AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

MATHEMATICS FOR INTELLIGENT SYSTEM

PROJECT:
LOAN PREDICTION

GROUP 6

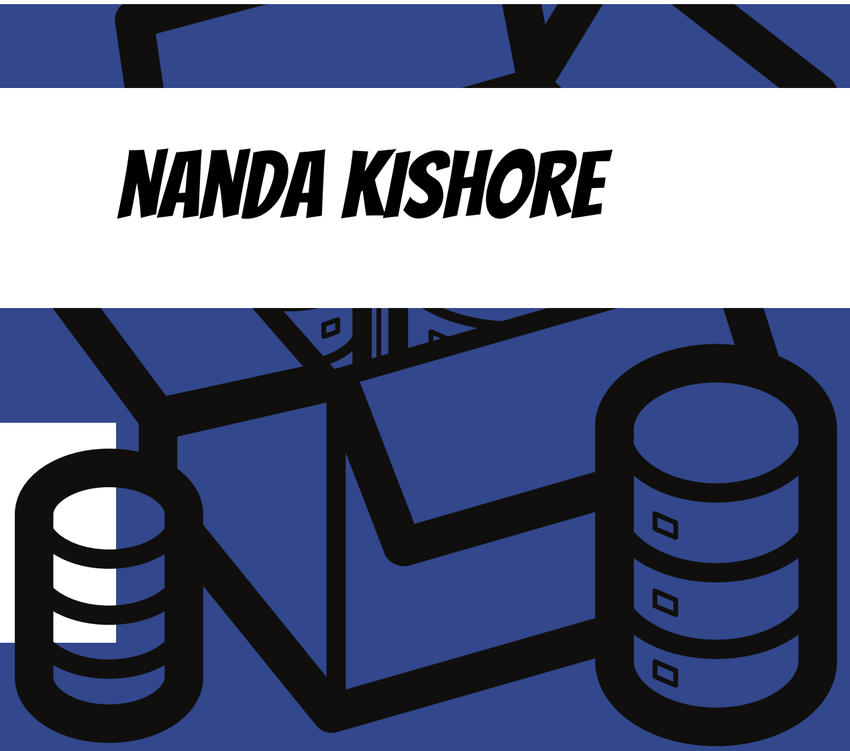
THATI AYYAPPA SWAMY

MADIRAJU BHARADWAJ

ANUGRAH NAMBIAR

NANDA KISHORE

ANUPA SAJIKUMAR



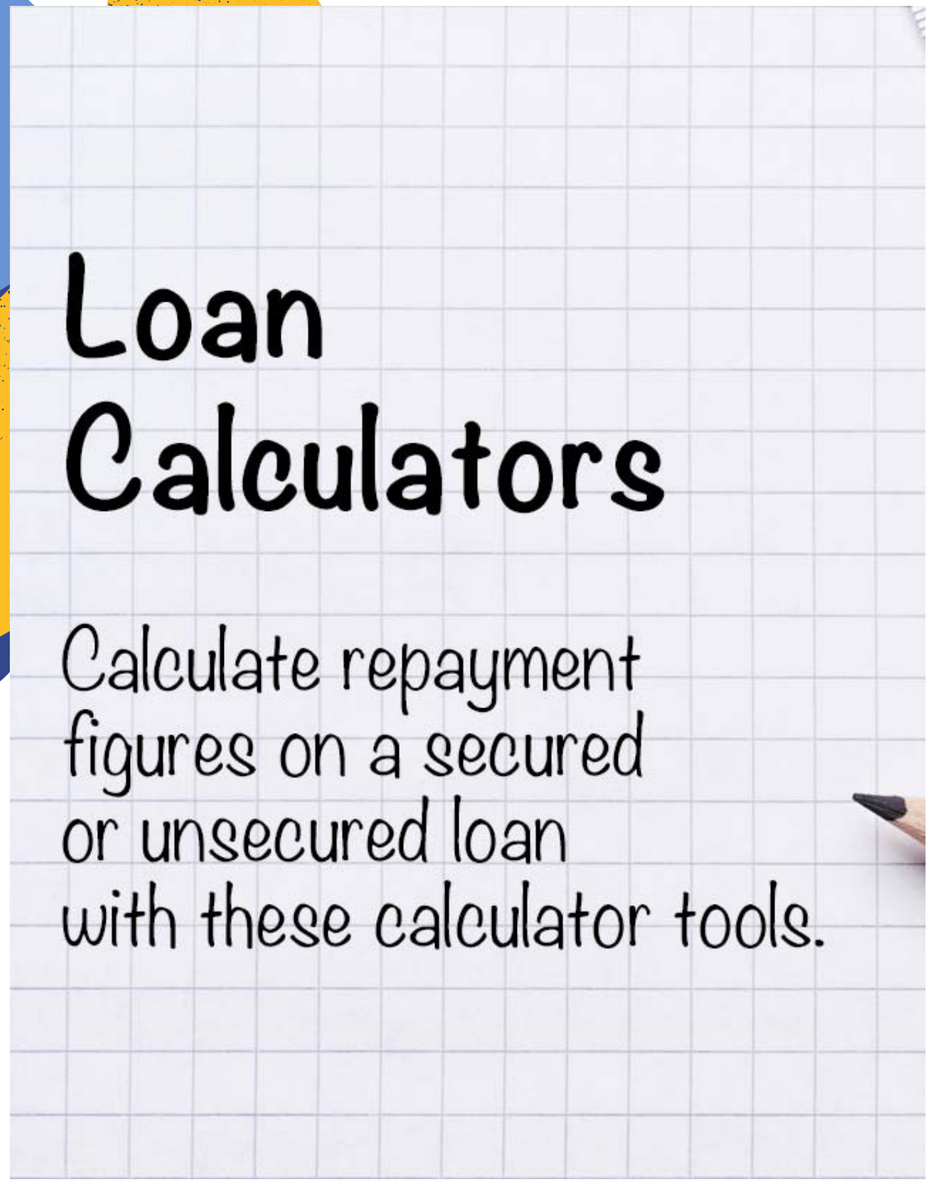
INTRODUCTION

Loans are the main business for banks. The main profit comes directly from the loan's interest. The loan companies grant a loan after an intensive process of verification and prediction. However, they still don't have assurance if the person is able to repay the loan with no difficulties.

we'll build a predictive model to predict if an person is able to repay the lending company or not. We will prepare the data using Jupyter Notebook/collab and use various models to predict the target variable.

OBJECTIVE

We are going to build an application using python libraries. Which is going to help the whole bank loan approval methodology. By getting the data from the person itself and then verifying & predicting the data, then concluding whether loan should be given to the person or not, if yes how much is the person able to get from the bank. So that they can predict by what time period will the person be able to give the whole money back.



