LOAN PREDICTION USING MACHINE LEARNING AND PYTHON

A PROJECT REPORT

Under the guidance of
Sofikul Mullick

BY
SHUBHAM KUMAR
SHASANKA GOLDAR
TEJASWI BAHADUR

In association with
Ardent Computech Pvt. Ltd.

• 1.	• Title of the project	• **LOAN PREDICTION USING MACHINE LEARNING AND PYTHON**
• 2.	• • Project members	• **Shubham Kumar** • **Shasanka Goldar** • **Tejaswi Bahadur**
• 3.	• Name of guide	○ **Sofikul Mullick**

Version	**Primary Author**	**Description of version**	**Date Completed**
Final	Shubham Kumar Shasanka Goldar Tejaswi Bahadur	Project Report	14/01/2022

Signature of Team Member
Date:

(For Office Use Only)

Signature of Approver
Date:

SOFIKUL MULLICK
Project Proposal Evaluator

DECLARATION

We hereby declare that the project work being presented in the project proposal entitled "**LOAN PREDICTION USING MACHINE LEARNING AND PYTHON**" in partial fulfilment of the requirements for the award of the degree of BACHELORS OF TECHNOLOGY at RAM KRISHNA MAHATO GOVERNMENT ENGINEERING COLLEGE, PURULIA, WEST BENGAL, is an authentic work carried out under the guidance of **SOFIKUL MULLICK**. The matter embodied in this project work has not been submitted elsewhere for theaward of any degree of our knowledge and belief.

Date: **25.06.2021**

Name of the Student:

Shubham Kumar

Shasanka Goldar

Tejaswi Bahadur

CERTIFICATE

This is to certify that this proposal of minor project entitled "**LOAN PREDICTION USING MACHINE LEARNING AND PYTHON**" is a record of bona fide work, carried out by **Shubham Kumar, Shashanka Goldar and Tejaswi Bahadur** under my guidance at Ardent Computech Pvt.Ltd.. In my opinion, the report in its present form is in partial fulfilment of the requirementsfor the award of the degree of **BACHELORS OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING** and as per regulations of the Ardent Computech Pvt.Ltd. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report.

Guide / Supervisor
SOFIKUL MULLICK
Ardent Computech Pvt. Ltd.

ACKNOWLEDGEMENT

Success of any project depends largely on the encouragement and guidelines of many others. We take this sincere opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project work.

We would like to show our greatest appreciation to **SOFIKUL MULLICK, Project Engineer at Ardent Computech Pvt. Ltd..** We always feel motivated and encouraged every time by his valuable advice and constant inspiration; without his encouragement and guidance this project would not have materialized.

Words are inadequate in offering our thanks to the other trainees, project assistants and other members at Ardent Computech Pvt. Ltd. for their encouragement and cooperation in carrying out this project work. The guidance and support received from all the members and who are contributing to this project, was vital for the success of this project.

INDEX

Sl No.	Topic	Page No.
1.	Abstract	7
2.	Introduction	8
3.	Problem definition	10
4.	Project Goal	11
5.	Methodology	12
6.	Project Objective	13
7.	Project Workflow	13
8.	Project Implementation	14
9.	Step-by-Step Working	15
10.	Project Limitations	20
11.	Future Scope	20
12.	Summary	21
13.	Bibliography	22
14.	CODE	23

ABSTRACT

Loan approval is a very important process for banking organizations. The system approved or reject the loan applications. Recovery of loans is a major contributing parameter in the financial statements of a bank. It is very difficult to predict the possibility of payment of loan by the customer. In recent years many researchers worked on loan approval prediction systems. Machine Learning (ML)techniques are very useful in predicting outcomes for large amount of data. In this paper three machine learning algorithms, Logistic Regression (LR), Decision Tree (DT) and Random Forest (RF) are applied to predict the loan approval of customers. The experimental results conclude that the accuracy of Logistic Regression machine learning algorithm is better as compared to Decision Tree and Random Forest machine learning approaches.

