



AMARITA
VISHWA VIDYAPEETHAM

MATHEMATICS FOR INTELLIGENT SYSTEM

PROJECT:

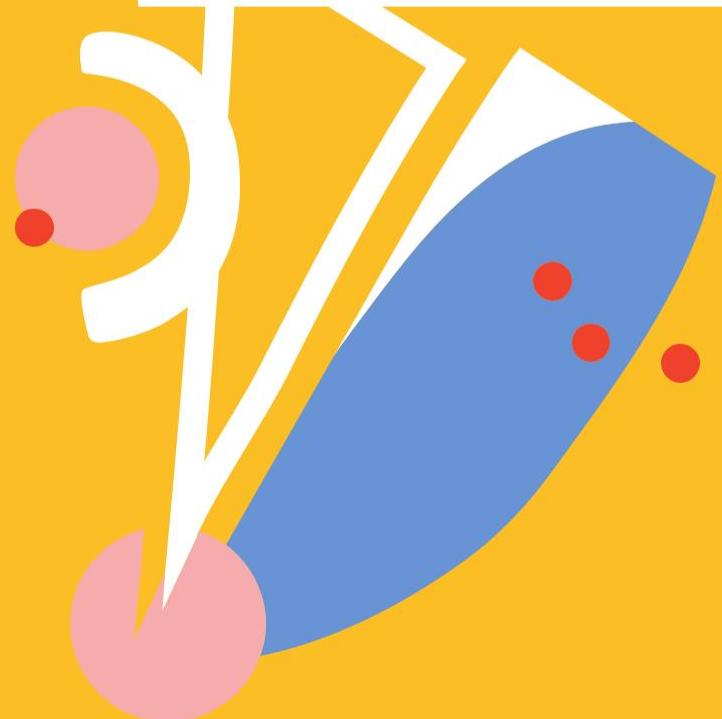
**LOAN PREDICTION USING DECISION TREE,
RANDOM FOREST & LOGISTIC REGRESSION**

BY:

- MADIRAJU BHARADWAJ-41
- AYYAPPA SWAMY-84
- ANUGRAH NAMBIAR-16
- NANDAKISHORE-45
- ANUPA SAJI KUMAR-71

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.

