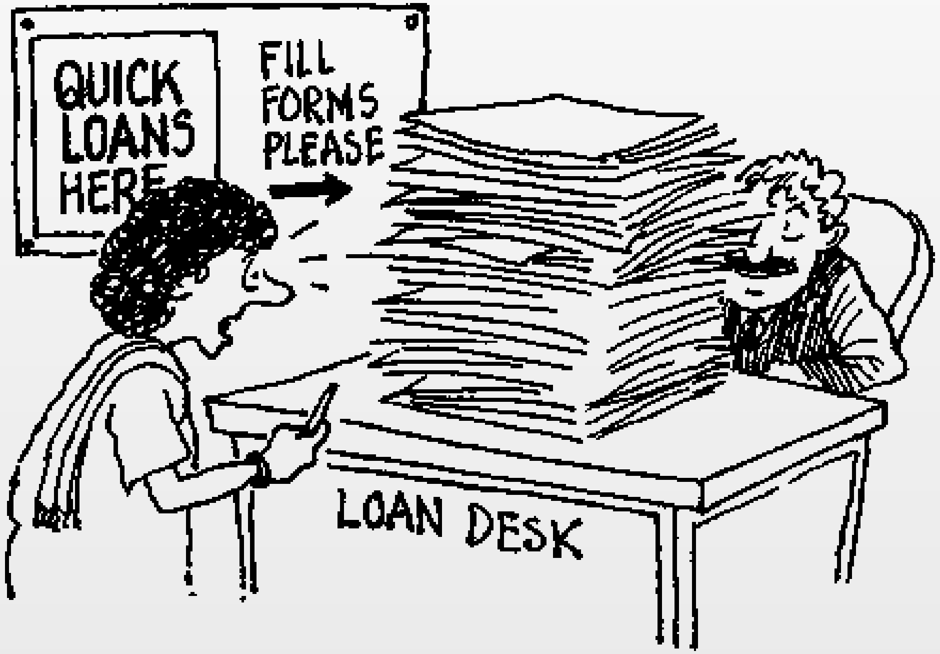
Bank A offers a home loan credit package

Customers in need will apply for a home loan

Bank validates the customer eligibility for a loan

Loan prediction practice problem

Understanding the problem



Banks need to speed up the working process

Bank wants to automate the loan eligibility process based on customer detail provided

---

To automate this process, they have given a problem to identify the customer's segments, those are eligible for loan amount so that they can specifically target these customers.

Hypothesis generation

Financial ability: higher income makes it easier to repay the bank debt

---

Loan history: an applicant who has paid their debts before are likely to gain the trust of the bank

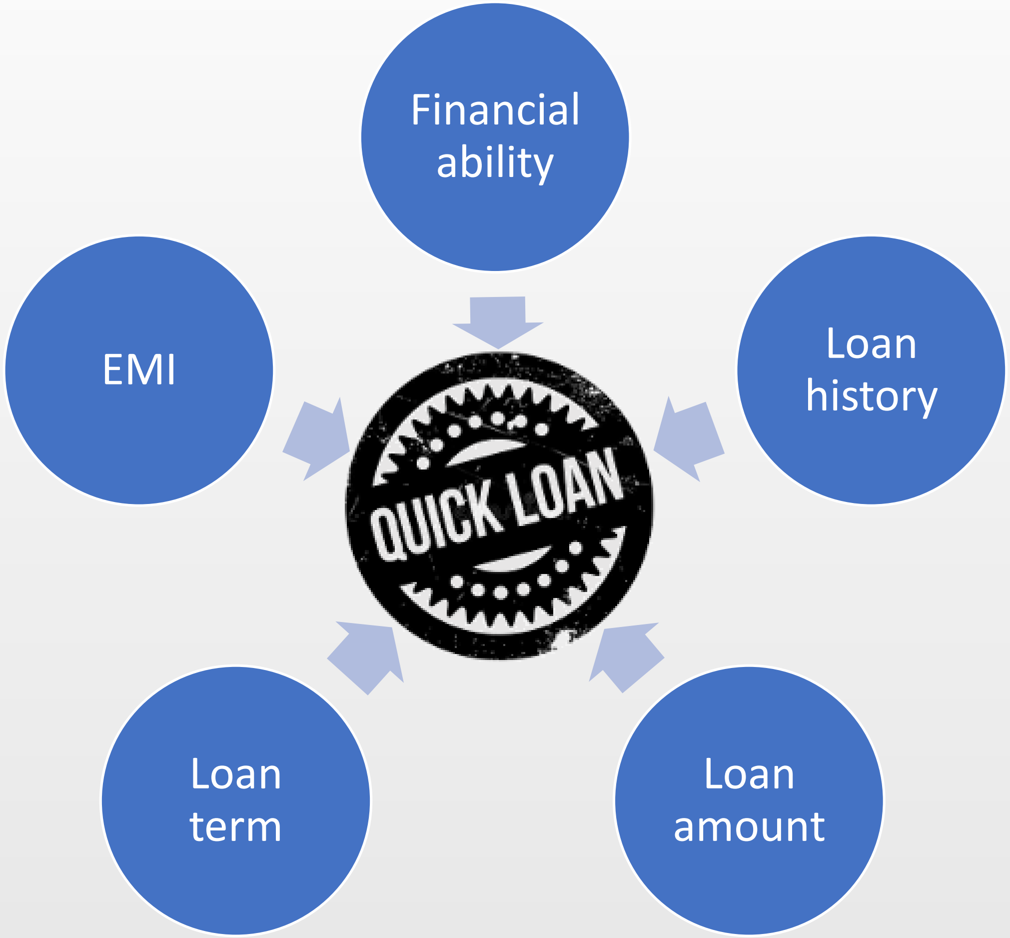
---

Loan amount: smaller loans make it easier to get approved

Loan term: A loan for less period and less amount have higher chances of approval

---

EMI: Lesser the amount to be paid monthly to repay the loan, the higher the chances of loan approval