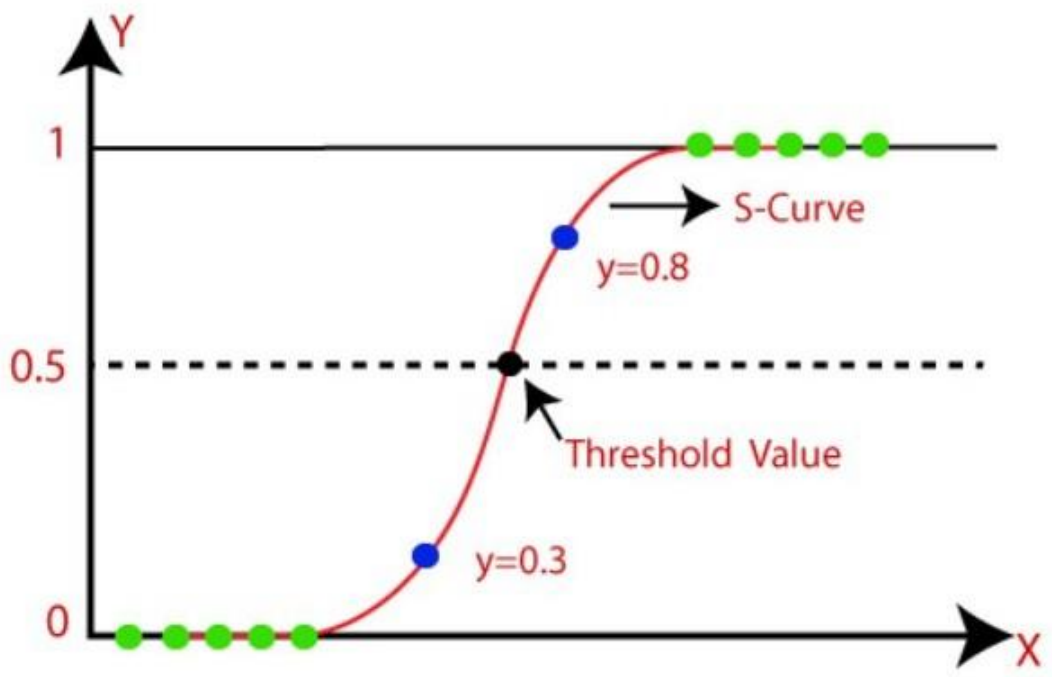
INTRODUCTION

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is 9 dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no). Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest
ML algorithms that can be used for various classification problems such as spam detection, diabetes prediction, cancer detection etc.



Type of Logistic Regression: On the basis of the categories, Logistic Regression can be classified into three types:

• **Binomial**: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

• **Multinomial**: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep" .

• **Ordinal**: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

PROBLEM DEFINITION

Banks, Housing Finance Companies and some NBFC deal in various types of loans like housing loan, personal loan, business loan etc in all over the part of countries. These companies have existence in Rural, Semi Urban and Urban areas. After applying loan by customer these companies validates the eligibility of customers to get the loan or not.
This project provides a solution to automate this process by employing machine learning algorithm. So the customer will fill an online loan application form. This form consist details like Gender, Marital Status, Qualification, Details of Dependents, Annual Income, Amount of Loan, Credit History of Applicant and others.

PROJECT GOAL

The aim of this Project is to provide quick, immediate and easy way to choose the deserving applicants. It can provide special advantages to the bank. The goal is achieved by using logistic regression in machine learning.

METHODOLOGY

• Data Selection: Data is the foundation for any machine learning project. The job is to find ways and sources of collecting relevant and comprehensive data, interpreting it, and analyzing results with the help of statistical techniques.

•Data Visualization: A large amount of information represented in graphic form is easier to understand and analyze. Some companies specify that a data analyst must know how to create slides, diagrams, charts, and templates.

•Data cleaning: This set of procedures allows for removing noise and fixing inconsistencies in data. A data scientist can fill in missing data using imputation techniques. A specialist also detects outliers — observations that deviate significantly from the rest of distribution.

•Data Splitting: A dataset used for machine learning should be partitioned into three subsets — training, test, and validation sets.

•Model Selection: After a data scientist has preprocessed the collected data and split it into three subsets, they can proceed with a model training. This process entails "feeding" the algorithm with training data. An algorithm will process data and output a model that is able to find a target value in new data. The purpose of model training is to develop a model.

•Model Evaluation: The goal of this step is to develop the simplest model able to formulate a target value fast and well enough and check the accuracy.

PROJECT OBJECTIVE

The main objective of this is **Loan Prediction Using Machine Learning and Python.**

PROJECT WORKFLOW

This is the detailed work architecture where we are showing the process of Diabetes Prediction Using Python and Machine Learning using Logistic Regression.