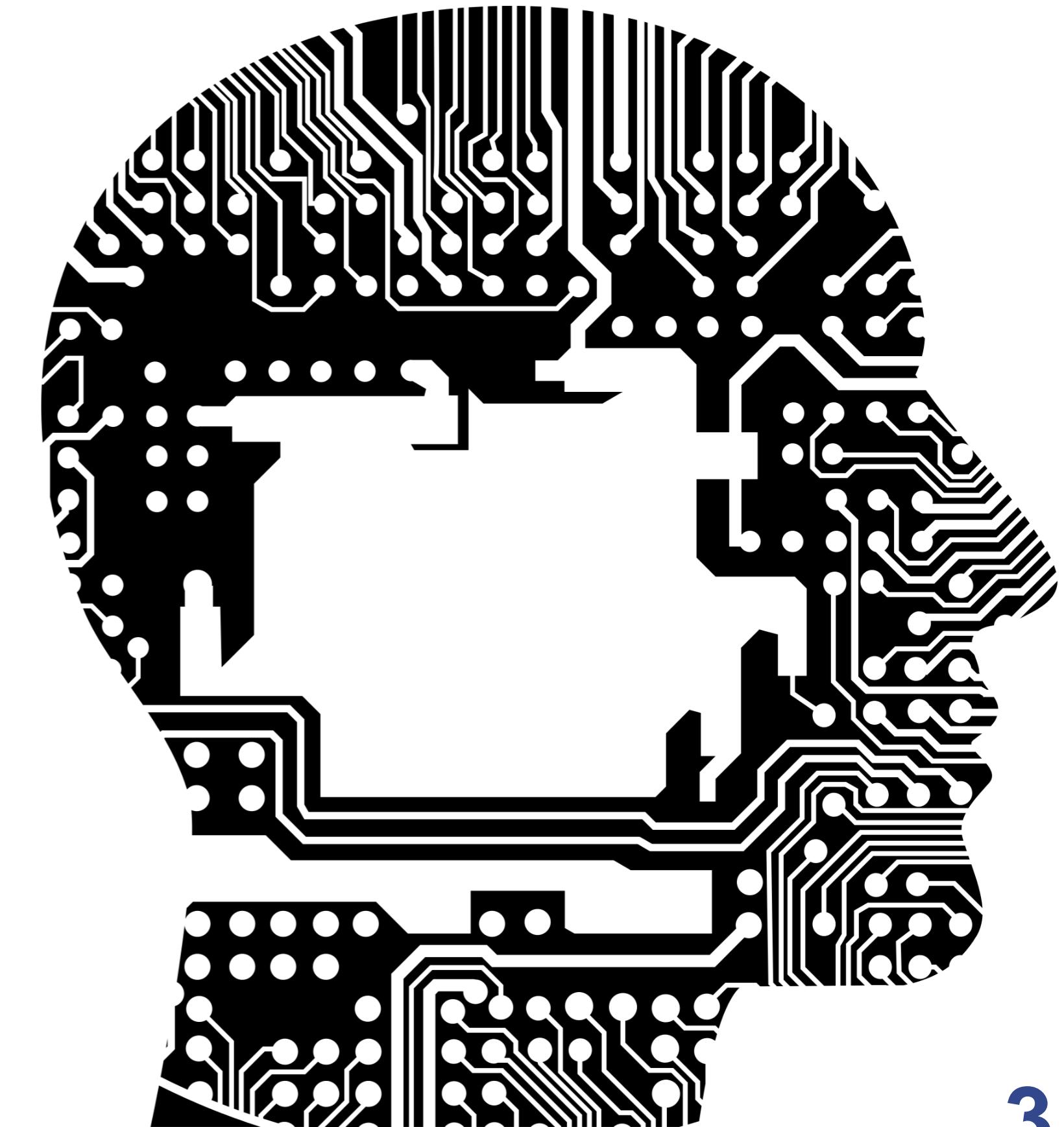


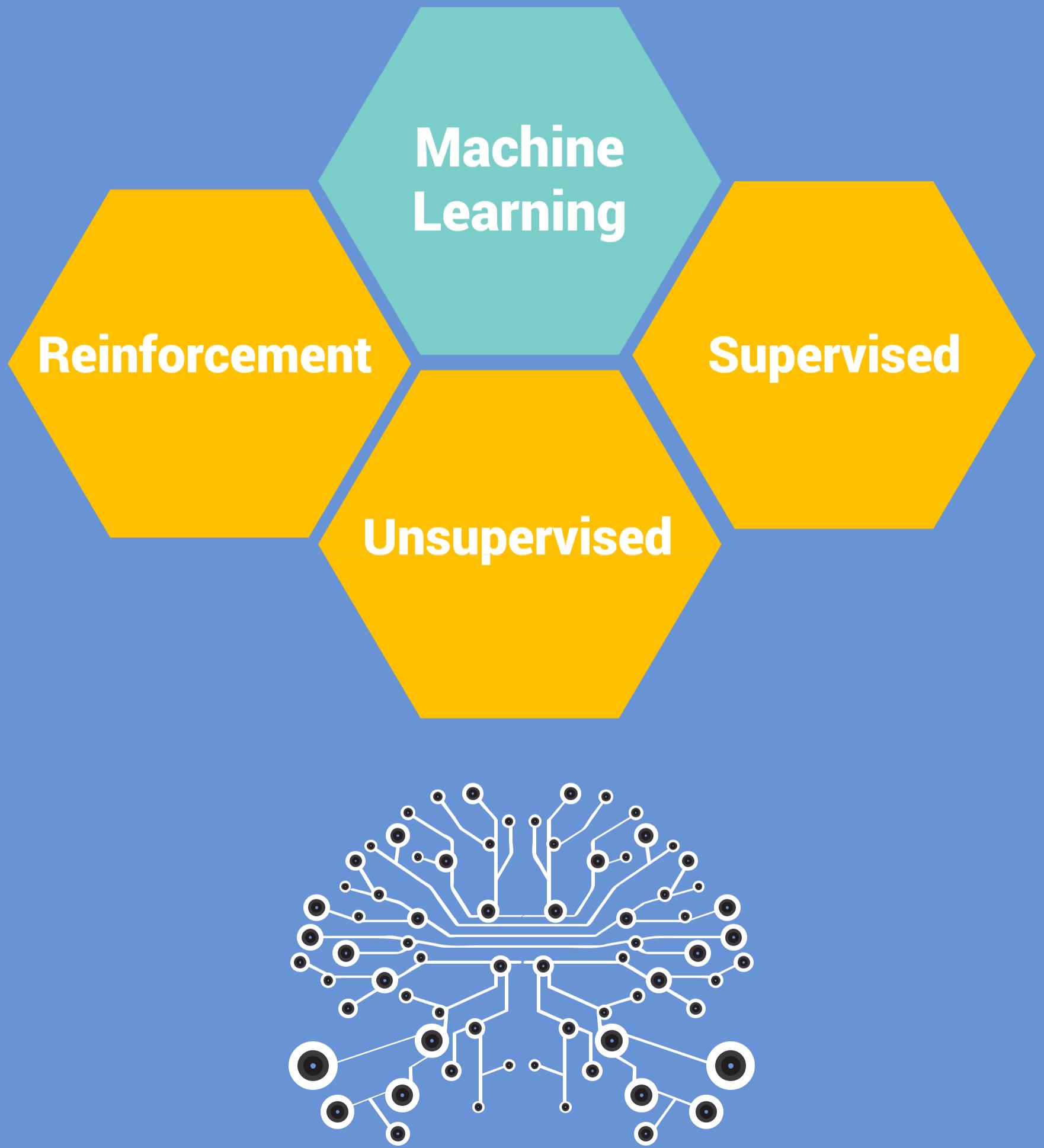
we'll build a predictive model to predict if a person is able to repay the lending company or not. We will prepare the data using Jupyter Notebook/collab and use various models to predict the target variable.

WHAT IS MACHINE LEARNING?

Machine Learning is the process of creating models which can perform a certain task without the need of human explicitly programming to do something.

Machine learning is important because it gives enterprises a view of trends in customer behavior and business operational patterns, as well as supports the development of new products.





SUPERVISED LEARNING

