RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054
(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

Internship Report

on

LOAN ELIGIBILITY PREDICTION

INT411: Intra Institutional Internship

TEAM MEMBERS

Name	USN
ABHAY KARTHIK D	1MS21AI002
CHAITANNYA NAIDU	1MS21AI013
AMULYA A R	1MS21AI007

Ramaiah Institute of Technology

(Autonomous Institute, Affiliated to VTU)

MSR Nagar, MSRIT Post, Bangalore-560054

September – October, 2022

RAMAIAH
Institute of Technology

INTERNSHIP ASSESSMENT

---------- Internship Title ----------

INT411: Intra Institutional Internship

SL No.	Component	Maximum Marks	Marks Obtained
1	Continuous Evaluation	50	
2	Presentation	20	
3	Report	30	
Total Marks		100	

Signature of Student **Signature of Faculty Coordinator**

RAMAIAH INSTITUTE OF TECHNOLOGY, BANGALORE – 560054
(Autonomous Institute, Affiliated to VTU)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that Mr./Ms. _____, a student of Bachelor of Engineering, bearing USN: _____, has successfully completed, 30 Hours: from 29.09.2022 to 15.10.2022 Intra Institutional Internship in ----------, from the Department of ----------,M S Ramaiah Institute of Technology, Bangalore.

TABLE OF CONTENTS

ABSTRACT

When any financial institution lends the money to the person, it is always been a high risk. Today data is increasing with the rapid pace in the banks, therefore the bankers need to evaluate the person's data before giving the loan. It can be a big headache to evaluate the data. This problem is solved by analyzing and training the data by using one of the Machine Learning algorithms. For this, we have generated a model for the prediction that the person will get the loan or not. The primary objective of this paper is to check whether the person can get the loan or not by evaluating the data with the help of decision tree classifiers which can gives the accurate result for the prediction.

Keywords—Loan, Machine Learning, Data training.

INTRODUCTION

1.1 GENERAL

LOAN

A loan is money, property, or other material goods given to another party in exchange for future repayment of the loan value amount with interest.

TYPES OF LOAN

1. Home loans
2. Gold loans
3. Personal loans
4. Education loans
5. Vehicle loans

Other forms of secured loans include loans against securities – such as shares, mutual funds, bonds, etc. This particular instrument issues customers a line of credit based on the quality of the securities pledged. Gold loans are issued to customers after evaluating the quantity and quality of gold in the items pledged. Corporate entities can also take out secured lending by pledging the company's assets, including the company itself

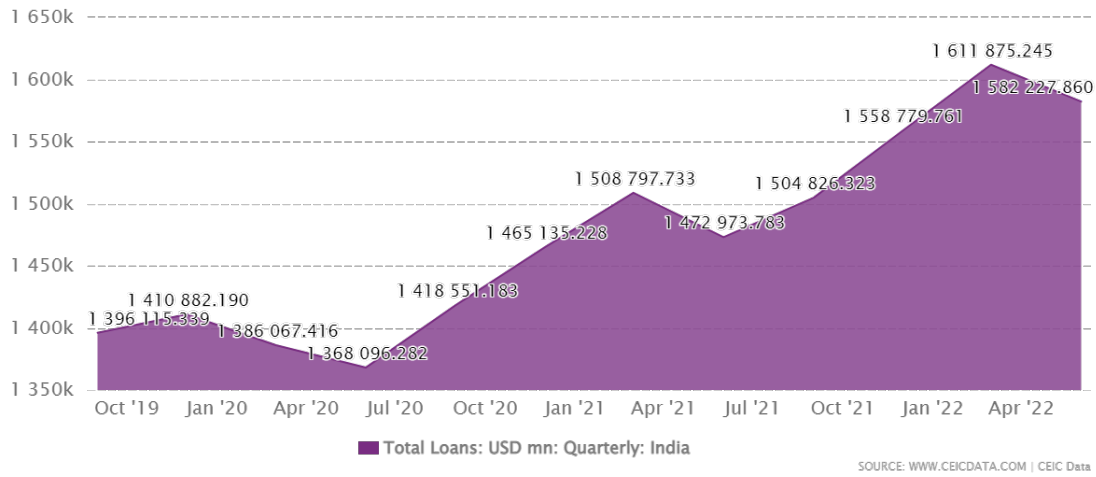
India Total Loans

. India Total Loans was reported at 1,582.228 USD bn in Jun 2022

. This records a decrease from the previous number of 1,611.875 USD bn for Mar 2022

. India Total Loans data is updated quarterly, averaging 771.308 USD bn from Dec 1998 to Jun 2022.

. The data reached an all-time high of 1,611.875 USD bn in Mar 2022 and a record low of 82.701 USD bn in Dec 1998



1.2 PROBLEM STATEMENT

. As of March 2022, this sector accounted for 460 million active loans. Active personal loans also experienced a growth of 46 per cent between March 2021 and March 2022. The personal loan portfolio outstanding grew by 23 per cent in the last one year.

. This means that banks have millions of customers applying for loans on a daily basis, and this results in a large dataset that has to be stored.

.For such large data it is difficult to check every single entry to sanction the loan.

.Sanctioning a loan is a tough and tedious job as it has many parameters to be checked before giving out the loan. Such as age, income ,credit history , gender , martial status etc.

. this stored data has to be sorted and our machine learning model will predict if the customer is eligible for a loan or not.

.

1.3 OBJECTIVES OF THE PROJECTS

. The main objective of the project is to create a system which can predict whether the customer is eligible for the loan or not.

. We created a module which makes the process easier and very efficient.

.It predicts the result for a large amount of data in a short span of time.