Introduction to dataset

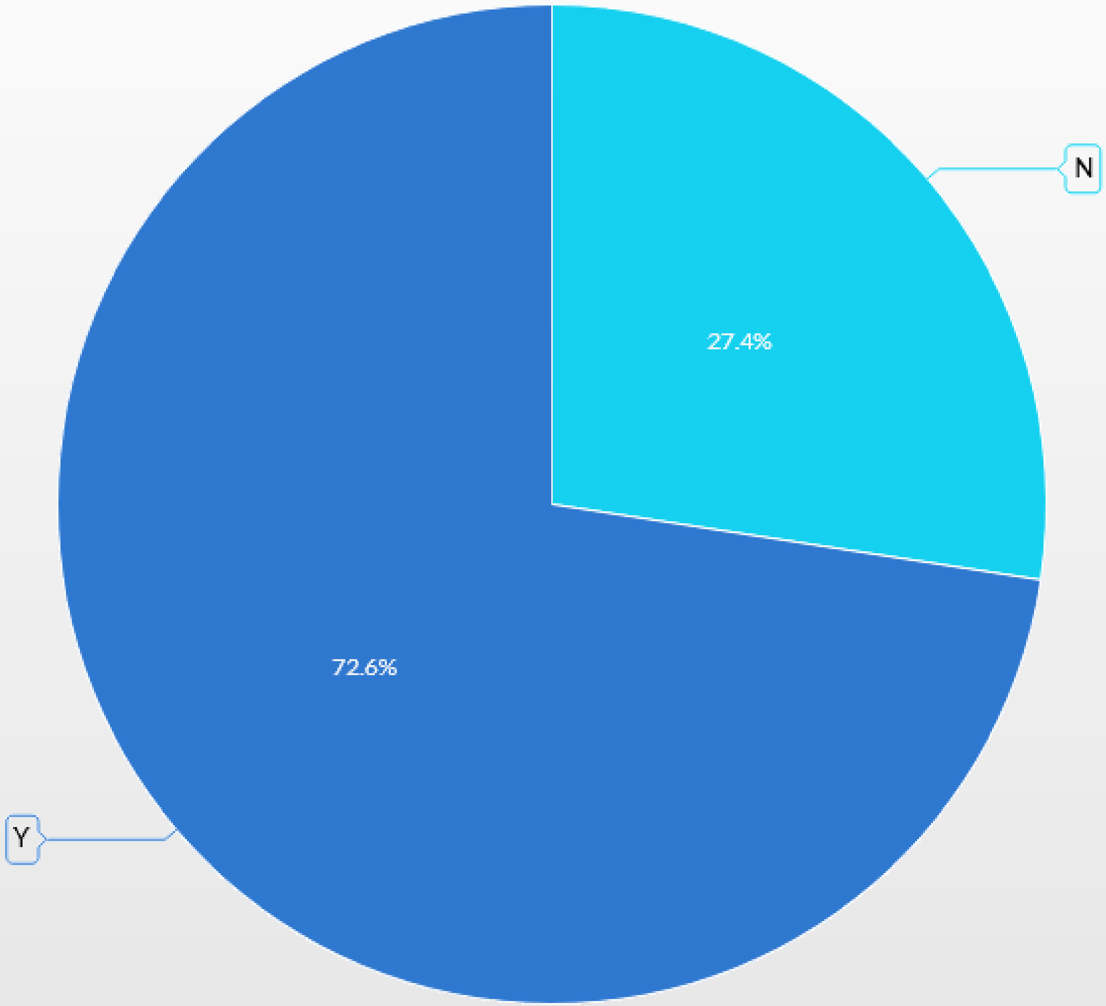
Loan Prediction Dataset

Link download

What data do we have?

Variable	Description
Loan_ID	Unique Loan ID
Gender	Male/ Female
Married	Applicant married (Yes/No)
Dependents	Number of dependents(1,2,3+)
Education	Applicant Education (Graduate/ Under Graduate)
Self_Employed	Self employed (Yes/No)
ApplicantIncome	Applicant income
CoapplicantIncome	Coapplicant income
LoanAmount	Loan amount in thousands
Loan_Amount_Term	Term of loan in months
Credit_History	credit history meets guidelines
Property_Area	Urban/ Semi Urban/ Rural
Loan_Status	Loan approved (Y/N)