Supervised learning is an approach to creating artificial intelligence (AI), where a computer algorithm is trained on input data that has been labeled for a particular output. The model is trained until it can detect the underlying patterns and relationships between the input data and the output labels, enabling it to yield accurate labeling results when presented with never-before-seen data.

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.

Loan Calculators

Calculate repayment figures on a secured or unsecured loan with these calculator tools.

abstract

Data

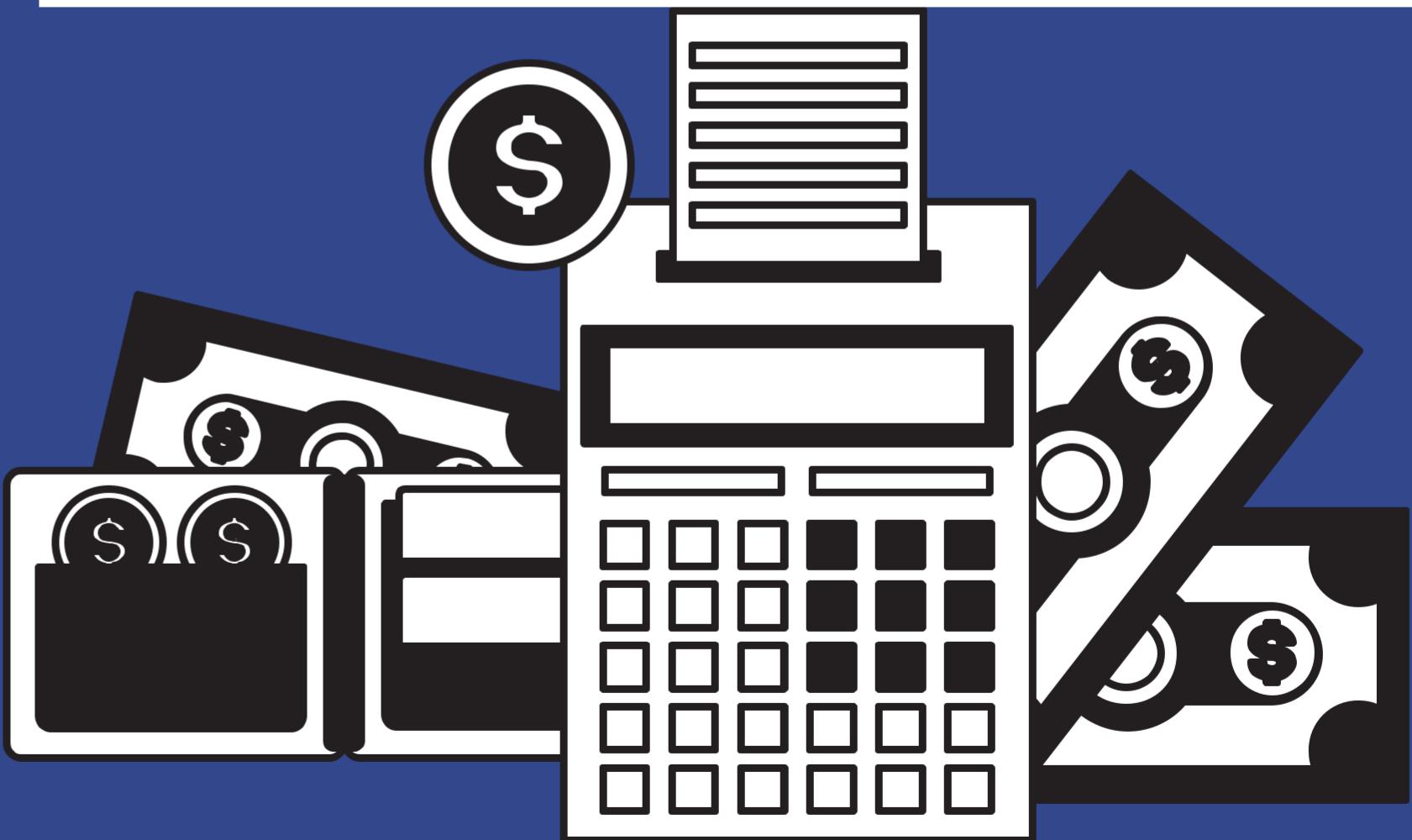
For this problem, we have two CSV files: train, and test.

we going to analyse in three models:

- logistic regression.
- random forest
- decision tree.