MACHINE LEARNING

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

Terminologies of Machine Learning

Model

A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called hypothesis.

- **Feature**

A feature is an individual measurable property of our data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a loan, there may be features like age, income, credit, etc. **Note:** Choosing informative, discriminating and independent features is a crucial step for effective algorithms. We generally employ a feature extractor to extract the relevant features from the raw data.

- **Target (Label)**

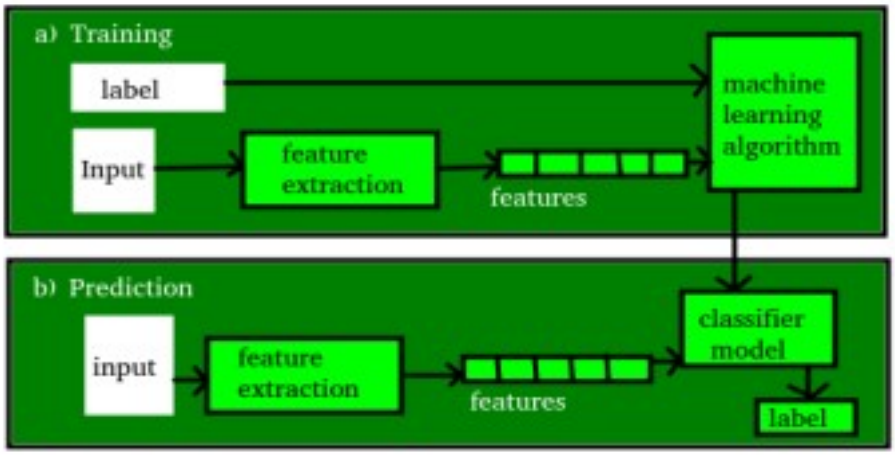
A target variable or label is the value to be predicted by our model. For the loan example discussed in the features section, the label with each set of input would be the unique id of the customer.

- **Training**

The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.

- **Prediction**

  Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).



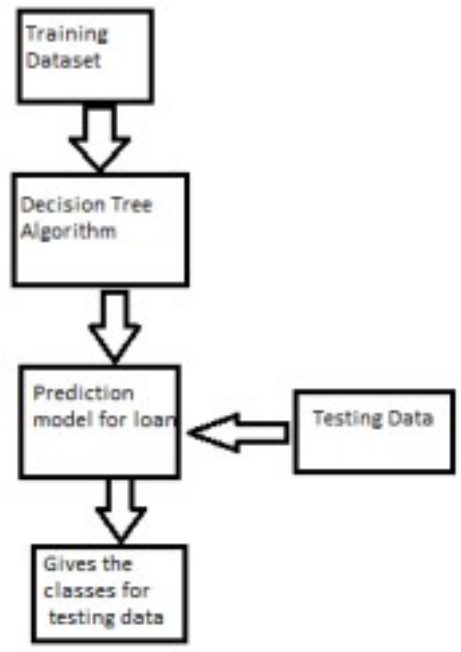
IMPLEMENTATION OF MODEL

EXISTING SYSTEM

Banks need to analyze for the person who applies for the loan will repay the loan or not. Sometime it happens that customer has provided partial data to the bank, in this case person may get the loan without proper verification and bank may end up with loss. Bankers cannot analyze the huge amounts of data manually, it may become a big headache to check whether a person will repay its loan or not. It is very much necessary to know the person getting loan is going in safe hand or not. So, it is pretty much important to have a automated model which should predict the customer getting the loan will repay the loan or not.

PROPOSED SYSTEM

we have developed a prediction model for Loan sanctioning which will predict whether the person applying for loan will get loan or not. The major objective of this project is to derive patterns from the datasets which are used for the loan sanctioning process and create a model based on the patterns derived in the previous step. This model is developed by using the one of the machine learning algorithms.