ANALYSIS DATA

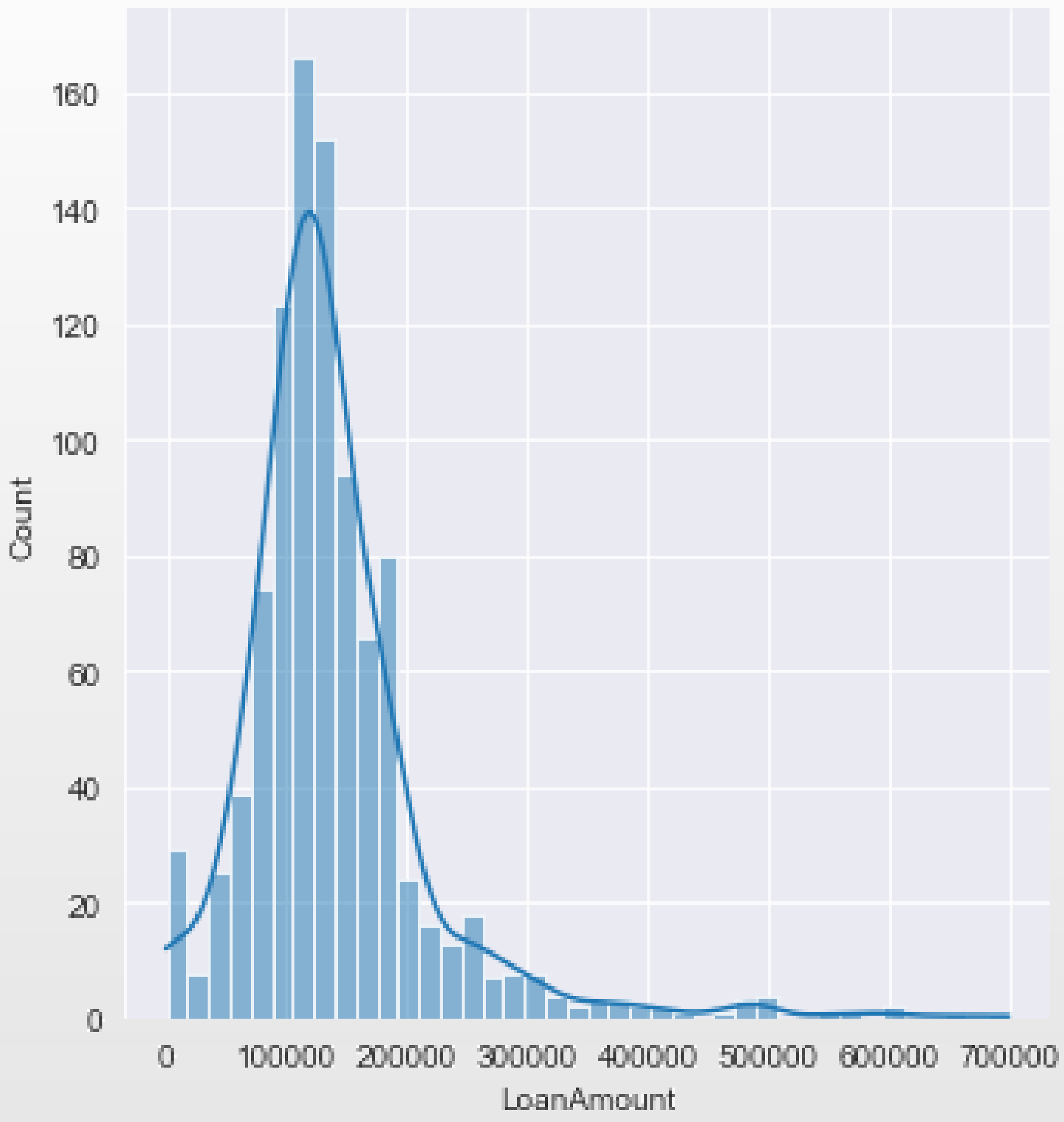


Target variable

Loan Status

Loan Approval Status

Over $\frac{2}{3}$ of applicants have been granted a loan



Histogram

Distribution of Loan Amount

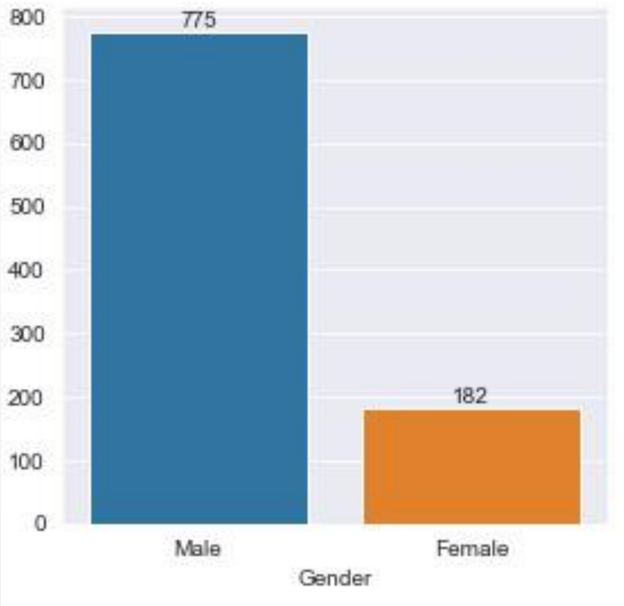
ANALYSIS DATA

Relationship between Loan Amount and Applicant Income