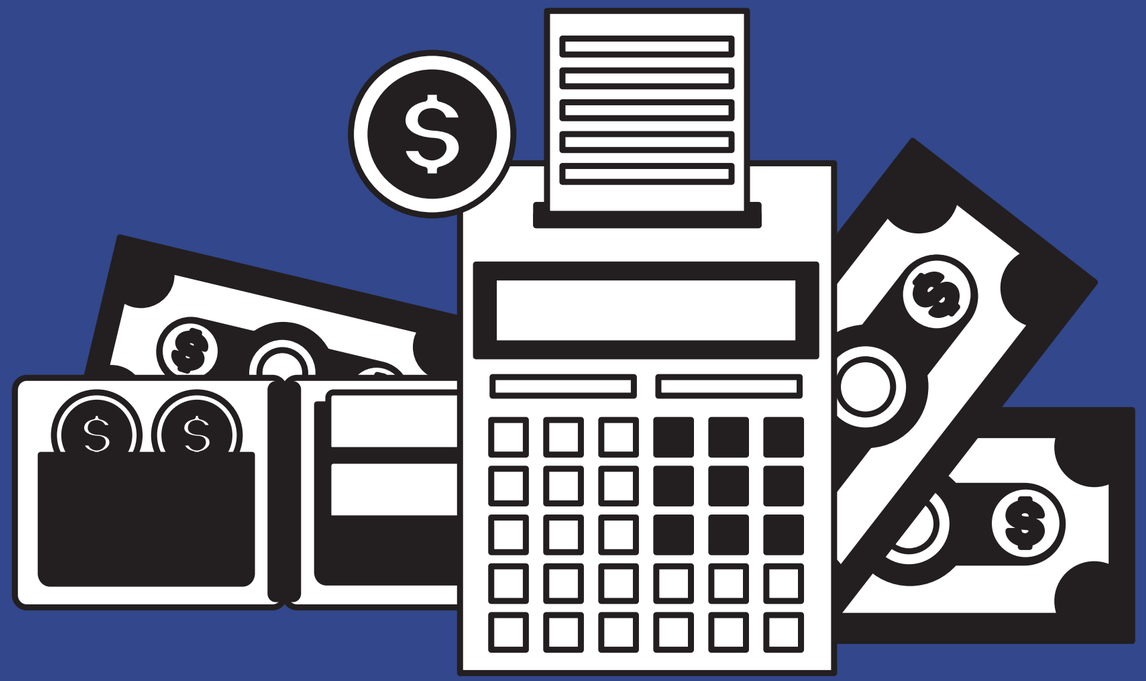
ABSTRACT

here we are going to using few libraries like

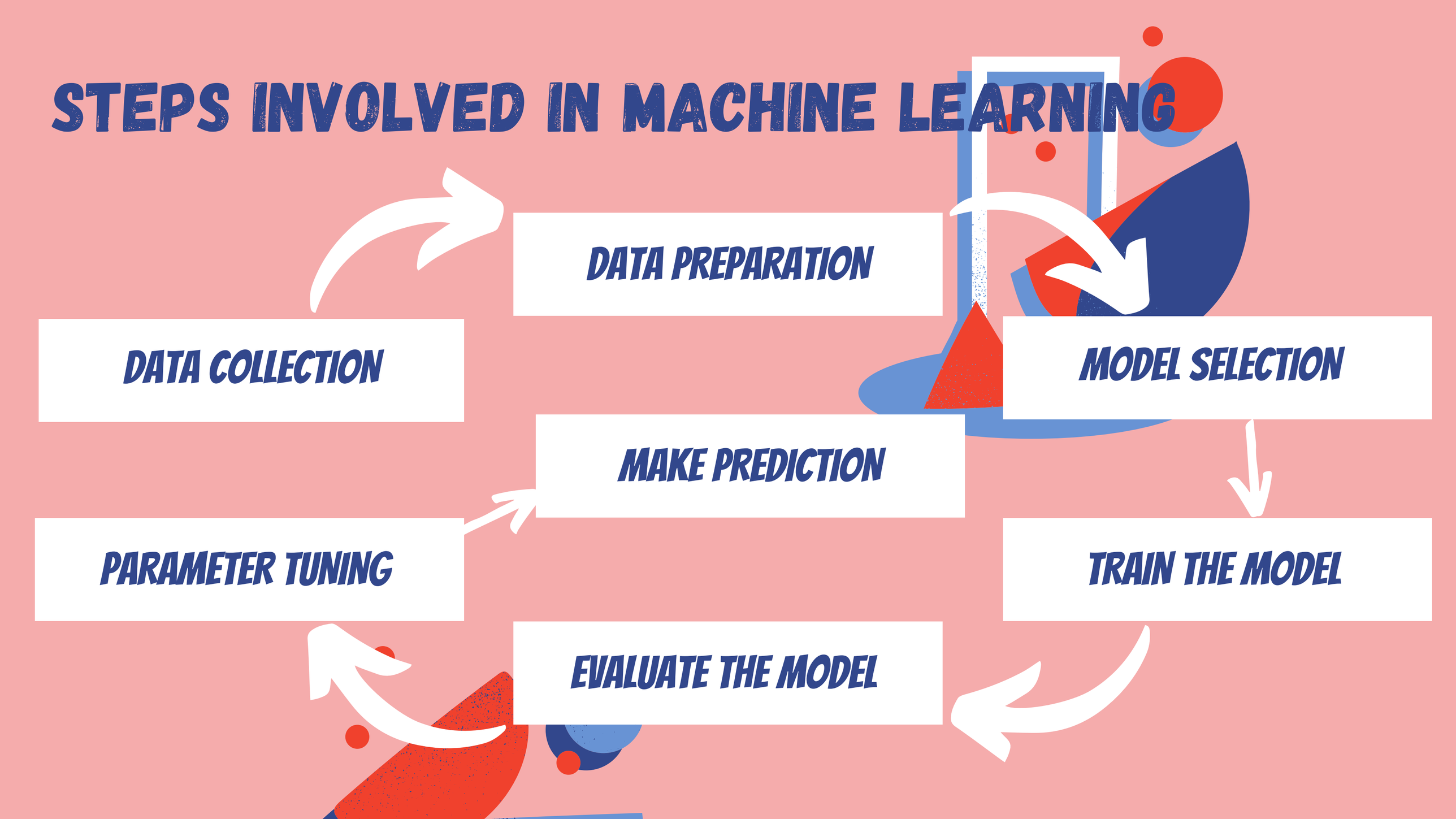
- python
- pandas
- numpy
- seaborn
- sklearn etc..