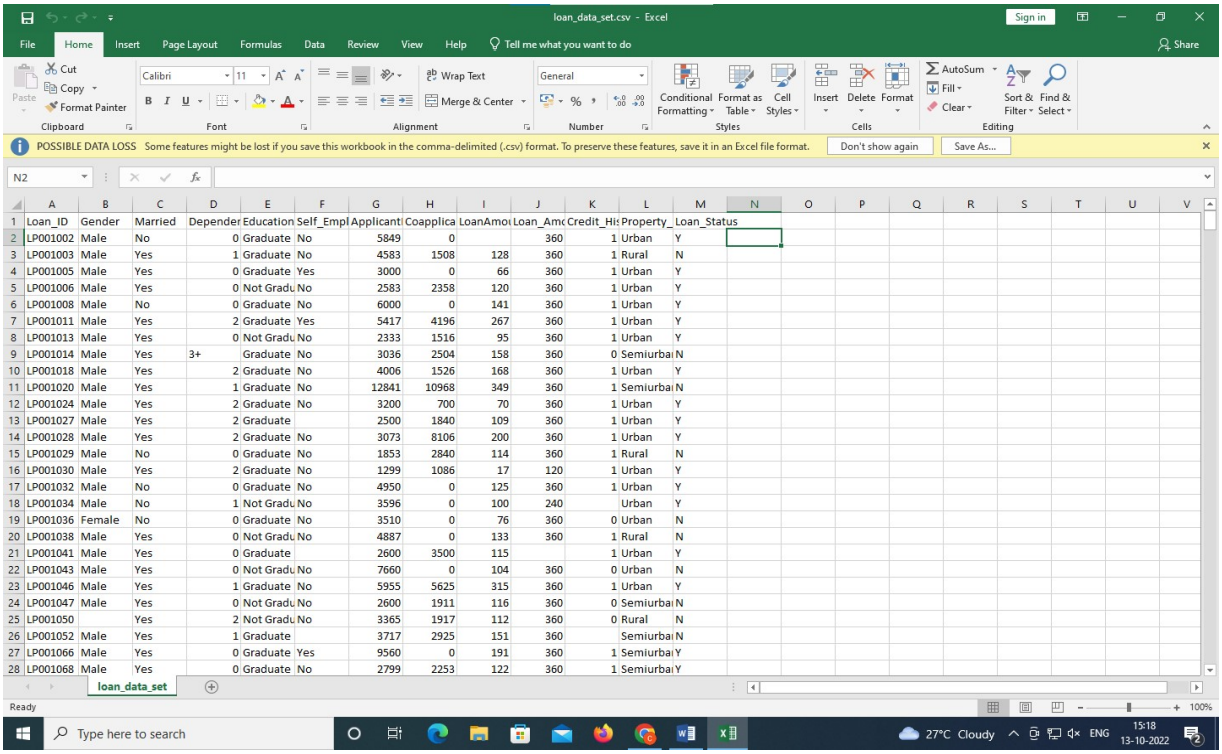


In the proposed model for loan prediction, Dataset is split into training and testing data. After then training datasets are trained using the decision tree algorithm and a prediction model is developed using the algorithm. Testing datasets are then given to model for the prediction of loan. The motive of this paper is to predict the defaults who will repay the loan or not. Various libraries like pandas, numpy have been used. After the loading of datasets, Data Preprocessing like missing value treatment of numerical and categorical is done by checking the values. Numerical and categorical values are segregated. Outliers and frequency analysis are done, outliers are checked by getting the boxplot diagram of attributes.

:

DATA SET

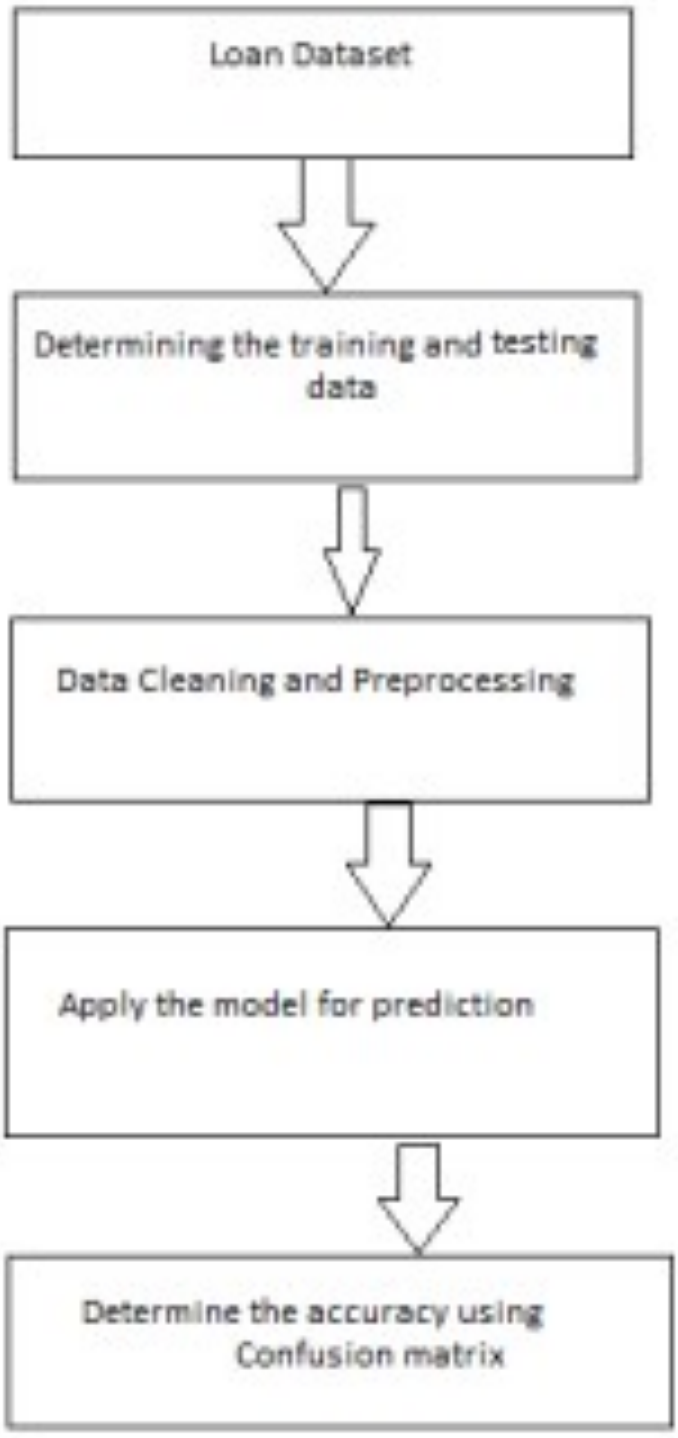
Datasets are gathered from Kaggle. Data set is now provided to Machine learning models on the basis of this facts this version is trained. Data sets are divided into Existing and New Customers. Every new applicant info act as a fact test set. After the operation of testing, model expect whether the brandnew applicant is in case for approval of the loan or now not primarily based upon the inference it concludes on the idea of training information sets.