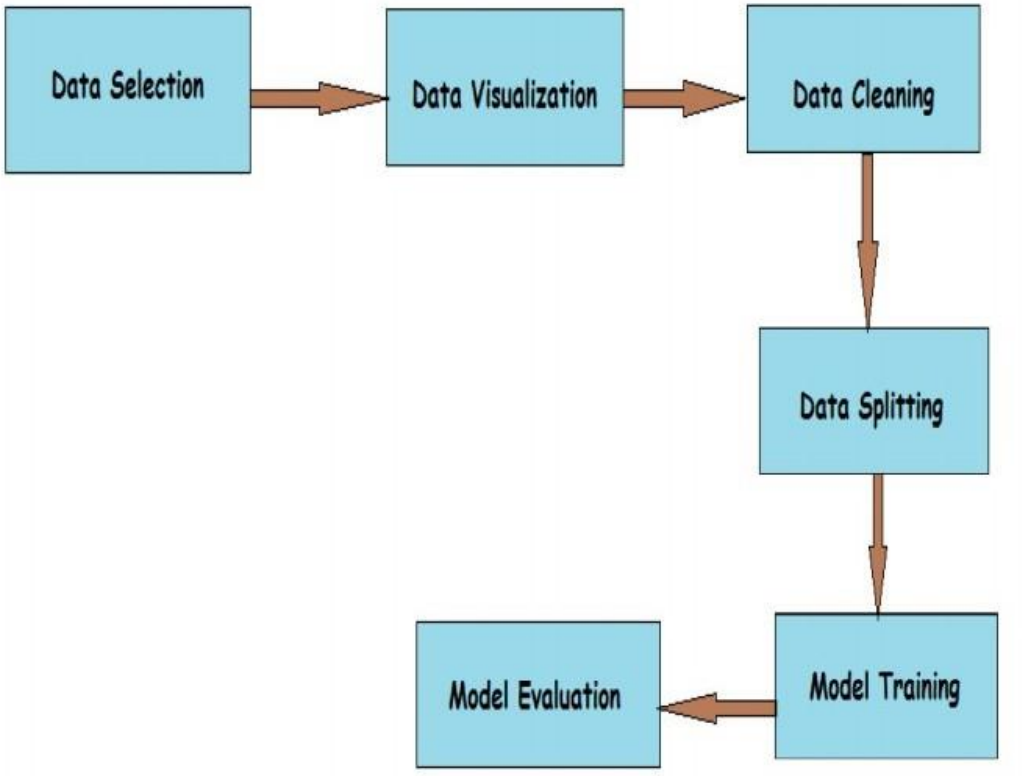


Figure: Workflow Architecture

PROJECT IMPLEMENTATION

• **SELECTION OF DATA:** The process of selecting data depends on the type of project we desire to. The data set can be collected from various sources such as a file, database, sensor and many other such sources.

• **VISUALIZATION OF DATA:** Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns.

• **DATA PRE-PROCESSING:** As we know that data pre-processing is a process of cleaning the raw data into clean data, so that can be used to train the model. So, we definitely need data pre-processing to achieve good results from the applied model in machine learning and deep learning projects.

• **SELECTION OF DEPENDENT AND INDEPENDENT DATA:** We need to select the dependent and independent data and store them in y and x.

• **SPLITTING OF THE DATA:** We train the classifier using 'training data set', then test the performance of your classifier on unseen 'test data set'. We split the data for training and testing by using the 'train_test_split'

• **FITTING THE MODEL:** In a data set, a training set is implemented to build up a model. Once the model is trained, we can use the same trained model to predict using the testing data i.e., the unseen data. Once this is done, we can develop a confusion matrix, this tells us how well our model is trained.

• **MODEL EVALUATION:** It is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future