Data

For this problem, we have three CSV files: train, test, and sample submission.



STEPS INVOLVED IN MACHINE LEARNING



DATA SETS

- Dataset description
- Data Exploration Using Pig and Hive
- Data Exploration MatplotLib in PySpark
- Data Cleaning and Missing Values Imputation
- Developing Different Classification Models
- Comparison of Accuracy scores
- Building the Final Model

- Here we have two datasets. First is train_dataset.csv, test_dataset.csv.
- These are datasets of loan approval applications which are featured with annual income, married or not, dependents are there or not, educated or not, credit history present or not, loan amount etc.
- The outcome of the dataset is represented by loan_status in the train dataset.
- This column is absent in test_dataset.csv as we need to assign loan status with the help of training dataset.
- These two datasets are already uploaded on google colab.

USER INPUTS

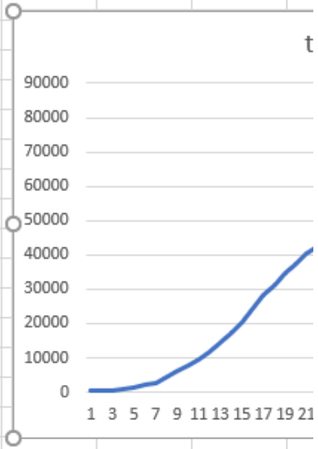
- We are going to train the data algorithm via train.csv,
- here the variable will be the user inputs, where loan id is a string, gender a boolean, married a boolean etc. based on the user input the algorithm is going to build a decision tree and check whether the person is eligible to get the loan or not.
- The inputs will be taken to the test case and predict the applicants capacity of loan.
- later on the inputs will be taken to test.csv.

Variable	Description
Loan_ID	Unique Loan ID
Gender	Male/ Female
Married	Applicant married (Y/N)
Dependents	Number of dependents
Education	Applicant Education (Graduate/ Under Graduate)
Self_Employed	Self employed (Y/N)
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)

PROBLEMS

- We need more data to analyse the logistic regression.
- The accuracy for the loan approval analysis not that perfect. It may be wrong sometimes and does't valuate properly.
- Few sets like if a person paid the previous loans on time with no issues then the algorithm shows that the person is completely eligible. but in that case he can also be able to ditch the loan payment in th 11th time.

	A	B	C	D	E	F
1	date	location	new_case	new_deat	total_case	total_deaths
2	21/01/2020	China			278	6
3	22/01/2020	China	32	0	310	6
4	23/01/2020	China	264	11	574	17
5	24/01/2020	China	261	8	835	25
6	25/01/2020	China	462	16	1297	41
7	26/01/2020	China	688	15	1985	56
8	27/01/2020	China	776	24	2761	80
9	28/01/2020	China	1776	26	4537	106
10	29/01/2020	China	1460	26	5997	132
11	30/01/2020	China	1739	38	7736	170
12	31/01/2020	China	1984	43	9720	213
13	01/02/2020	China	2101	46	11821	259
14	02/02/2020	China	2590	45	14411	304
15	03/02/2020	China	2827	57	17238	361
16	04/02/2020	China	3233	64	20471	425
17	05/02/2020	China	3892	66	24363	491
18	06/02/2020	China	3697	73	28060	564
19	07/02/2020	China	3151	73	31211	637
20	08/02/2020	China	3387	86	34598	723
21	09/02/2020	China	2653	89	37251	812
22	10/02/2020	China	2984	97	40235	909
23	11/02/2020	China	2473	108	42708	1017
24	12/02/2020	China	2022	97	44730	1114
25	13/02/2020	China	1820	254	46550	1368
26	14/02/2020	China	1998	13	48548	1381
27	15/02/2020	China	1506	143	50054	1524
28	16/02/2020	China	1120	142	51174	1666
29	17/02/2020	China	19461	106	70635	1772
30	18/02/2020	China	1893	98	72528	1870
31	19/02/2020	China	1752	136	74280	2006
32	20/02/2020	China	395	115	74675	2121
33	21/02/2020	China	894	118	75569	2239