loan_data_set.csv - Excel

Loan_ID	Gender	Married	Dependents	Education	Self_Empl	Applicant	Coapplica	LoanAmo	Loan_Amo	Credit_His	Property_	Loan_Status
LP001002	Male	No	0	Graduate	No	5849	0		360	1	Urban	Y
LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1	Rural	N
LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1	Urban	Y
LP001006	Male	Yes	0	Not Gradu	No	2583	2358	120	360	1	Urban	Y
LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Urban	Y
LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1	Urban	Y
LP001013	Male	Yes	0	Not Gradu	No	2333	1516	95	360	1	Urban	Y
LP001014	Male	Yes	3+	Graduate	No	3036	2504	158	360	0	Semiurbai	N
LP001018	Male	Yes	2	Graduate	No	4006	1526	168	360	1	Urban	Y
LP001020	Male	Yes	1	Graduate	No	12841	10968	349	360	1	Semiurbai	N
LP001024	Male	Yes	2	Graduate	No	3200	700	70	360	1	Urban	Y
LP001027	Male	Yes	2	Graduate		2500	1840	109	360	1	Urban	Y
LP001028	Male	Yes	2	Graduate	No	3073	8106	200	360	1	Urban	Y
LP001029	Male	No	0	Graduate	No	1853	2840	114	360	1	Rural	N
LP001030	Male	Yes	2	Graduate	No	1299	1086	17	120	1	Urban	Y
LP001032	Male	No	0	Graduate	No	4950	0	125	360	1	Urban	Y
LP001034	Male	No	1	Not Gradu	No	3596	0	100	240		Urban	Y
LP001036	Female	No	0	Graduate	No	3510	0	76	360	0	Urban	N
LP001038	Male	Yes	0	Not Gradu	No	4887	0	133	360	1	Rural	N
LP001041	Male	Yes	0	Graduate		2600	3500	115		1	Urban	Y
LP001043	Male	Yes	0	Not Gradu	No	7660	0	104	360	0	Urban	N
LP001046	Male	Yes	1	Graduate	No	5955	5625	315	360	1	Urban	Y
LP001047	Male	Yes	0	Not Gradu	No	2600	1911	116	360	0	Semiurbai	N
LP001050		Yes	2	Not Gradu	No	3365	1917	112	360	0	Rural	N
LP001052	Male	Yes	1	Graduate		3717	2925	151	360		Semiurbai	N
LP001066	Male	Yes	0	Graduate	Yes	9560	0	191	360	1	Semiurbai	Y
LP001068	Male	Yes	0	Graduate	No	2799	2253	122	360	1	Semiurbai	Y

IMPLEMENTATION OF THE MODEL



CODE OF MAIN MODULES

1.Import Packages

In [1]:
```
import numpy as np
import pandas as pd
```

2.Data Gathering and Understanding

In [43]:
```
loan_train= pd.read_csv('/Users/abhay/Downloads/loan_train .csv')
loan_test = pd.read_csv('/Users/abhay/Downloads/load_test 1.csv')
```

In [44]:
```
loan_train.head()
```

Out[44]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
0	LP001002	Male	No	0.0	Graduate	No	5849	0.0	NaN	360.0	1.0
1	LP001003	Male	Yes	1.0	Graduate	No	4583	1508.0	128.0	360.0	1.0
2	LP001005	Male	Yes	0.0	Graduate	Yes	3000	0.0	66.0	360.0	1.0
3	LP001006	Male	Yes	0.0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0
4	LP001008	Male	No	0.0	Graduate	No	6000	0.0	141.0	360.0	1.0

Lets display the some few information from our large datasets.Here, We shows the first five rows from datasets

In [45]:
```
loan_train
```

Out[45]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_Histo
0	LP001002	Male	No	0.0	Graduate	No	5849	0.0	NaN	360.0	1
1	LP001003	Male	Yes	1.0	Graduate	No	4583	1508.0	128.0	360.0	1
2	LP001005	Male	Yes	0.0	Graduate	Yes	3000	0.0	66.0	360.0	1
3	LP001006	Male	Yes	0.0	Not Graduate	No	2583	2358.0	120.0	360.0	1
4	LP001008	Male	No	0.0	Graduate	No	6000	0.0	141.0	360.0	1
...	...	...	...	...	...	...	...	...	...	...	
364	LP002180	Male	No	0.0	Graduate	Yes	6822	0.0	141.0	360.0	1
365	LP002181	Male	No	0.0	Not Graduate	No	6216	0.0	133.0	360.0	1
366	LP002187	Male	No	0.0	Graduate	No	2500	0.0	96.0	480.0	1
367	LP002188	Male	No	0.0	Graduate	No	5124	0.0	124.0	NaN	0
368	LP002190	Male	Yes	1.0	Graduate	No	6325	0.0	175.0	360.0	1

369 rows × 13 columns

```
In [49]: loan_train_columns = loan_train.columns
         loan_train_columns

Out[49]: Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
                'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
                'Loan_Amount_Term', 'Credit_History', 'Property_Area', 'Loan_Status'],
               dtype='object')
```

```
In [50]: loan_train.describe()
```

Out[50]:

	Dependents	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	358.000000	369.000000	369.000000	355.000000	357.000000	340.000000
mean	0.731844	5323.162602	1504.473713	145.225352	344.033613	0.858824
std	1.010113	5837.058539	1842.164096	83.756709	64.078244	0.348717
min	0.000000	150.000000	0.000000	17.000000	36.000000	0.000000
25%	0.000000	2927.000000	0.000000	100.000000	360.000000	1.000000
50%	0.000000	3858.000000	1229.000000	127.000000	360.000000	1.000000
75%	1.000000	5649.000000	2333.000000	168.000000	360.000000	1.000000
max	3.000000	63337.000000	11300.000000	700.000000	480.000000	1.000000