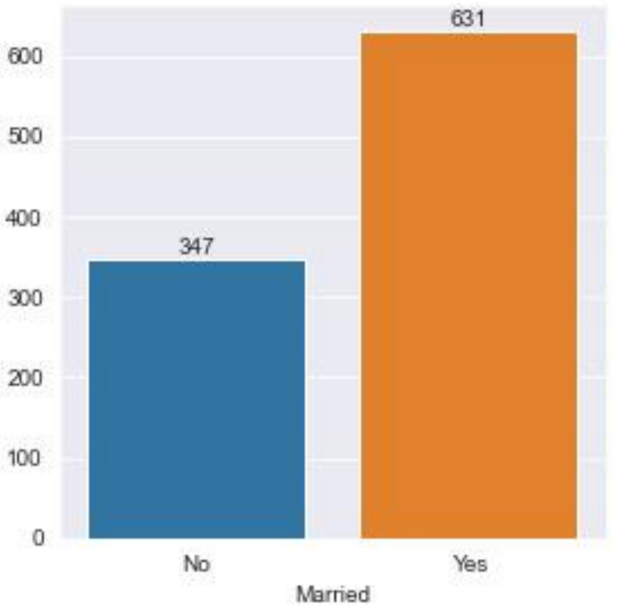
ANALYSIS DATA

BLANK

ANALYSIS DATA

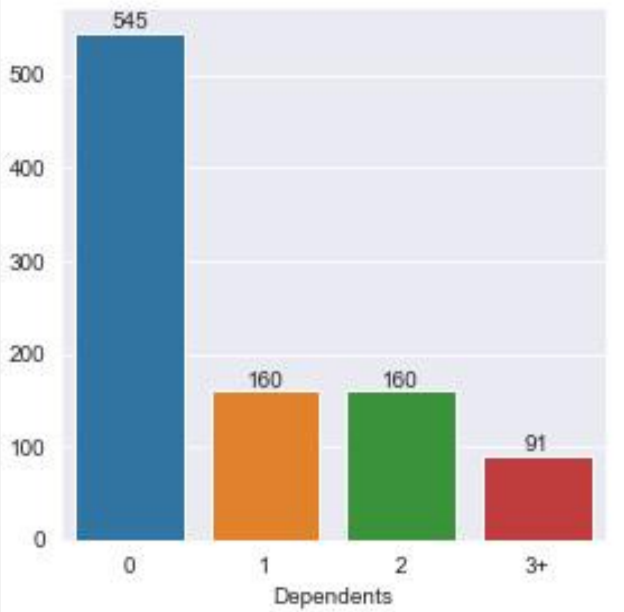


SEX

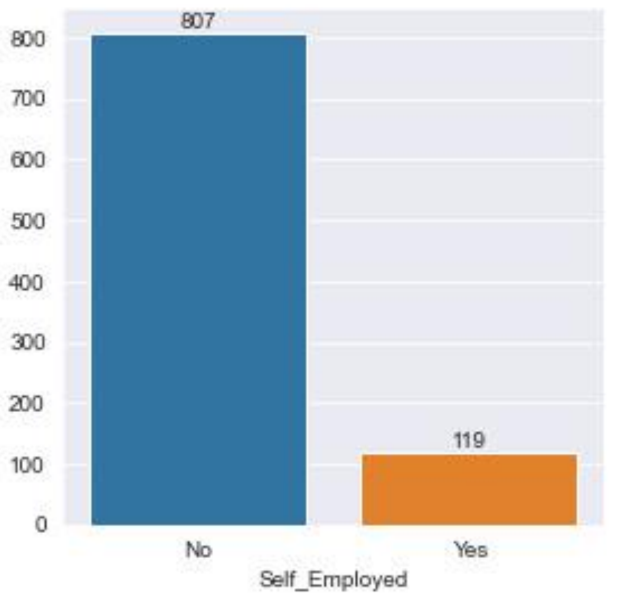


MARRIAGE

NUMBER OF DEPENDENTS

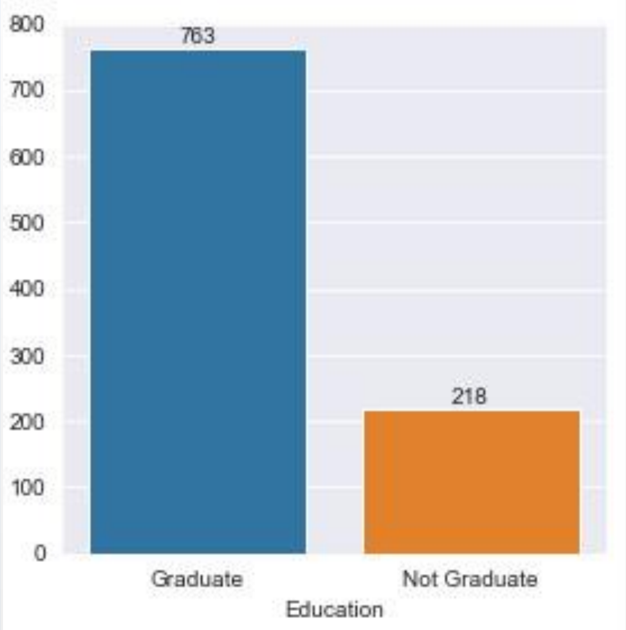


EMPLOYMENT STATUS