SOLUTION/PROBLEM APPROACH

1). SOLUTION

we can give the logistic regression data directly from the user it self be can saved. From the first person. The data given by the person and save for the further applicants

2). SOLUTION

There is no perfect solution for but we can just verify the person manually and permitting it.

DATA/VISUALISATION

We are going going to save both python scratch code and python note book code in the GitHub. we are going to implement it in the jupyter notebook or google collab. The GitHub link is mentioned below:

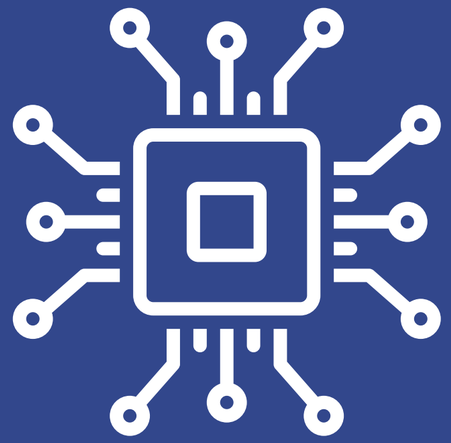
https://github.com/AyyappaSwamy-Sam/LoanPrediction-S1

The working system and the Loan Prediction will be more detailed in the GitHub.

more precisely the working will be explained more detail

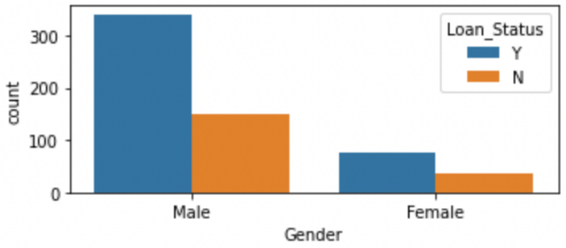
IMPLEMENTATION

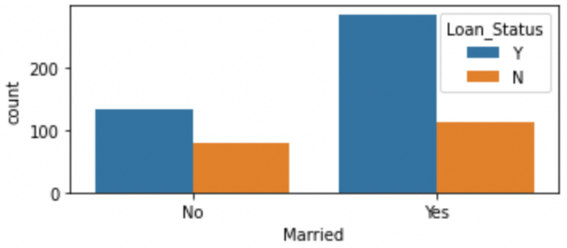
- LOAN_ID : STRING
- GENDER : BOOLEAN
- MARRIED : BOOLEAN
- DEPENDENTS : BOOLEAN
- EDUCATION : STRING
- SELF_EMPLOYED : BOOLEAN
- APPLICANT INCOME : INT
- CO-APPLICANT INCOME : INT
- LOAN_AMOUNT_TERM : FLOAT 64
- CREDIT HISTORY : FLOAT64
- PROPERTY AREA : STRING
- LOAN_STATUS : OBJECT

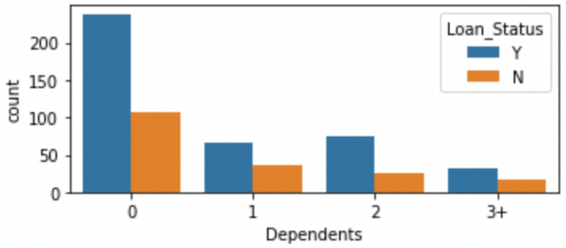


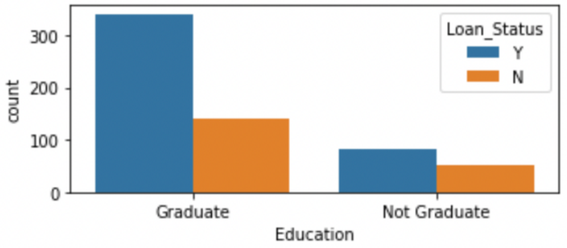
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            614 non-null    object 
 1   Gender             601 non-null    object 
 2   Married            611 non-null    object 
 3   Dependents         599 non-null    object 
 4   Education          614 non-null    object 
 5   Self_Employed      582 non-null    object 
 6   ApplicantIncome    614 non-null    int64  
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object 
 12  Loan_Status        614 non-null    object 
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