First of all we use the loan_train.describe() method to shows the important information from the dataset.It provides the count, mean, standard deviation (std), min, quartiles and max in its output

```
In [55]: import missingno as msno
```

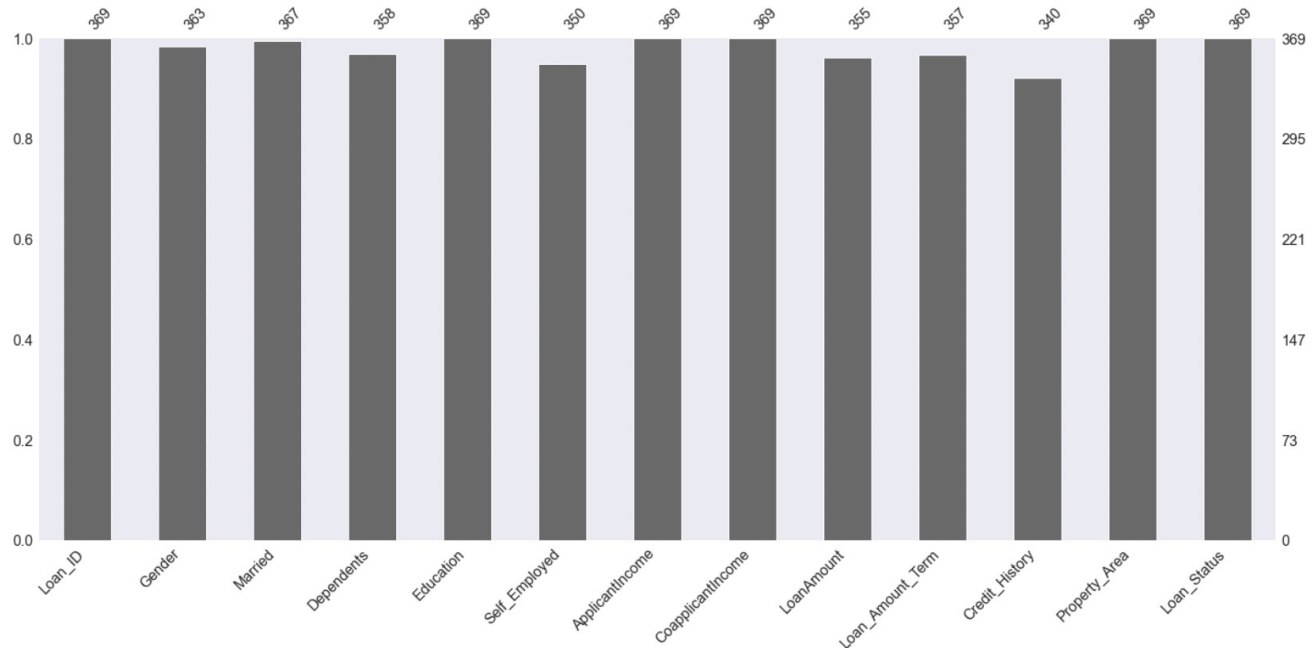
```
In [56]: loan_train

         loan_train.isna().sum()

Out[56]: Loan_ID               0
         Gender                6
         Married               2
         Dependents           11
         Education             0
         Self_Employed        19
         ApplicantIncome       0
         CoapplicantIncome     0
         LoanAmount           14
         Loan_Amount_Term     12
         Credit_History       29
         Property_Area         0
         Loan_Status           0
         dtype: int64
```

.

As we can see here, there are too many columns missing with small amount of null values so we use mean and mode to replace with NaN values.

In [57]:
```
msno.bar(loan_train)
```

Out[57]: <AxesSubplot:>



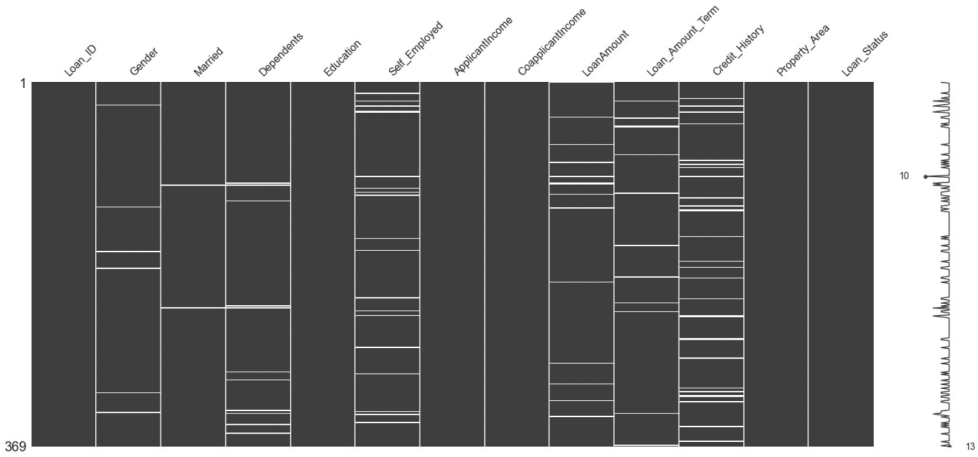
Loan_Status feature boolean values, So we replace Y values with 1 and N values with 0 and same for other Boolean types of columns

In [59]:
```
loan_train['Credit_History'].fillna(loan_train['Credit_History'].mode(), inplace=True)
loan_test['Credit_History'].fillna(loan_test['Credit_History'].mode(), inplace=True)


loan_train['LoanAmount'].fillna(loan_train['LoanAmount'].mean(), inplace=True)
loan_test['LoanAmount'].fillna(loan_test['LoanAmount'].mean(), inplace=True)
```