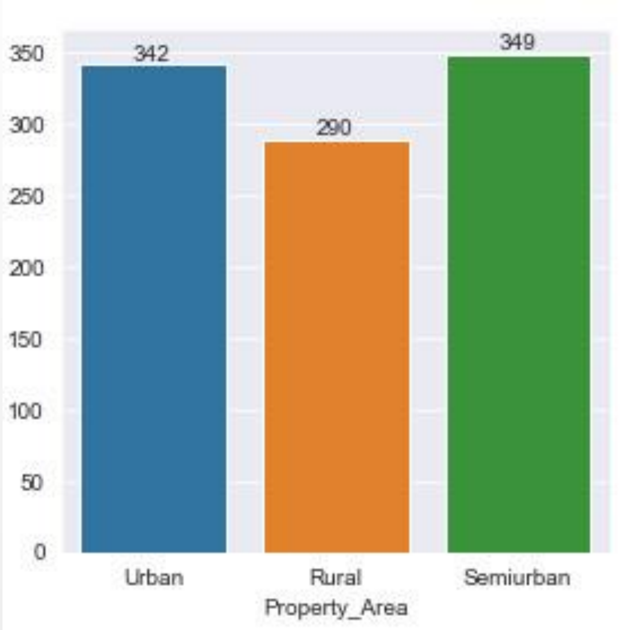


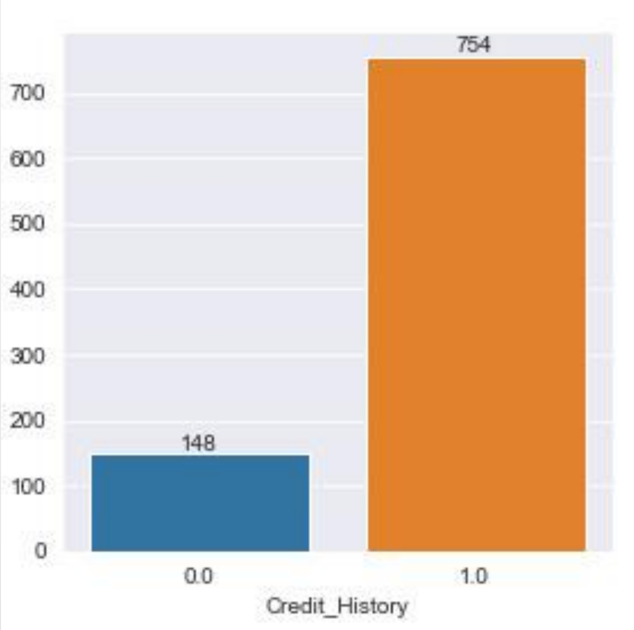
ANALYSIS DATA

EDUCATION

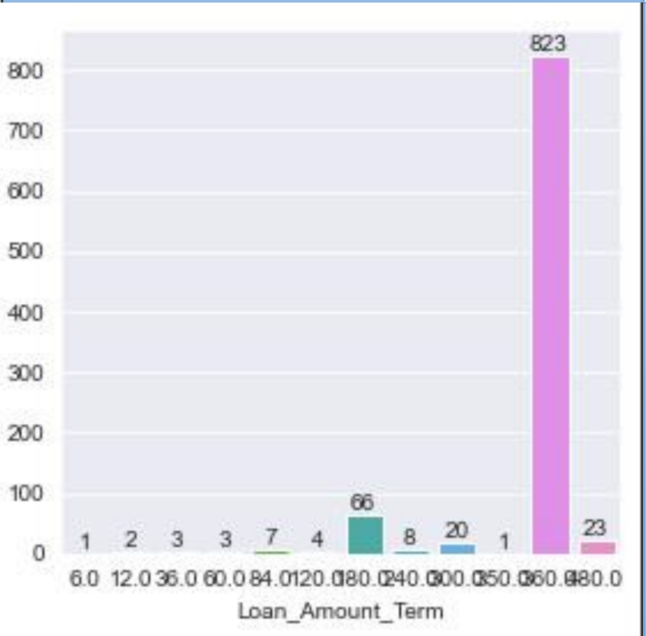


AREA





CREDIT HISTORY



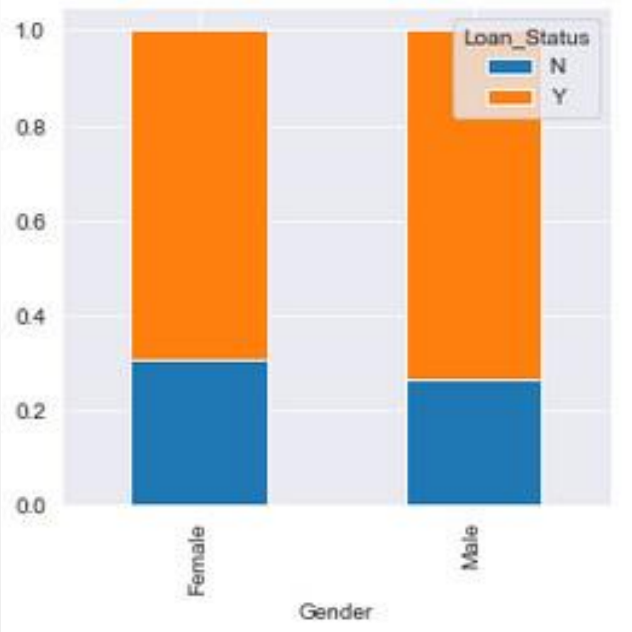
LOAN TERM

ANALYSIS DATA

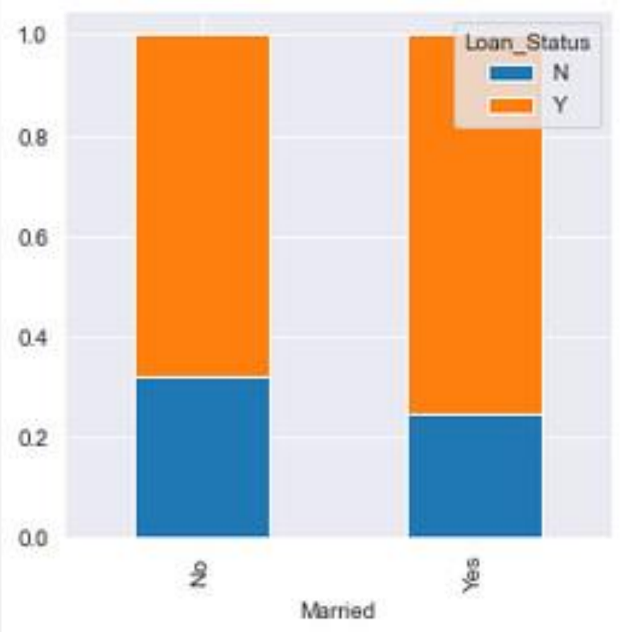
CUSTOMER DESCRIPTION?