here we are going to using few libraries like

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

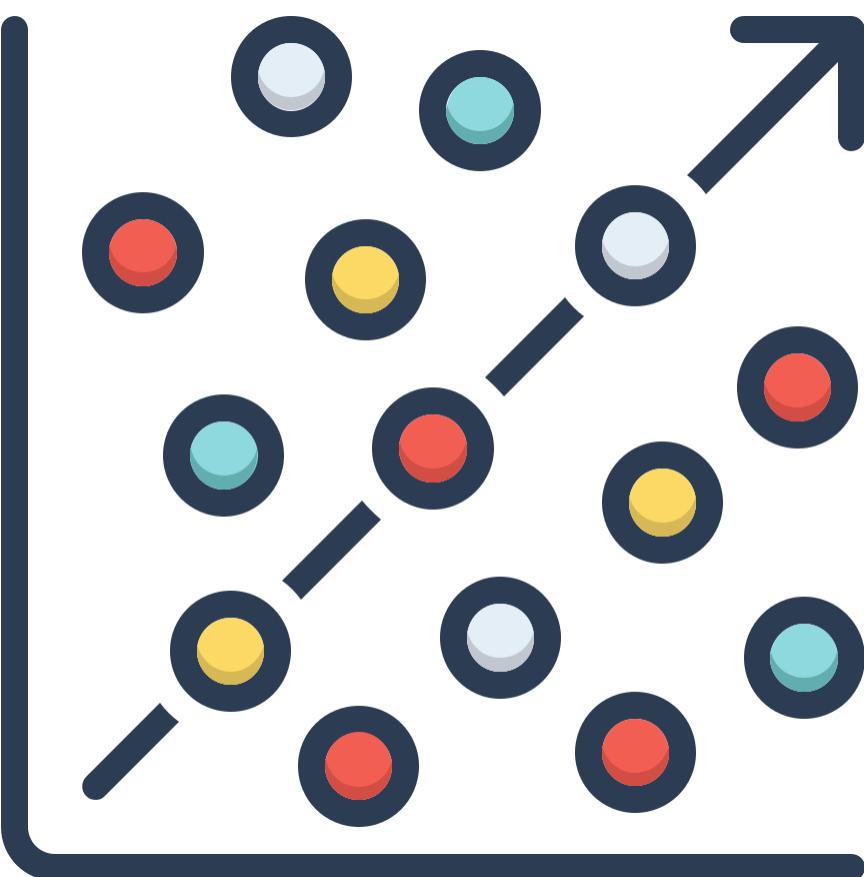


STEPS INVOLVED IN MACHINE LEARNING

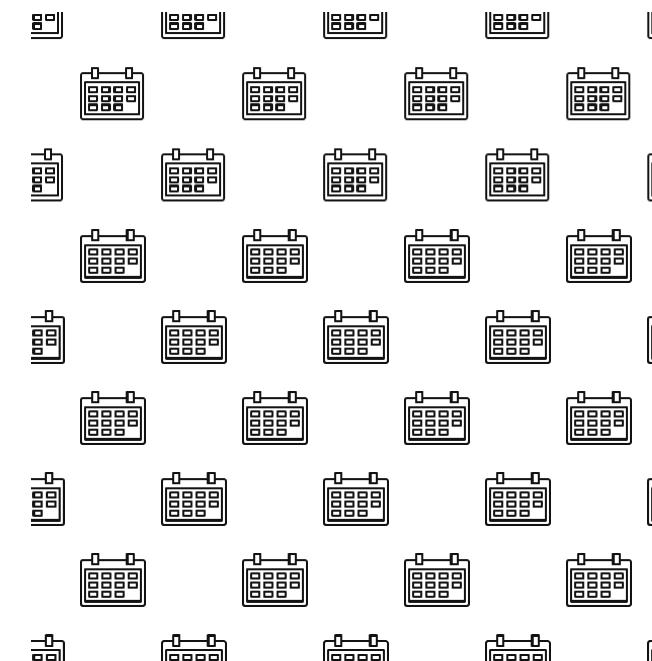


METHODS USED

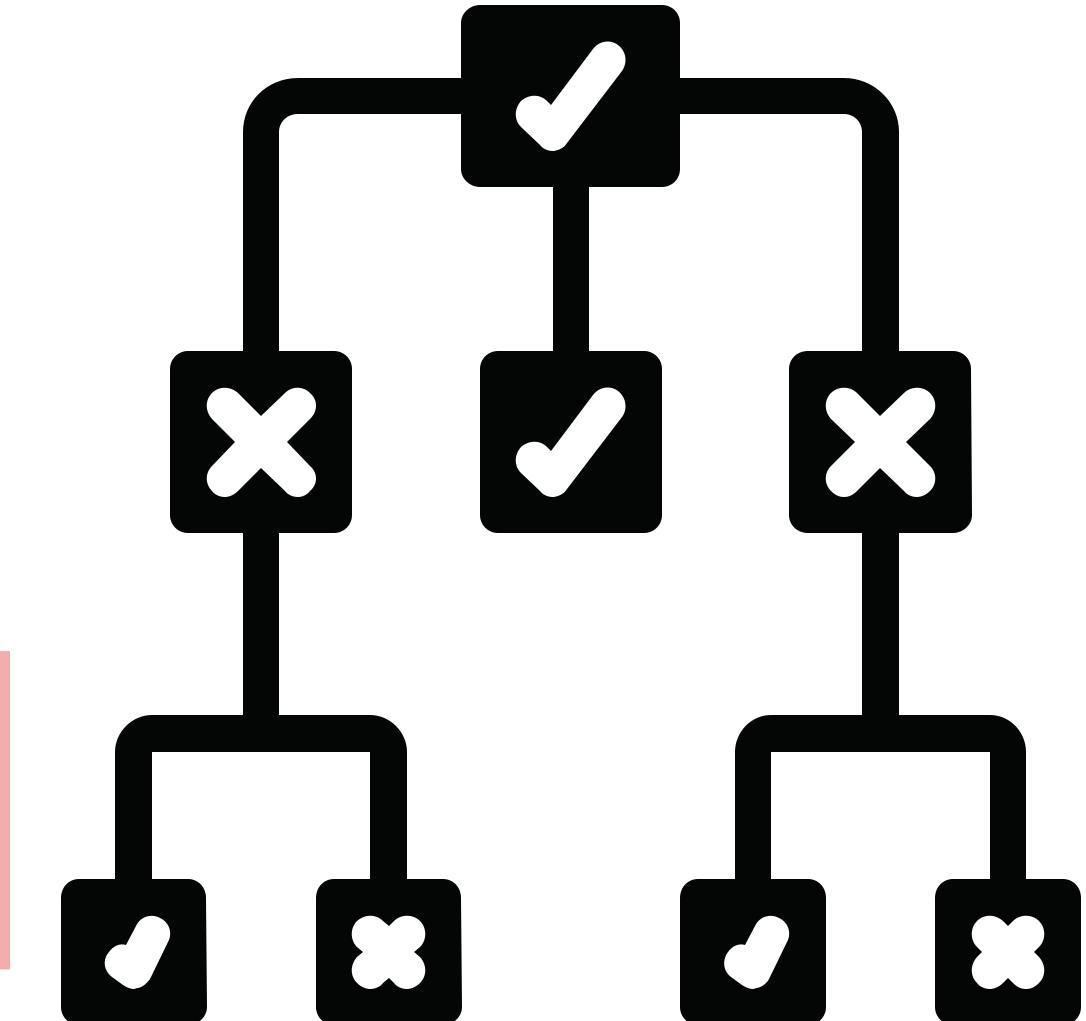
LOGISTIC
REGRESSION



RANDOM FOREST

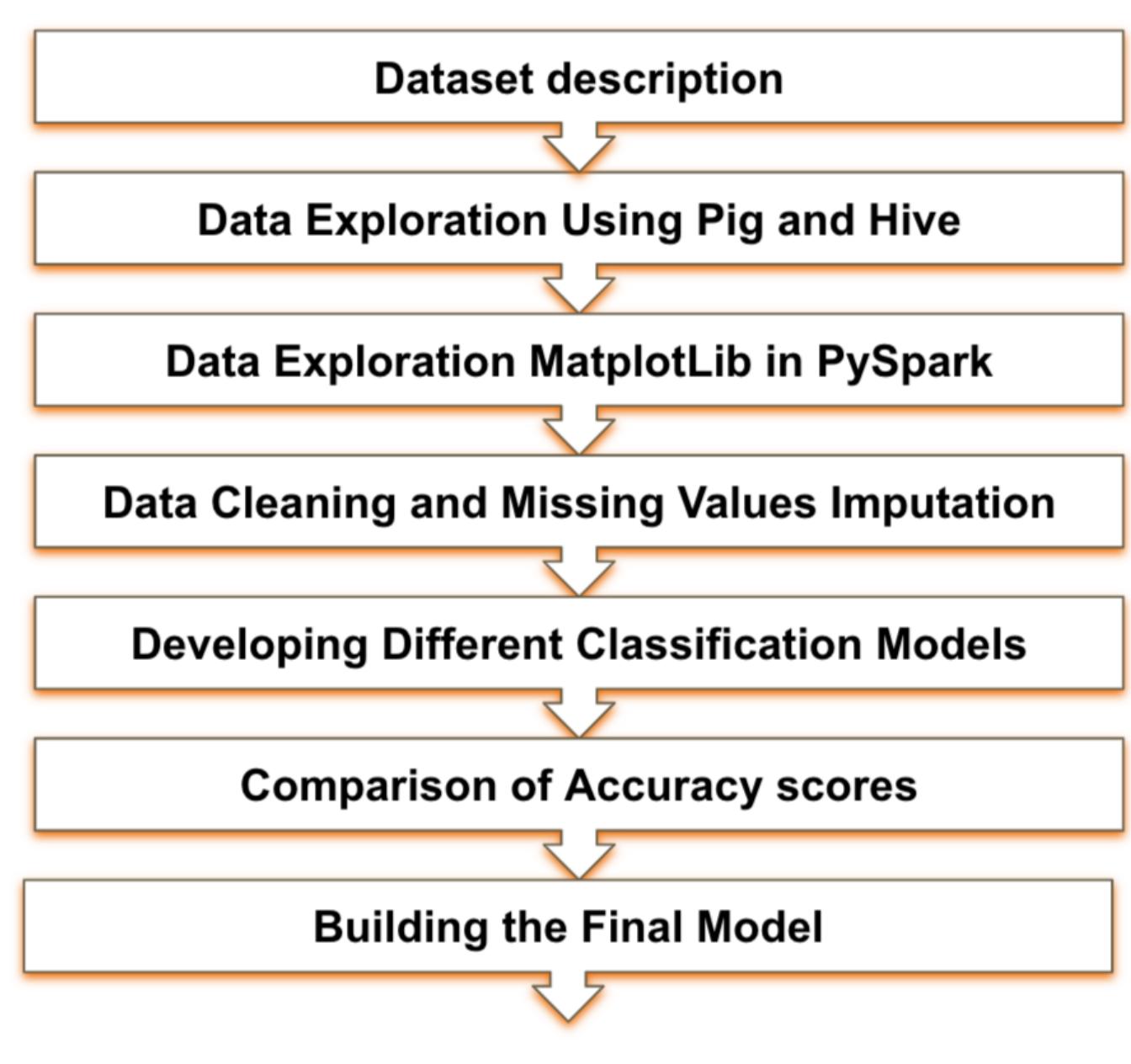


DECISION TREE





DATA SETS



- 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

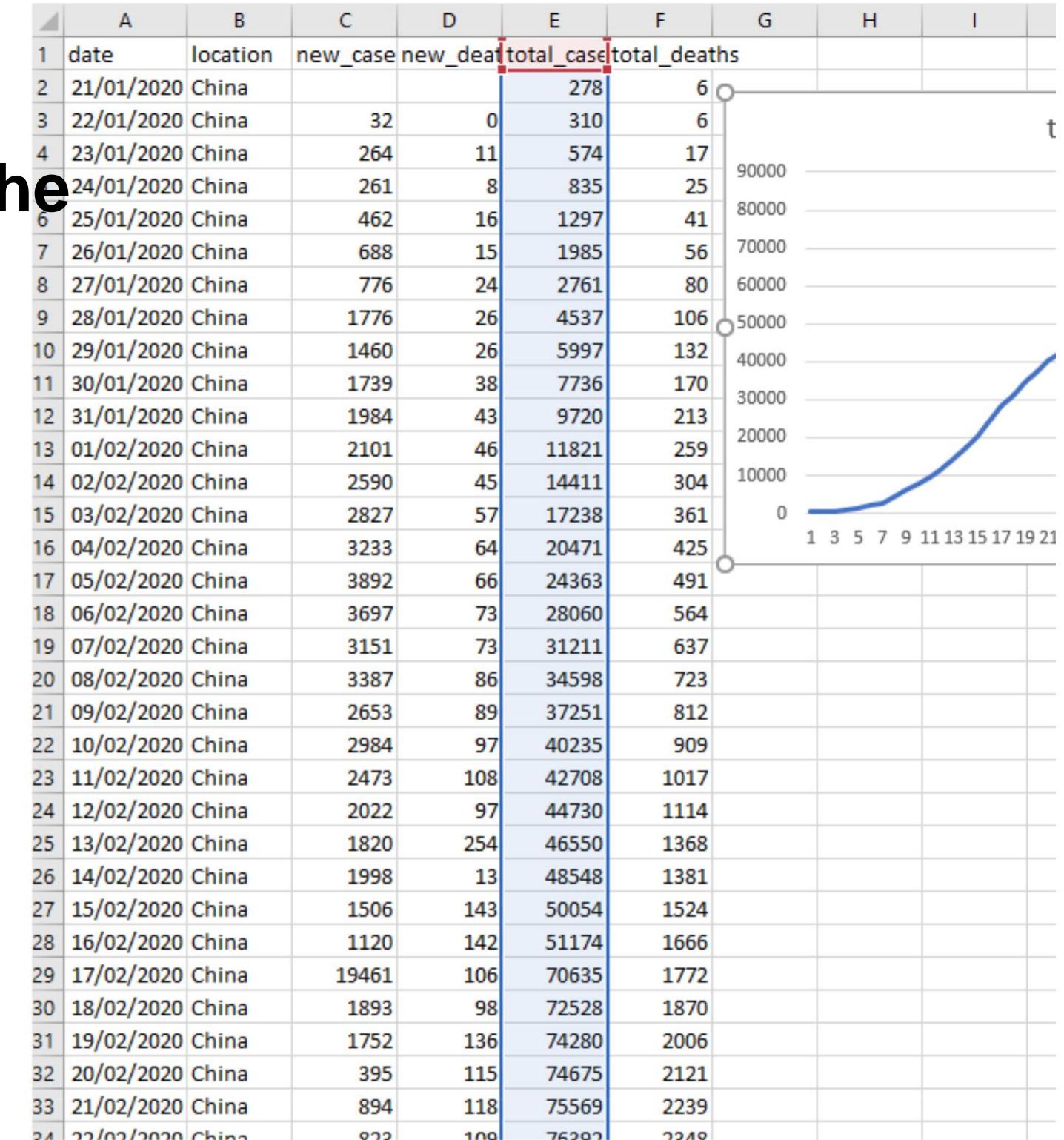
- We are going to train the data algorithm via train.csv,
- here the variable will be the user inputs, where loan id is a string, gender a boolean, married a boolean etc. based on the user input the algorithm is going to build three different models and check whether the person is eligible to get the 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 these models. The accuracy for the loan approval analysis not that perfect. It may be wrong sometimes and does't valueate 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.