In [60]:
```
loan_train.Loan_Status = loan_train.Loan_Status.replace({"Y": 1, "N" : 0})


loan_train.Gender = loan_train.Gender.replace({"Male": 1, "Female" : 0})
loan_test.Gender = loan_test.Gender.replace({"Male": 1, "Female" : 0})

loan_train.Married = loan_train.Married.replace({"Yes": 1, "No" : 0})
loan_test.Married = loan_test.Married.replace({"Yes": 1, "No" : 0})

loan_train.Self_Employed = loan_train.Self_Employed.replace({"Yes": 1, "No" : 0})
loan_test.Self_Employed = loan_test.Self_Employed.replace({"Yes": 1, "No" : 0})
```

In [61]:
```
loan_train['Gender'].fillna(loan_train['Gender'].mode()[0], inplace=True)
loan_test['Gender'].fillna(loan_test['Gender'].mode()[0], inplace=True)

loan_train['Dependents'].fillna(loan_train['Dependents'].mode()[0], inplace=True)
loan_test['Dependents'].fillna(loan_test['Dependents'].mode()[0], inplace=True)

loan_train['Married'].fillna(loan_train['Married'].mode()[0], inplace=True)
loan_test['Married'].fillna(loan_test['Married'].mode()[0], inplace=True)

loan_train['Credit_History'].fillna(loan_train['Credit_History'].mean(), inplace=True)
loan_test['Credit_History'].fillna(loan_test['Credit_History'].mean(), inplace=True)
```

```
In [58]: msno.matrix(loan_train )
Out[58]: <AxesSubplot:>
```



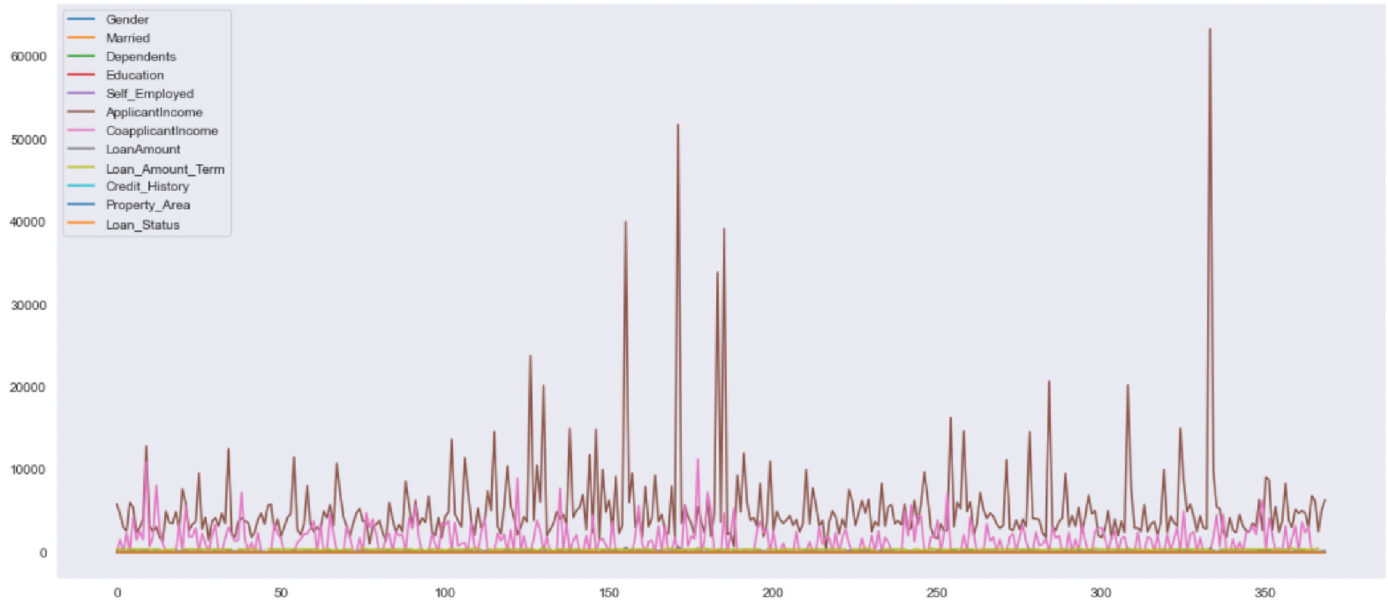
Here, Property_Area, Dependents and Education has multiple values so now we can use LabelEncoder from sklearn package

Data Understanding Graphs

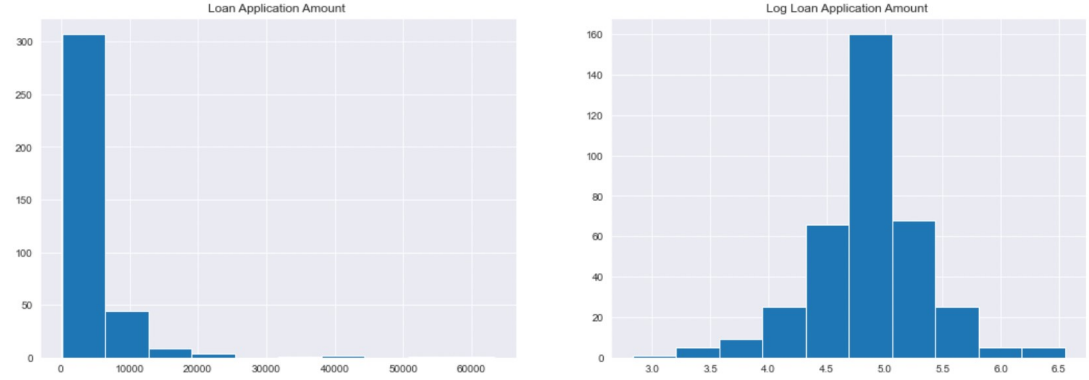
In [62]:
```
from sklearn.preprocessing import LabelEncoder
feature_col = ['Property_Area','Education', 'Dependents',]
le = LabelEncoder()
for col in feature_col:
    loan_train[col] = le.fit_transform(loan_train[col])
    loan_test[col] = le.fit_transform(loan_test[col])
```