1 …………
2 …………
3 …………
4 …………
…

ANALYSIS DATA

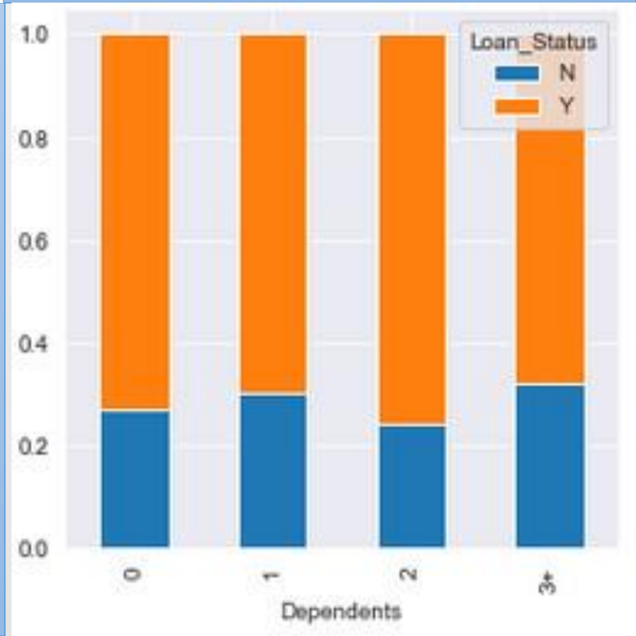


SEX

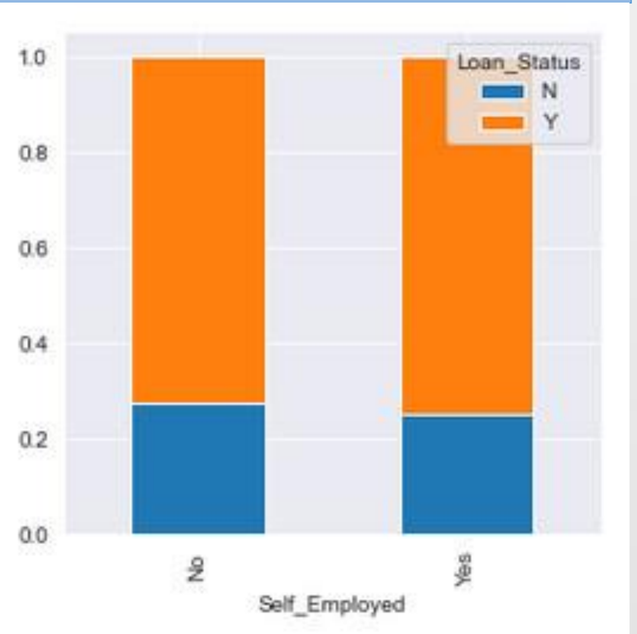


MARRIAGE

NUMBER OF DEPENDENTS

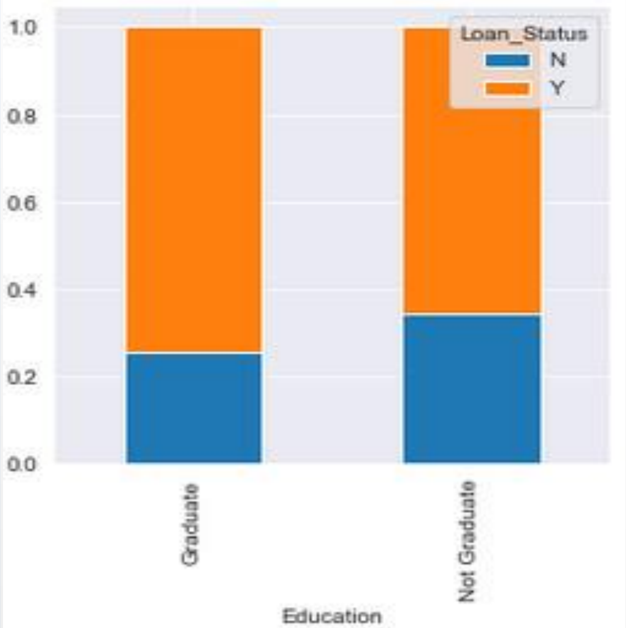


EMPLOYMENT STATUS

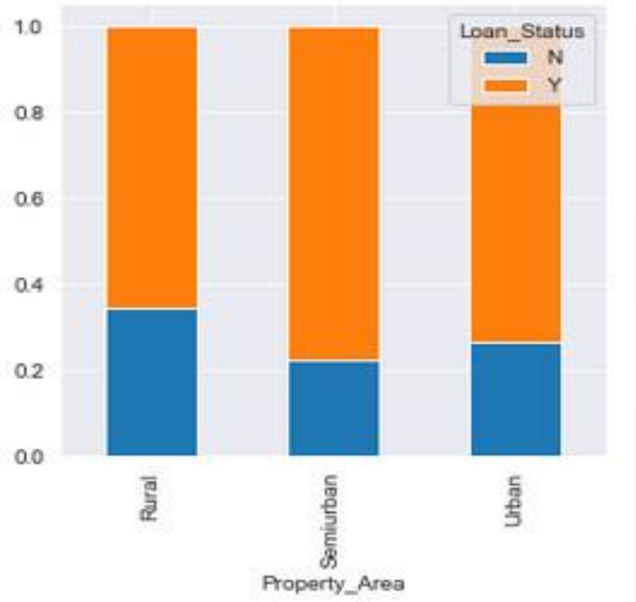


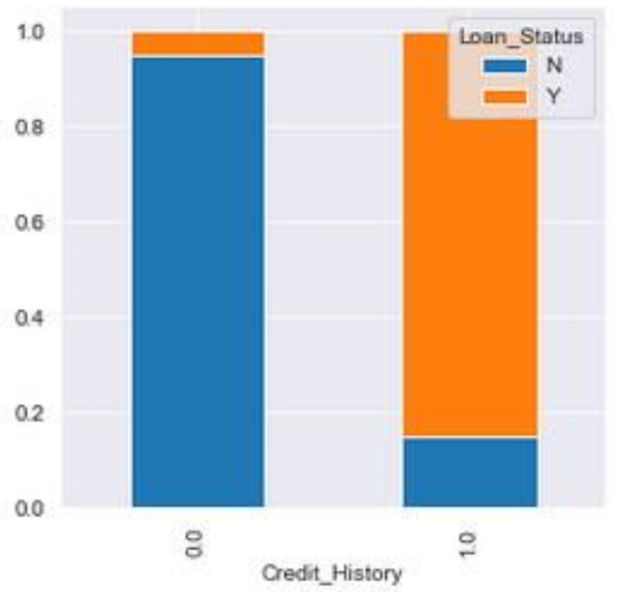
ANALYSIS DATA

EDUCATION



AREA





CREDIT HISTORY

LOAN TERM

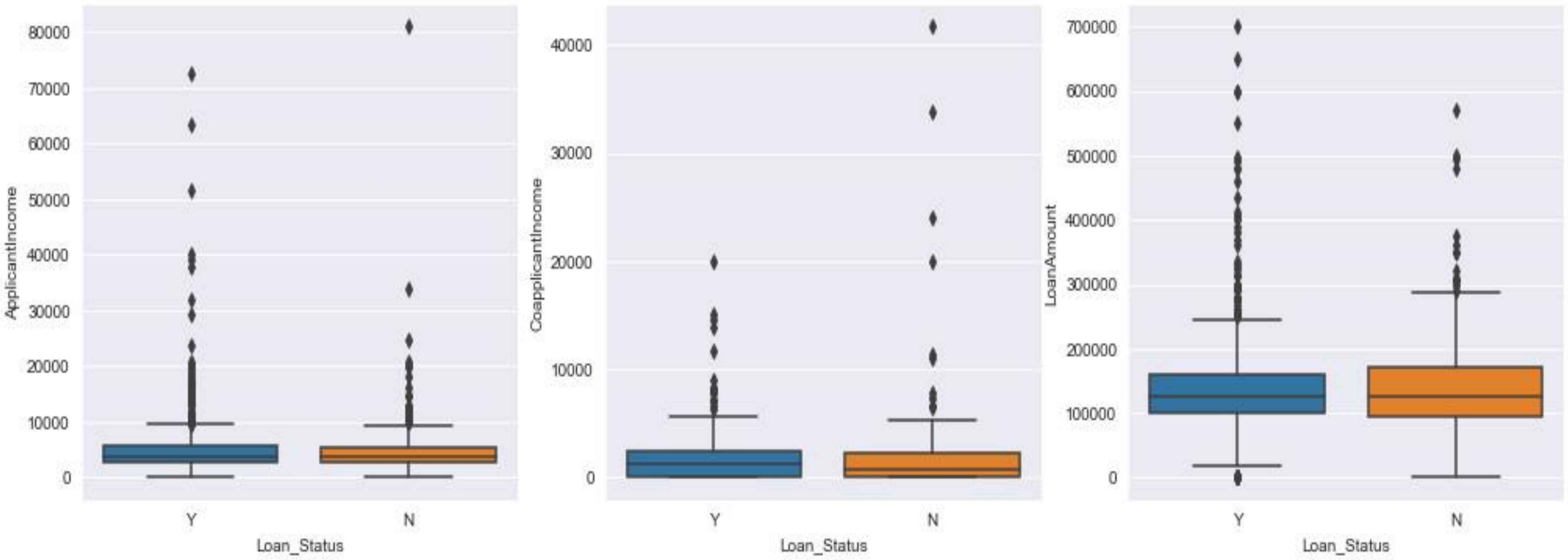
ANALYSIS DATA