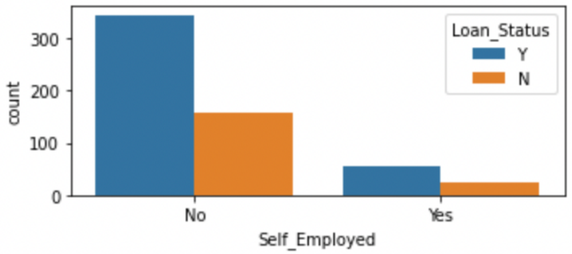
SEA BORN & MATPLOTLIB

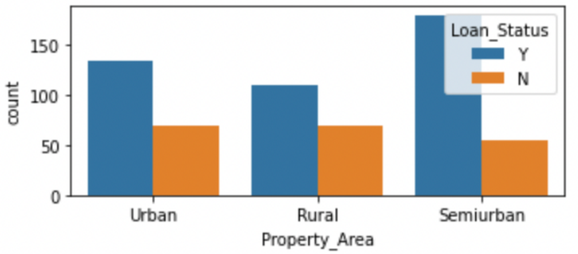


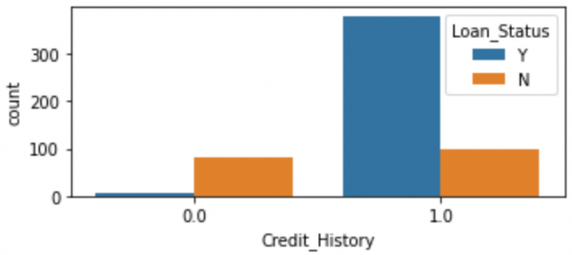


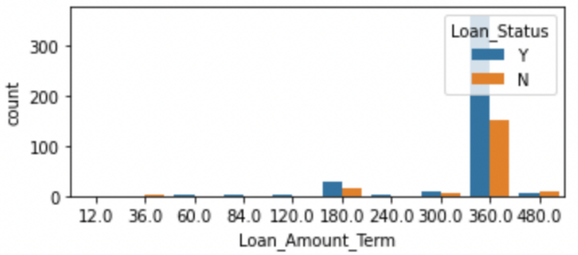




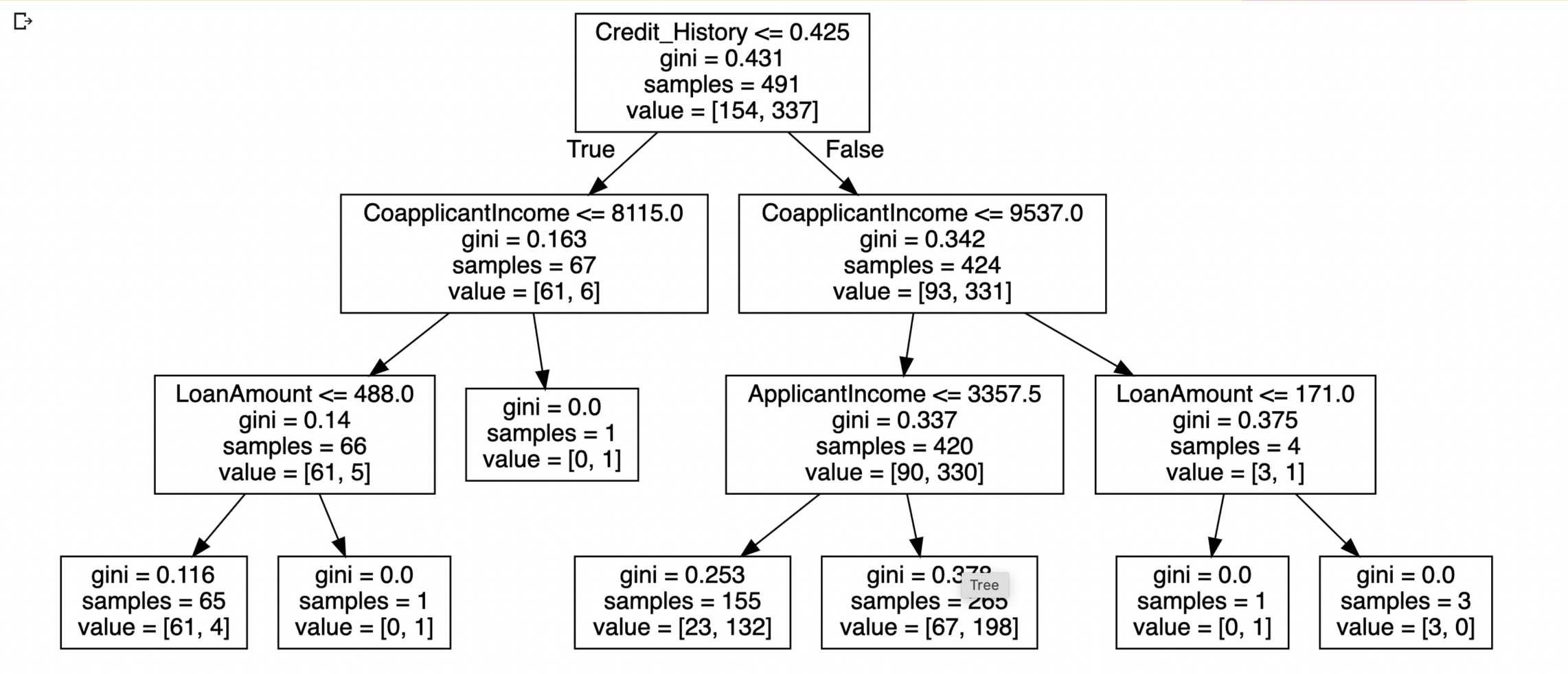






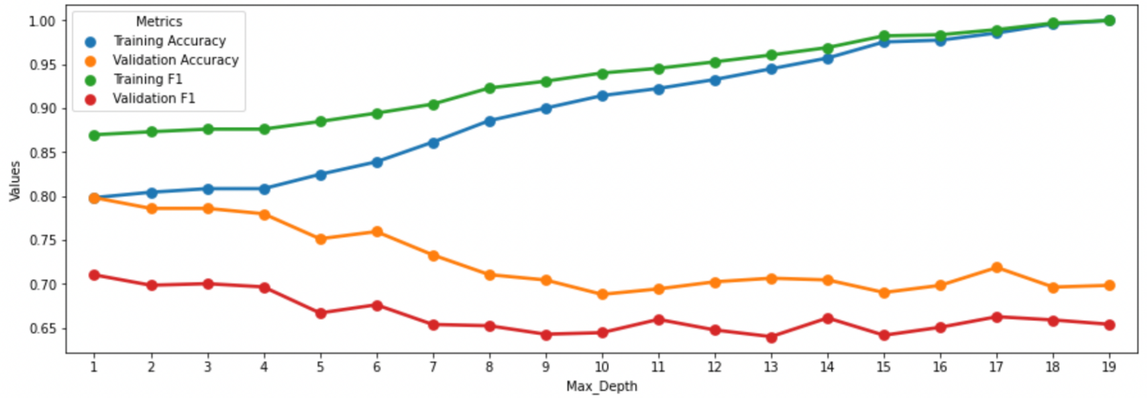


DECISION TREE



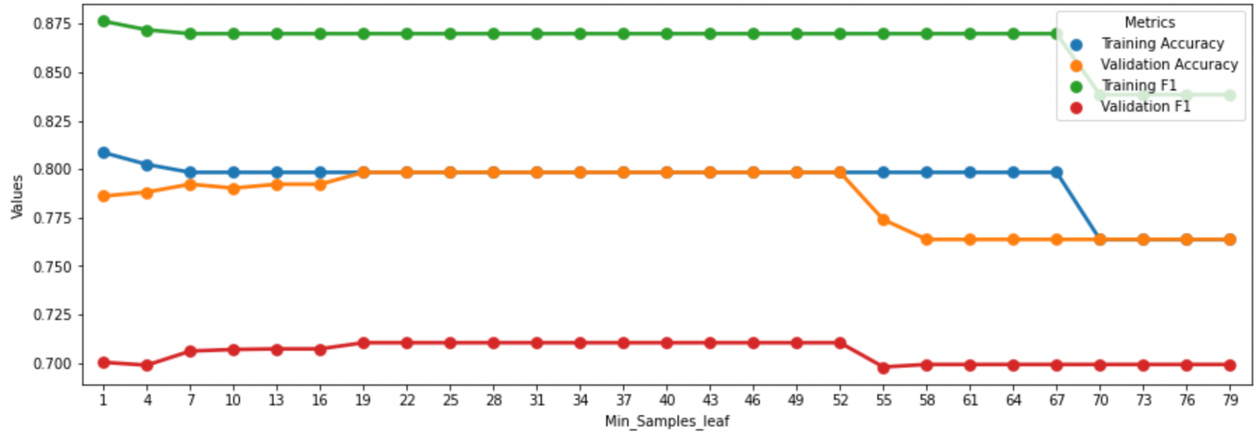
RETURN OF RESULTS IN GRAPHS

<matplotlib.axes._subplots.AxesSubplot at 0x7f2ebdc5e550>



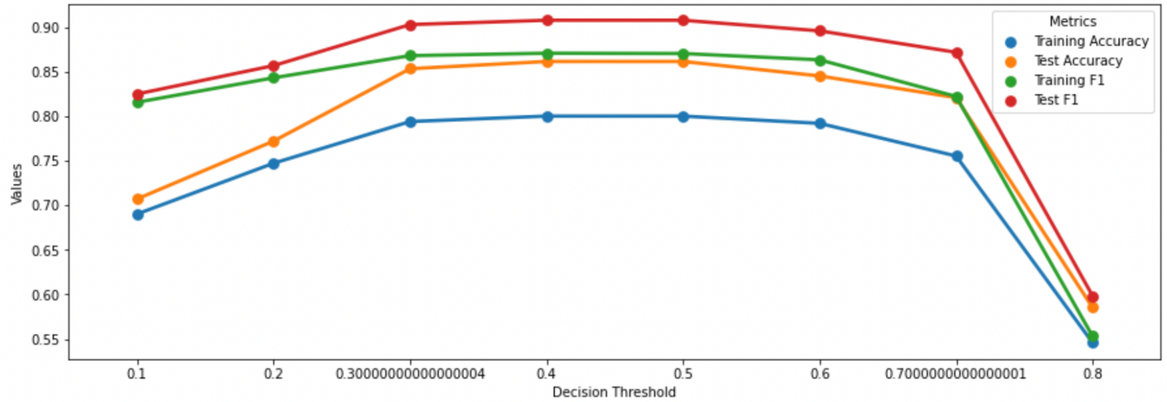
SITUATION 1

<matplotlib.axes._subplots.AxesSubplot at 0x7f2ebdccc190>



SITUATION 2

<matplotlib.axes._subplots.AxesSubplot at 0x7f2ebdfeedd0>



SITUATION 3

OBSERVATIONS

Here when we observe that are different outputs for different kind of situations. When the user gives the input for the code it analyse if he is married or not, have dependencies or not, if yes how many of the depend on the person. what is the income and loan history, then finalises and gives the result if the person is eligible

The input taken from the user will be executed in different formats of code in different situations as mentions earlier. And gives the result with respect to that, (like how much is the applicant eligible).
And finally the accuracy.

RESULTS

Test Accuracy: 0.8617886178861789
Test F1 Score: 0.9081081081081082
Confusion Matrix on Test Data

Predicted	0	1	All
True			
0	22	16	38
1	1	84	85
All	23	100	123

SITUATION 1

Test Accuracy: 0.8536585365853658
Test F1 Score: 0.903225806451613
Confusion Matrix on Test Data

Predicted	0	1	All
True			
0	21	17	38
1	1	84	85
All	22	101	123

SITUATION 3

SITUATION 2

Training Data Set Accuracy: 1.0
Training Data F1 Score 1.0
Validation Mean F1 Score: 0.627815620492227
Validation Mean Accuracy: 0.708740465883323

REFERENCES

https://www.kaggle.com/ajaymanwani/loan-approval-prediction/notebook

https://youtu.be/T9kgWBmUIRk