In [63]:
```
import matplotlib.pyplot as plt
%matplotlib inline


import seaborn as sns
sns.set_style('dark')
```

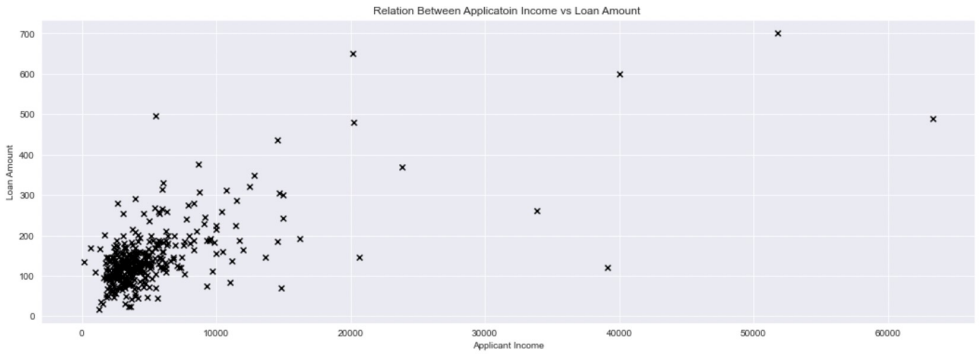
In [65]:
```
loan_train.plot(figsize=(18, 8))

plt.show()
```



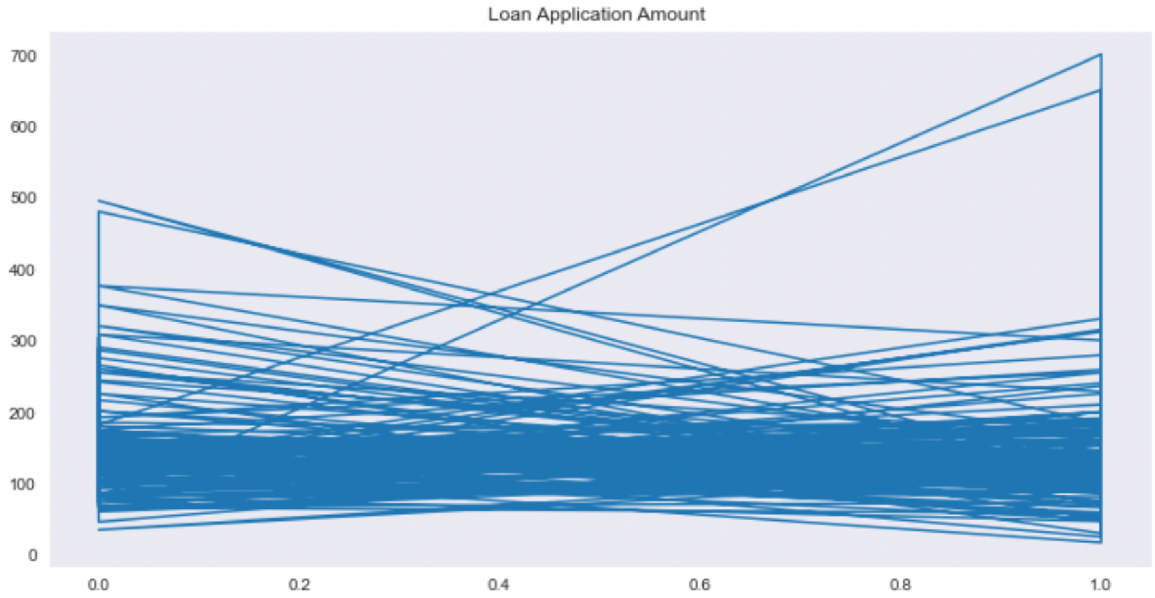
In [66]:
```
plt.figure(figsize=(18, 6))
plt.subplot(1, 2, 1)


loan_train['ApplicantIncome'].hist(bins=10)
plt.title("Loan Application Amount ")

plt.subplot(1, 2, 2)
plt.grid()
plt.hist(np.log(loan_train['LoanAmount']))
plt.title("Log Loan Application Amount ")

plt.show()
```