BLANK

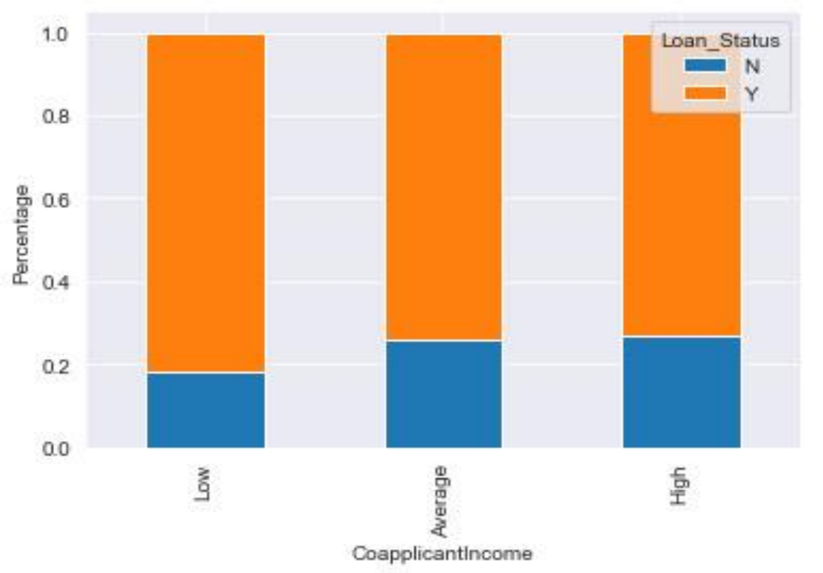
FINANCIAL ABILITY ANALYSIS

	ApplicantIncome	CoapplicantIncome	LoanAmount
count	981.000000	981.000000	981.000000
mean	5179.795107	1601.916330	138589.194699
std	5695.104533	2718.772806	79831.886151
min	0.000000	0.000000	0.000000
25%	2875.000000	0.000000	99000.000000
50%	3800.000000	1110.000000	125000.000000
75%	5516.000000	2365.000000	160000.000000
max	81000.000000	41667.000000	700000.000000