SOLUTION/PROBLEM APPROACH

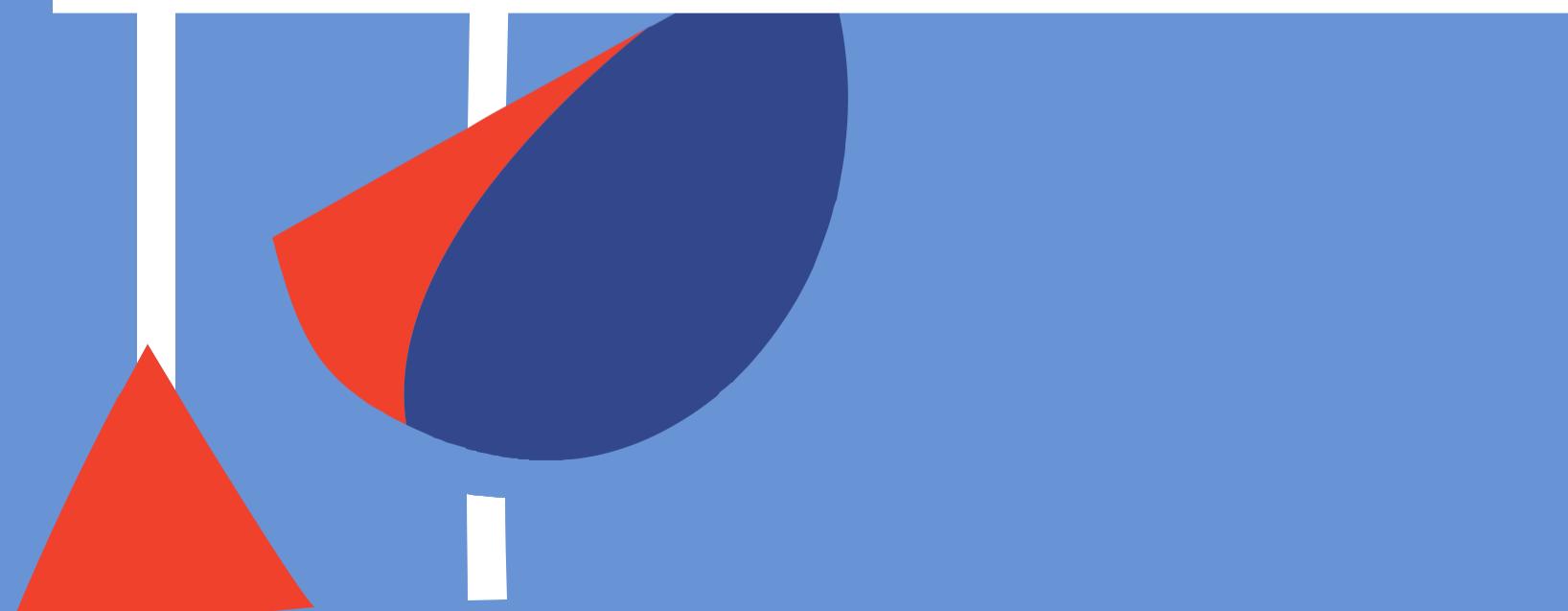
1). SOLUTION

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.



DATAVISUALISATION

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:

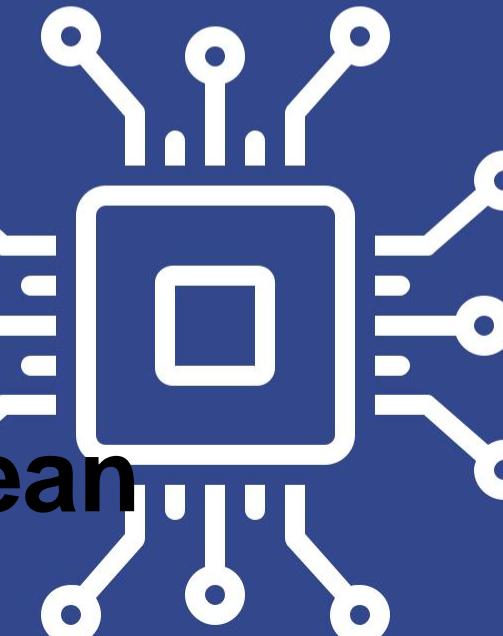
<https://github.com/Nandakishor12/S1-maths-project.git>

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



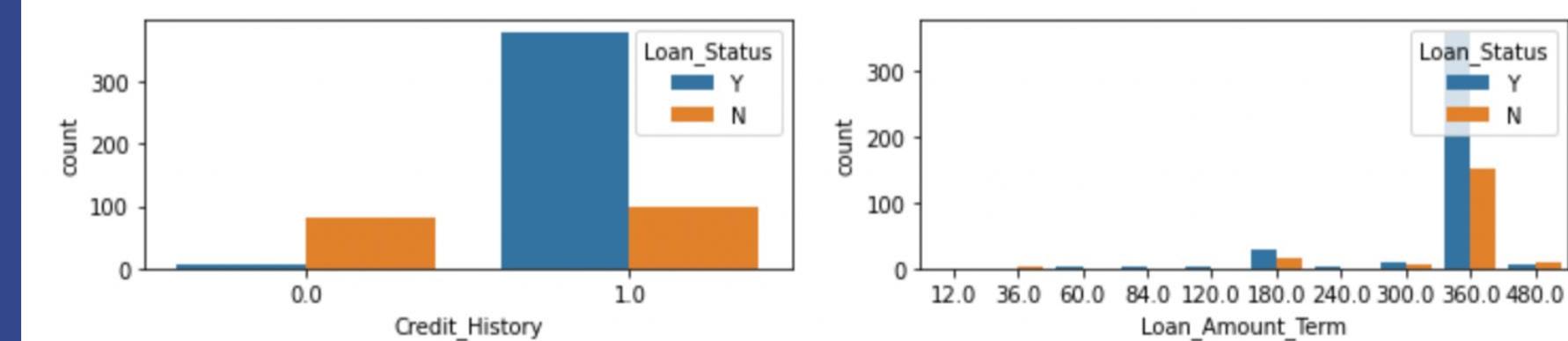
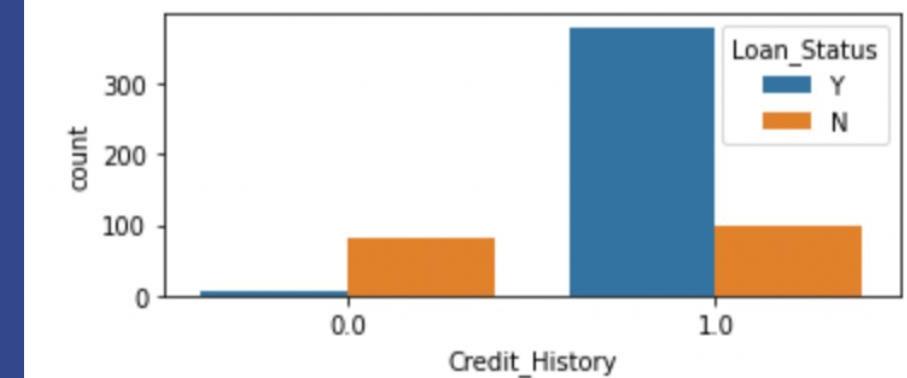
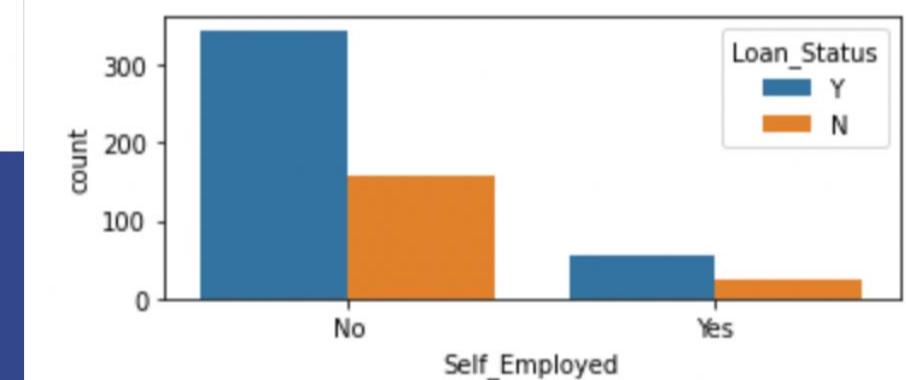
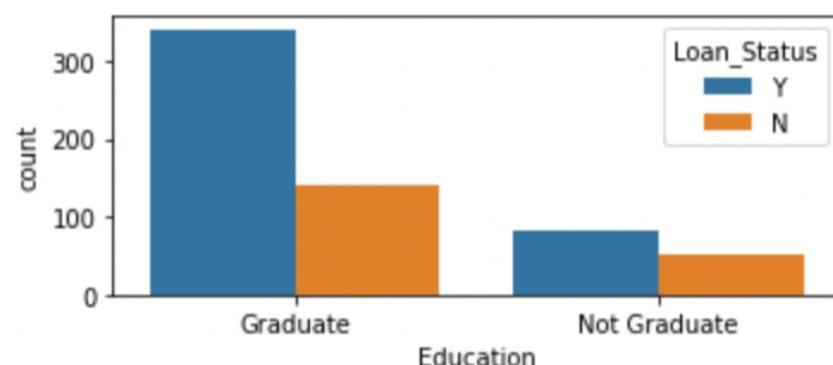
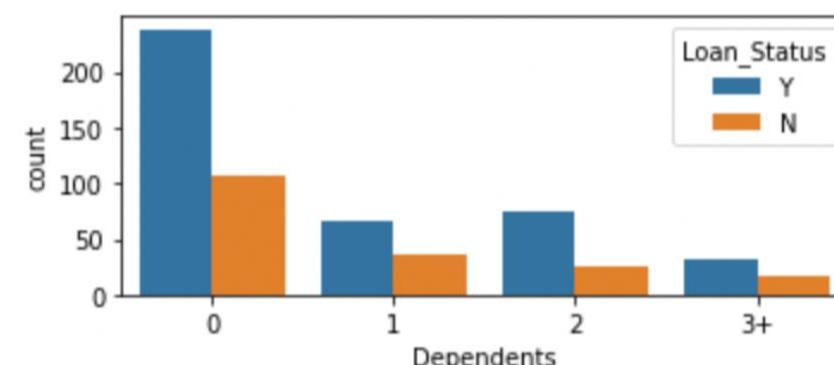
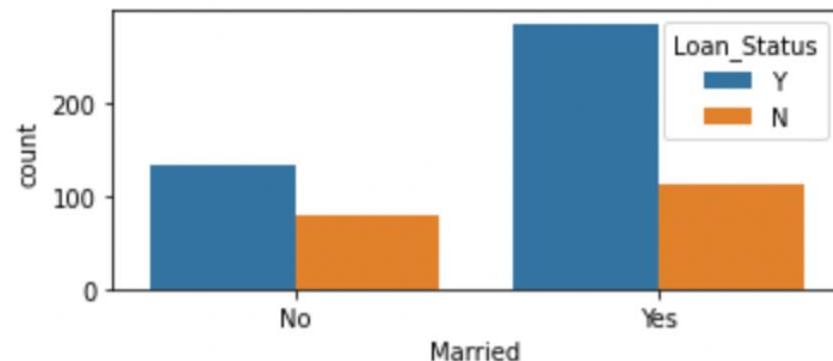
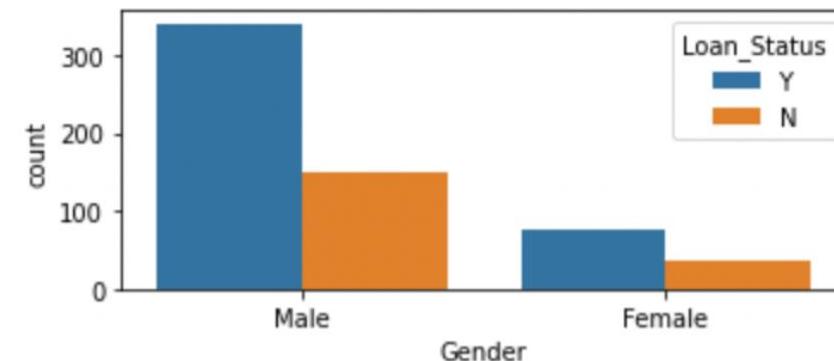
IMPLEMENTATION