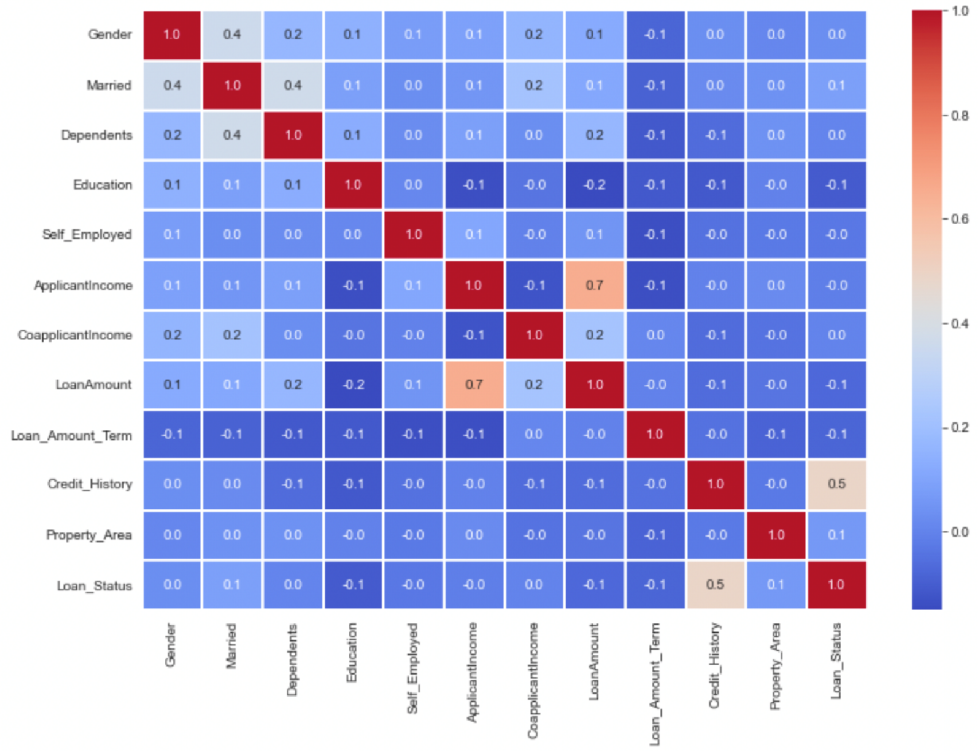
In [67]:
```
plt.figure(figsize=(18, 6))
plt.title("Relation Between Applicatoin Income vs Loan Amount ")

plt.grid()
plt.scatter(loan_train['ApplicantIncome'] , loan_train['LoanAmount'], c='k', marker='x')
plt.xlabel("Applicant Income")
plt.ylabel("Loan Amount")
plt.show()
```



8]:
```
plt.figure(figsize=(12, 6))
plt.plot(loan_train['Loan_Status'], loan_train['LoanAmount'])
plt.title("Loan Application Amount ")
plt.show()
```



In [69]:
```
plt.figure(figsize=(12,8))
sns.heatmap(loan_train.corr(), cmap='coolwarm', annot=True, fmt='.1f', linewidths=.1)
plt.show()
```



5.Training the model

In this step, We have a lots of Machine Learning Model from sklearn package, and we need to decide which model is give us the better performance. then we use that model in final stage and send to the production level.

In [70]:
```
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score
```

In [71]:
```
logistic_model = LogisticRegression()
```

In [72]:
```
train_features = ['Credit_History', 'Education', 'Gender']

x_train = loan_train[train_features].values
y_train = loan_train['Loan_Status'].values

x_test = loan_test[train_features].values
```

In [73]:
```
logistic_model.fit(x_train, y_train)
```

Out[73]:
```
LogisticRegression()
```

In [74]:
```
predicted = logistic_model.predict(x_test)
```

In [75]:
```
print('Coefficient of model :', logistic_model.coef_)
```

```
Coefficient of model : [[ 2.8089175  -0.36490391  0.16462381]]
```

In [76]:
```
print('Intercept of model',logistic_model.intercept_)
```

```
Intercept of model [-1.64433053]
```

In [77]:
```
score = logistic_model.score(x_train, y_train)
print('accuracy_score overall :', score)
print('accuracy_score percent :', round(score*100,2))
```

```
accuracy_score overall : 0.7913279132791328
accuracy_score percent : 79.13
```

RESULTS

Effiency-79.13%

```
In [117]: predict_test = logistic_model.predict(x_test)
          print('Target on test data',predict_test)
```

```
Target on test data [1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 1
 1 1 0 0 1 1 0 1 0 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1
 1 1 1 1 1 1 0 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1
 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 0
 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0 1
 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0]
```

```
In [118]: predict_test
```

```
Out[118]: array([1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
                 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0,
                 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
                 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1,
                 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
                 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
                 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1,
                 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
                 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                 1, 1, 0])
```

```
In [76]: loan_test['predicted result'] = data_p
         loan_test
```

Out[76]:

Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	predicted result
Yes	0.0	Graduate	No	19730	5266.0	570.0	360.0	1.0	Rural	Y
No	0.0	Graduate	Yes	15759	0.0	55.0	360.0	1.0	Semiurban	Y
Yes	2.0	Graduate	No	5185	0.0	155.0	360.0	1.0	Semiurban	Y
Yes	2.0	Graduate	Yes	9323	7873.0	380.0	300.0	1.0	Rural	Y
No	1.0	Graduate	No	3062	1987.0	111.0	180.0	0.0	Urban	N
...	...	...	...	...	...	...	...	...	...	...
No	0.0	Graduate	No	2900	0.0	71.0	360.0	1.0	Rural	Y
Yes	3.0	Graduate	No	4106	0.0	40.0	180.0	1.0	Rural	Y
Yes	1.0	Graduate	No	8072	240.0	253.0	360.0	1.0	Urban	Y
Yes	2.0	Graduate	No	7583	0.0	187.0	360.0	1.0	Urban	Y
No	0.0	Graduate	Yes	4583	0.0	133.0	360.0	0.0	Semiurban	N

CONCLUSION

Future Work

In future, this model can be used to compare various machine learning algorithm generated prediction models and the model which will give higher accuracy will be chosen as the prediction model.

After this work, we are able to conclude that Decision tree version is extraordinary efficient and gives a higher end result. We have developed a model which can easily predict that the person will repay its loan or not. we can see our model has reduced the efforts of bankers. Machine learning has helped a lot in developing this model which gives precise results.

We were successfully able to predict the loan status for around 700 entries using our trained model.

THANK YOU