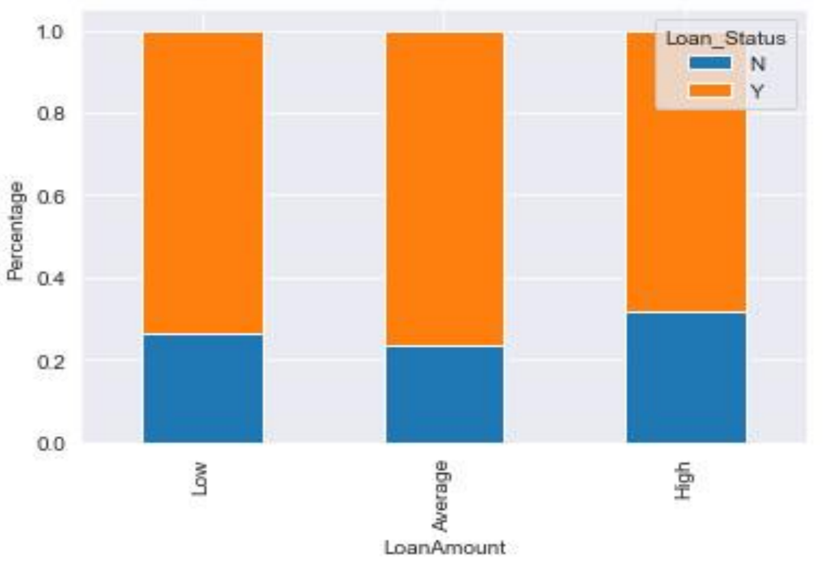
ANALYSIS DATA



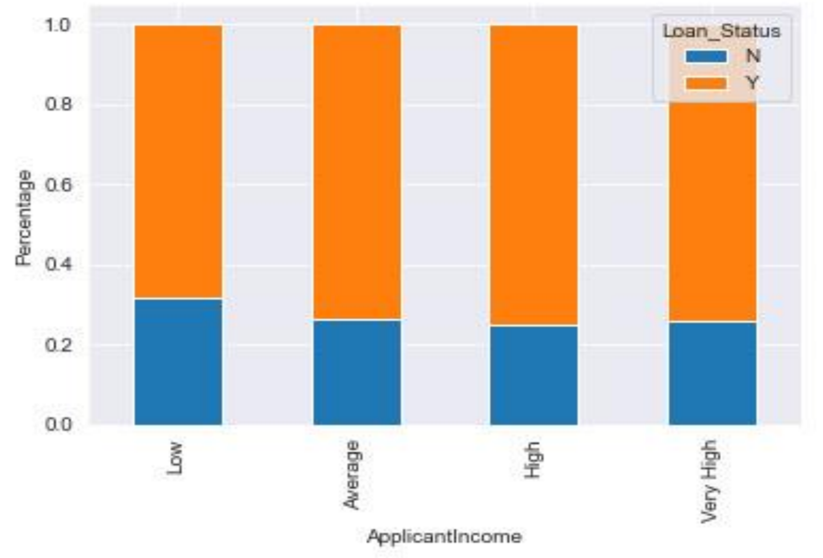
ANALYSIS DATA



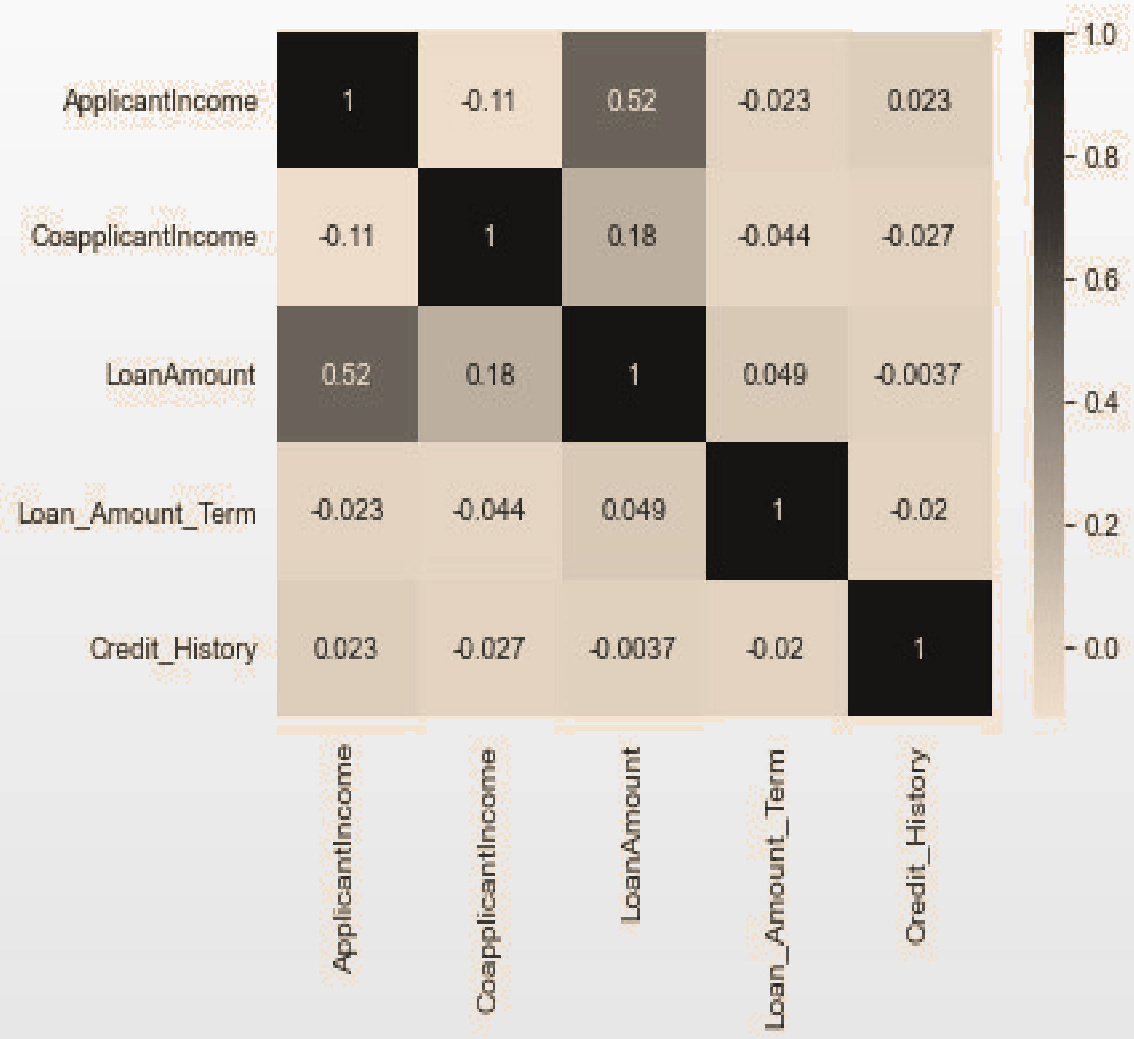
BLANK



BLANK



BLANK



CORRELATION

DATA PROCESSING

Handling missing data

Code

```
# Handle missing values
# Categorical data, ordinal data, fill with mode
data.Gender.fillna(data.Gender.mode()[0], inplace= True)
data.Married.fillna(data.Married.mode()[0], inplace= True)
data.Dependents.fillna(data.Dependents.mode()[0], inplace= True)
data.Self_Employed.fillna(data.Self_Employed.mode()[0], inplace= True)
data.Loan_Amount_Term.fillna(data.Loan_Amount_Term.mode()[0], inplace= True)
data.Credit_History.fillna(data.Credit_History.mode()[0], inplace= True)
# Numerical data, continuous data, fill with median
data.LoanAmount.fillna(data.LoanAmount.median(), inplace= True)
# Median is better than mean in this data as the data is skewed towards one side that was visualized earlier
```

DATA PROCESSING

Feature engineering

Code

```
# Create a new feature
# Total Income: combine the applicant income and co-applicant income
data['TotalIncome']=data['ApplicantIncome']+data['CoapplicantIncome’]
# EMI: Equated monthly installments to be paid back
data['EMI']=data['LoanAmount']/data['Loan_Amount_Term’]
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
data_scaled = sc.fit_transform(data[['TotalIncome', 'EMI']])
```

BUILDING LOGISTIC MODEL

Code

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
logreg = LogisticRegression()
X = data.drop(['Loan_Status'],axis=1)
y = data['Loan_Status']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=7, stratify=y)
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
```

DATA MODELING

Classification report

	precision	recall	f1-score	support
N	0.91	0.54	0.67	54
Y	0.85	0.98	0.91	143
accuracy			0.86	197
macro avg	0.88	0.76	0.79	197
weighted avg	0.86	0.86	0.84	197

DATA MODELING

Weighted of features

	feature_names	weight	abs_weight
14	Credit_History_0.0	-2.321865	2.321865
15	Credit_History_1.0	2.180041	2.180041
16	Property_Area_Rural	-0.573206	0.573206
17	Property_Area_Semiurban	0.467221	0.467221
7	Dependents_1	-0.320746	0.320746
4	Married_No	-0.307240	0.307240
11	Education_Not Graduate	-0.260216	0.260216
12	Self_Employed_No	-0.195911	0.195911
5	Married_Yes	0.165415	0.165415
10	Education_Graduate	0.118391	0.118391

CONCLUSION

CONCLUSION & IDEAL

IDEAL


THANK
YOU