- 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

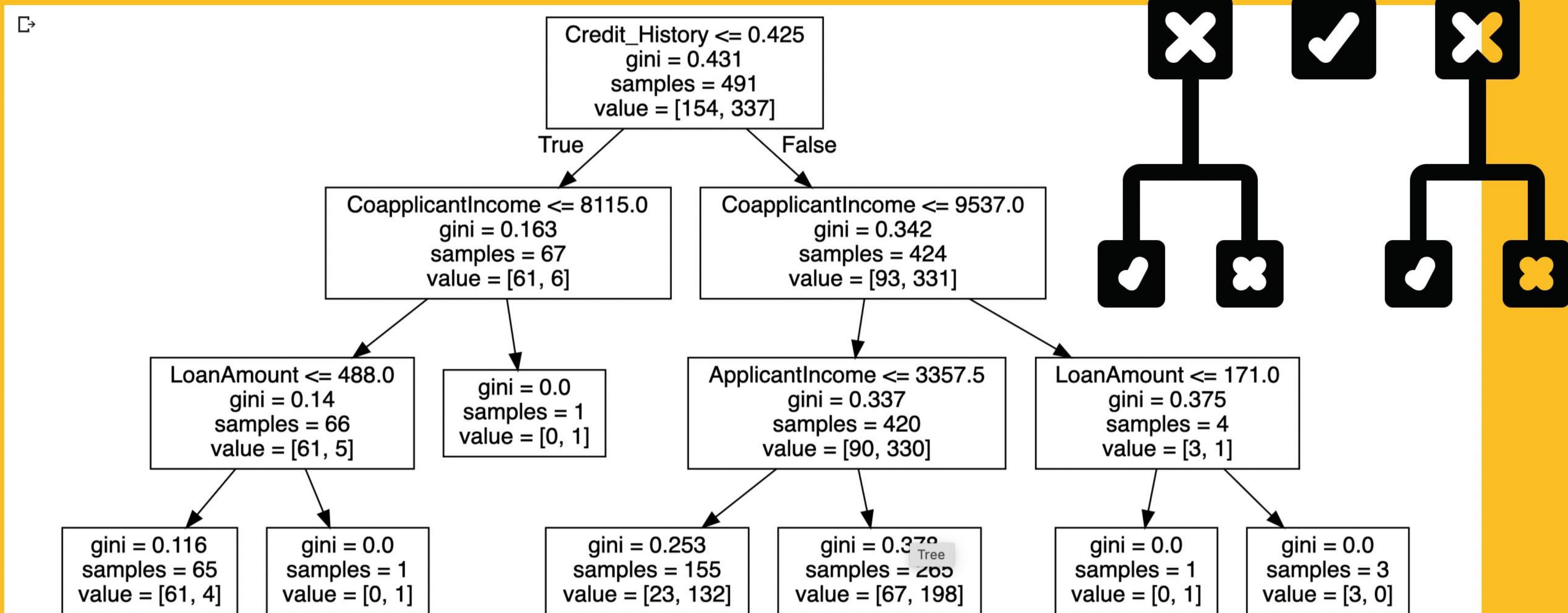


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Loan_ID          614 non-null    object  
 1   Gender           601 non-null    object  
 2   Married          611 non-null    object  
 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
```

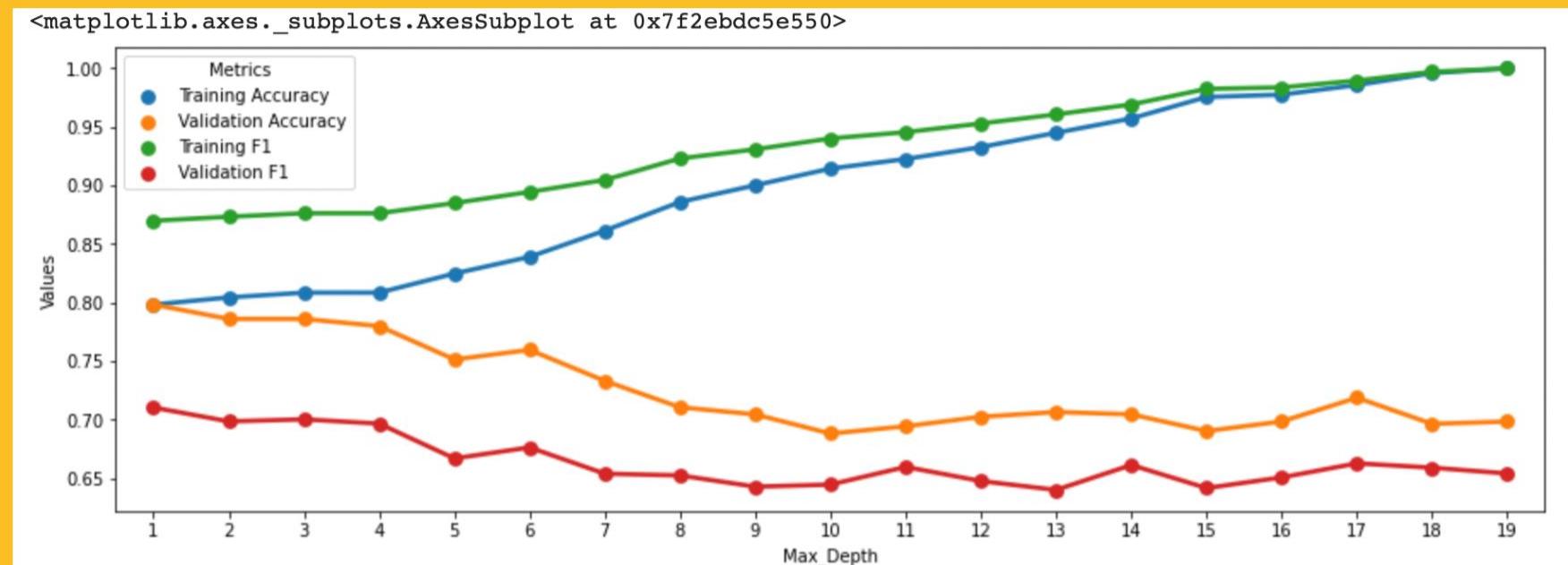
SEA BORN & MATPLOTLIB



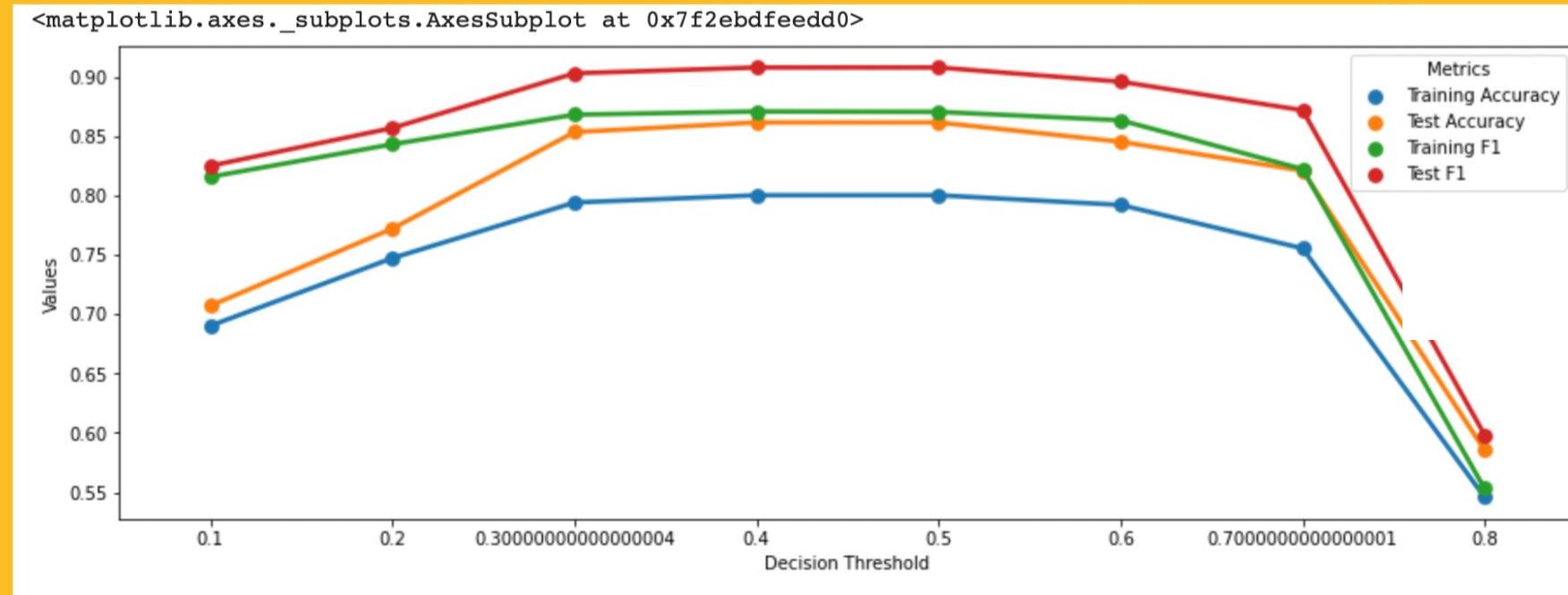
DECISION TREE



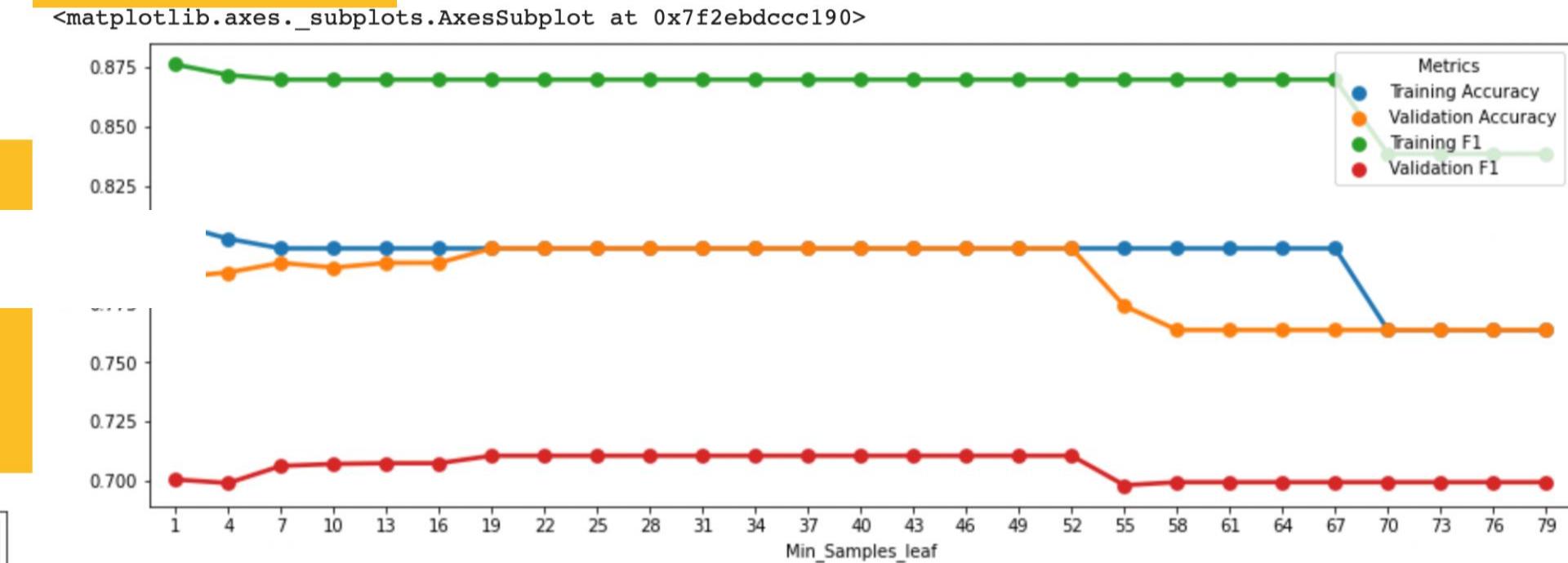
RETURN OF RESULTS IN GRAPHS



decision tree

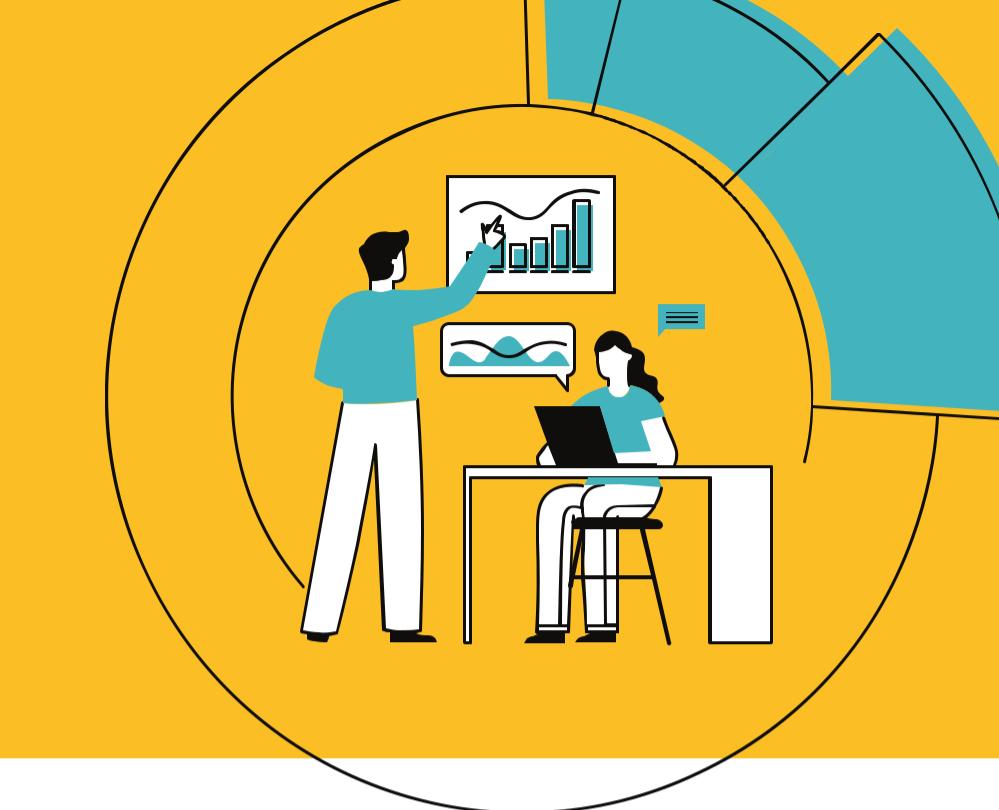


logistic regression



OBSERVATIONS

Here when we observe that are different outputs for different kind of models. 