STEP-BY-STEP WORKING

In [1]:
```
#Loan Prediction using Python and Machine Learning
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

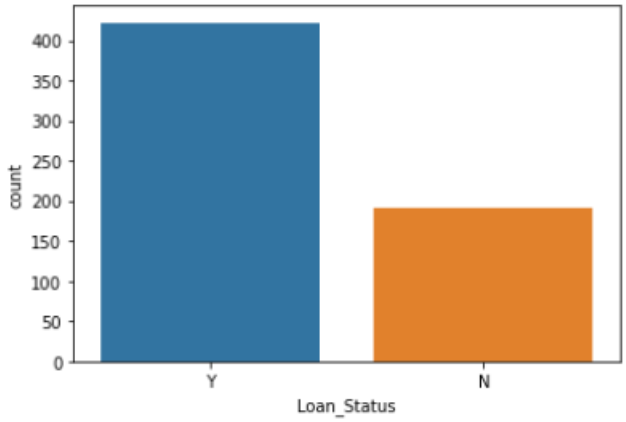
In [2]:
```
#Data Collection
Data=pd.read_csv("LoanPrediction(ML).csv")
Data.head(n=5)
```

Out[2]:

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

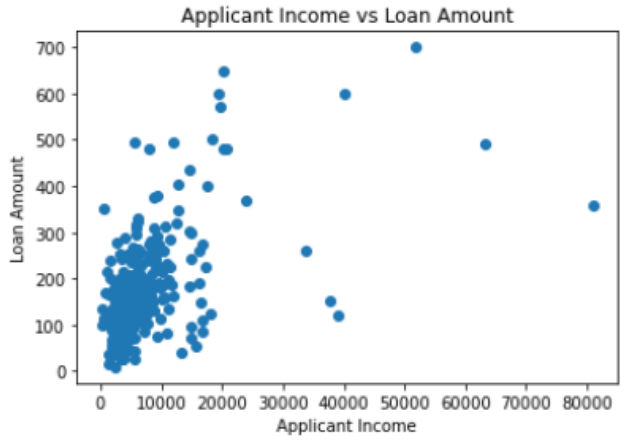
In [3]:
```
#Data Visualization
sns.countplot(x="Loan_Status",data=Data)
```

Out[3]:
```
<AxesSubplot:xlabel='Loan_Status', ylabel='count'>
```



In [4]:
```
##Plotting ApplicantIncome VS LoanAmount
plt.scatter(x=Data['ApplicantIncome'],y=Data['LoanAmount'])
plt.xlabel('Applicant Income')
plt.ylabel('Loan Amount')
plt.title('Applicant Income vs Loan Amount')
```

Out[4]:
```
Text(0.5, 1.0, 'Applicant Income vs Loan Amount')
```



In [5]:
```
#Changing the Text into Int values
Data['Dependents'].replace(['3+'],[3],inplace=True)
Data['Loan_Status'].replace(['Y','N'],[1,0],inplace=True)
Data['Married'].replace(['Yes','No'],[1,0],inplace=True)
Data['Self_Employed'].replace(['Yes','No'],[1,0],inplace=True)
Data['Property_Area'].replace(['Urban','Semiurban','Rural'],[3,2,1],inplace=True)
Data['Education'].replace(['Graduate','Not Graduate'],[1,0],inplace=True)
```

In [6]:
```
##Checking Null values with respect to sections
Data.isnull().sum()
```

Out[6]:
```
Loan_ID              0
Gender              13
Married              3
Dependents          15
Education            0
Self_Employed       32
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount          22
Loan_Amount_Term    14
Credit_History      50
Property_Area        0
Loan_Status          0
dtype: int64
```

In [7]:
```
#Handling the Null Values
Data['Gender'].value_counts()
```

Out[7]:
```
Male      489
Female    112
Name: Gender, dtype: int64
```

In [8]:
```
Data.Gender=Data.Gender.fillna('Male')
```

In [9]:
```
Data['Married'].value_counts()
```

Out[9]:
```
1.0    398
0.0    213
Name: Married, dtype: int64
```

In [10]:
```
Data.Married=Data.Married.fillna('1')
```

In [11]:
```
Data['Dependents'].value_counts()
```

Out[11]:
```
0    345
1    102
2    101
3     51
Name: Dependents, dtype: int64
```

In [12]:
```
Data.Dependents=Data.Dependents.fillna('0')
```

In [13]:
```
Data['Self_Employed'].value_counts()
```

Out[13]:
```
0.0    500
1.0     82
Name: Self_Employed, dtype: int64
```

In [14]:
```
Data.Self_Employed=Data.Self_Employed.fillna('0')
```

In [15]:
```
Data['LoanAmount'].value_counts()
```

Out[15]:
```
120.0    20
110.0    17
100.0    15
160.0    12
187.0    12
         ..
240.0     1
214.0     1
59.0      1
166.0     1
253.0     1
Name: LoanAmount, Length: 203, dtype: int64
```

In [16]:
```
Data.LoanAmount=Data.LoanAmount.fillna(Data.LoanAmount.mean())
```

In [17]:
```
Data['Loan_Amount_Term'].value_counts()
```

Out[17]:
```
360.0    512
180.0     44
480.0     15
300.0     13
240.0      4
84.0       4
120.0      3
60.0       2
36.0       2
12.0       1
Name: Loan_Amount_Term, dtype: int64
```

In [18]:
```
Data.Loan_Amount_Term=Data.Loan_Amount_Term.fillna('360')
```

In [19]:
```
Data['Credit_History'].value_counts()
```

Out[19]:
```
1.0    475
0.0     89
Name: Credit_History, dtype: int64
```

In [20]:
```
Data.Credit_History=Data.Credit_History.fillna('1')
```

In [21]:
```
##Again Checking if Null Vaues
Data.isnull().sum().sum()
```

Out[21]: 0

In [22]:
```
#Dividing Data into Dependent(y) and Independent Values(x)
x=Data[['Married','Dependents','Education','Self_Employed','ApplicantIncome','CoapplicantIncome','LoanAmount','Loan_Amount_Term','Credit_History','Property_Area']]
y=Data['Loan_Status']
```

In [23]:
```
#Dividing Independent and Dependent values into Training and Testing Data
from sklearn.model_selection import train_test_split
```

In [24]:
```
X_Train,X_Test,Y_Train,Y_Test=train_test_split(x,y,test_size=0.25,random_state=0)
```

In [25]:
```
#Creating Machine Learning Model
from sklearn.linear_model import LogisticRegression
```

In [26]:
```
My_mdl=LogisticRegression()
```

In [27]:
```
My_mdl.fit(X_Train,Y_Train)
```

Out[27]:
```
LogisticRegression()
```

In [28]:
```
Y_Pred=My_mdl.predict(X_Test)
```

In [29]:
```
Y_Pred
```

Out[29]:
```
array([1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 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, 0, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0],
      dtype=int64)
```

In [30]:
```
#Checking Accuracy
from sklearn.metrics import accuracy_score
```

In [31]:
```
print('So the model accuracy is ::::',accuracy_score(Y_Test, Y_Pred)*100 )
```