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 format of code in different model as mentions earlier. And gives the result with respect to that, (like how much is the applicant eligible). We prefer the logistic regression because it is better model compared to other two. It has the higher accuracy rate.

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

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

decision tree classifier

logistic regression

random forest

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

Predicted 0 1 All

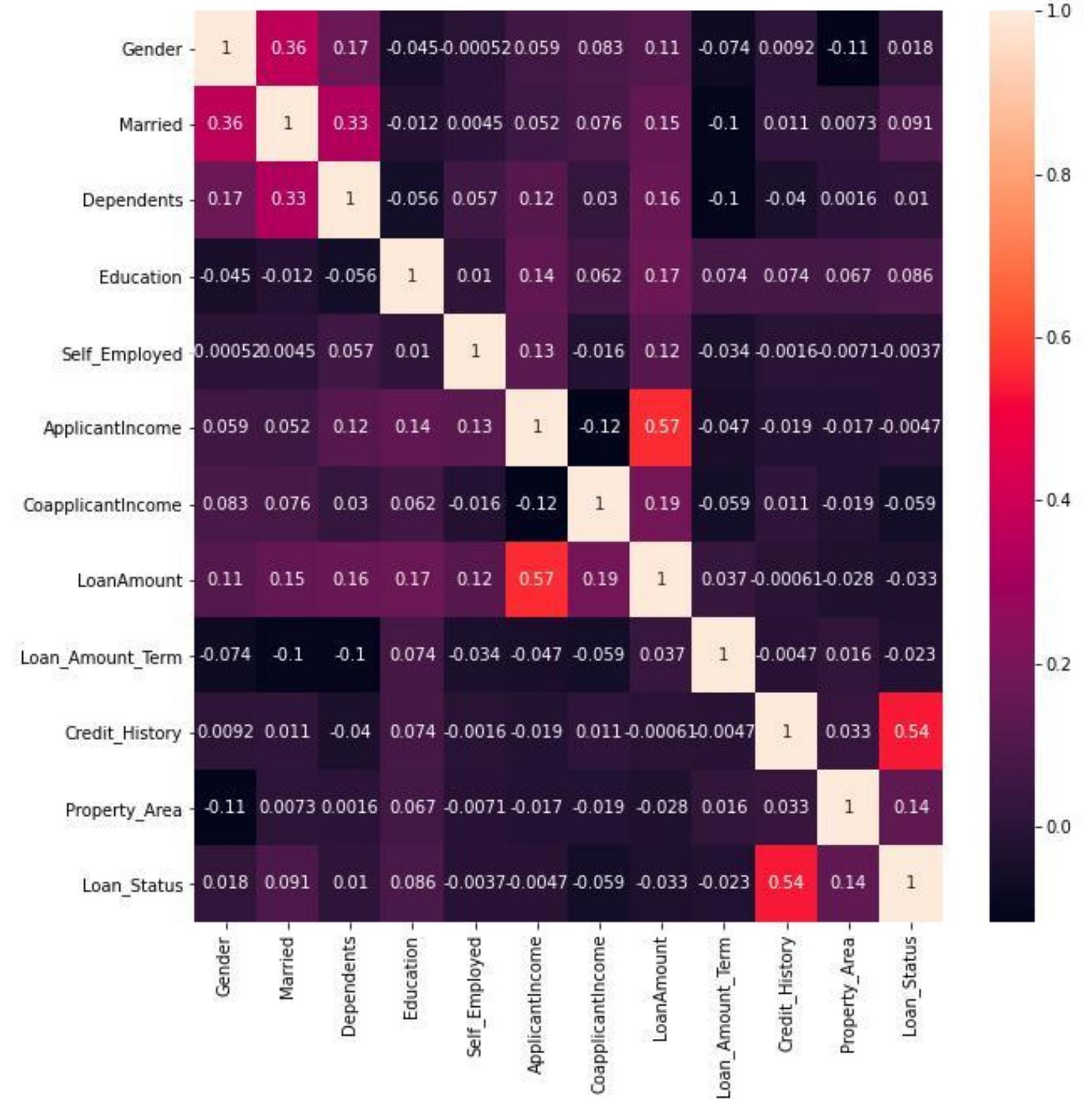
0	21	17	38
1	1	84	85
All	22	101	123

REFERENCES

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

<https://youtu.be/T9kgWBmUIRk>

<https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>



THANK

YOU