```
So the model accuracy is :::: 83.76623376623377
```

PROJECT LIMITATIONS

- We worked on the backend part of the system thus there is no frontend work associated which can result in more realistic look and focus on user experience
- While using dataset greater than 1gb, the project won't work properly

FUTURE SCOPE

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. This paper work can be extended to higher level in future. Predictive model for loans that uses machine learning algorithms, where the results from each graph of the paper can be taken as individual criteria for the machine learning algorithms.

SUMMARY

Logistic Regression is a powerful Machine Learning tool, and we can use it successfully for predicting categorical outputs of biomedical data. Data wrangling and data mining can benefit from excellent performances offered by Python and its libraries so well supported by the community. Linear Algebra programming has intrinsic advantages in avoiding, where possible, 'while' and 'for' loops. It is implementable by NumPy, a package that vectorizes the matrixes. NumPy makes working on them more comfortable, and guarantees better control over the operations, especially for large arrays. Moreover, the Machine Learning scenario with Python is enriched by the presence of many powerful packages (for example, Scikit-learn,) which provide excellently optimized classifications and predictions on data. The loan Prediction Data Set, with its 615 entries, offers an exhaustive assortment of parameters for classification and for this reason represents a perfect example for Machine Learning applications. Anyway, many of these features seem to be redundant, and a definite impact on classification and prediction by some of them remains still unknown.

BIBLOGRAPHY

- https://www.wikipedia.org
- https://www.slideshare.net
- https://www.sciencedirect.com
- https://www.irjet.net

CODE

```
Data['LoanAmount'].value_counts()
Data.LoanAmount=Data.LoanAmount.fillna(Data.LoanAmount.mean())
Data['Loan_Amount_Term'].value_counts()
Data.Loan_Amount_Term=Data.Loan_Amount_Term.fillna('360')
Data['Credit_History'].value_counts()
Data.Credit_History=Data.Credit_History.fillna('1')

##Again Checking if Null Vaues
Data.isnull().sum().sum()

#Dividing Data into Dependent(y) and Independent Values(x)
x=Data[['Married','Dependents','Education','Self_Employed','ApplicantIncome','Coapplica
ntIncome','LoanAmount','Loan_Amount_Term','Credit_History','Property_Area']]
y=Data['Loan_Status']

#Dividing Independent and Dependent values into Training and Testing Data
from sklearn.model_selection import train_test_split
X_Train,X_Test,Y_Train,Y_Test=train_test_split(x,y,test_size=0.25,random_state=0)

#Creating Machine Learning Model
from sklearn.linear_model import LogisticRegression
My_mdl=LogisticRegression()
My_mdl.fit(X_Train,Y_Train)
Y_Pred=My_mdl.predict(X_Test)
Y_Pred

#Checking Accuracy
from sklearn.metrics import accuracy_score
print('So the model accuracy is ::::',accuracy_score(Y_Test, Y_Pred)*100
```