LOAN STATUS PREDICTION

A COURSE PROJECT REPORT

Submitted by

Vignan (RA2011027010154)
Nishitha (RA2011027010169)

Under the guidance of

Dr. M.PRAKASH

In partial fulfilment for the Course

of

Machine Learning – 18CSE392T

in

DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

KATTANKULATHUR - 603 203

November, 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project titled "**LOAN STATUS PREDICTION"** is the bonafide work of **Guduguntla Nishitha(RA2011027010169), Vignan(RA2011027010154)** who carried out the project work under my supervision.

SUPERVISOR

Dr. M. Prakash
Associate Professor
Department of Data Science and Business Systems
SRM Institute of Science and Technology
Kattankulathur – 603 203

HEAD OF THE DEPARTMENT

Dr. M. LAKSHMI
Professor & Head
Department of Data Science and Business Systems
SRM Institute of Science and Technology
Kattankulathur – 603 203

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professor** and **Course Coordinator** for their constant encouragement and support.

We are highly thankful to my Course project Faculty Dr. **M.PRAKASH, Associate Professor, Department of Data Science and Business Systems,** for his assistance, timely suggestion and guidance throughout the duration of this course project.

We extend our gratitude to our **HoD, Dr. M. Lakshmi, Professor, Department of Data Science and Business Systems,** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

Loan Status Prediction

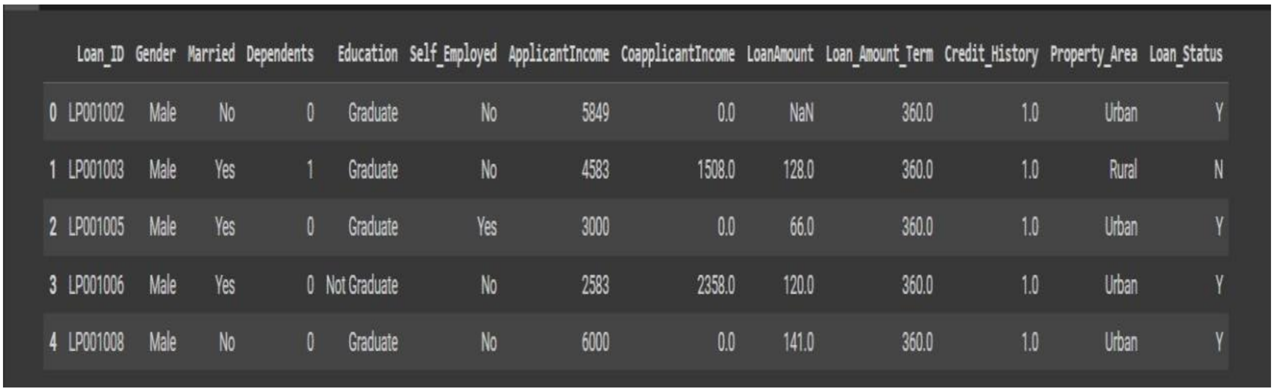
2. Abstract

Dream Housing Finance company deals in all home loans. They have presence across all urban, semi urban and rural areas. Customer first apply for home loan after that company validates the customer eligibility for loan.
The company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they have given a problem to identify the customers segments, those are eligible for loan amount so that they can specifically target these customers. Here they have provided a partial data set.

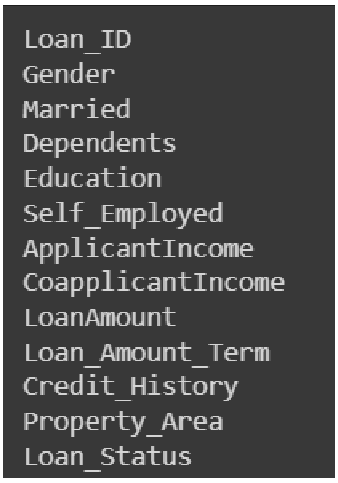
3. Dataset Description

Among all industries, insurance domain has the largest use of analytics & data science methods. This data set would provide you enough taste of working on data sets from insurance companies, what challenges are faced, what strategies are used, which variables influence the outcome etc. This is a classification problem. The data has 615 rows and 13 columns.
Problem-----
Company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they

have given a problem to identify the customers segments, those are eligible for loan amount so that they can specifically target these customers. Here they have provided a partial data set.



	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



```
Loan_ID
Gender
Married
Dependents
Education
Self_Employed
ApplicantIncome
CoapplicantIncome
LoanAmount
Loan_Amount_Term
Credit_History
Property_Area
Loan_Status
```

4. Modules Description

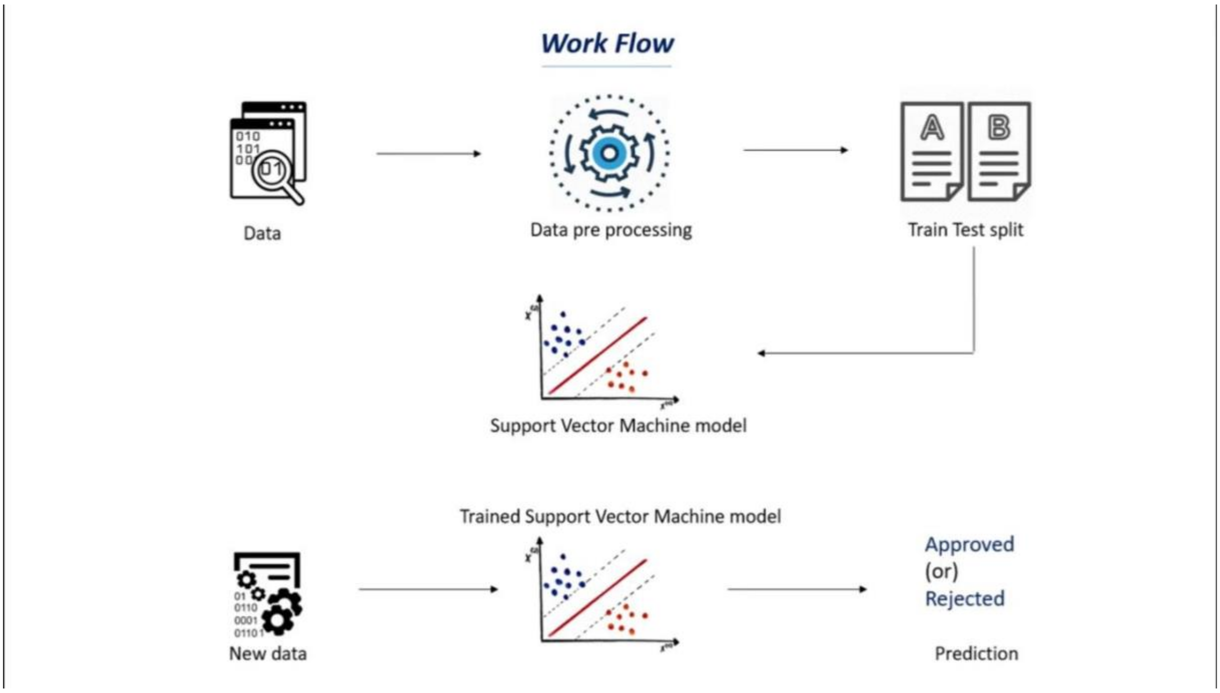
I) NumPy

NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

II) Pandas

Pandas is an open source library in Python. It provides ready to use high-performance data structures and data analysis tools. Pandas module runs on top of NumPy and it is popularly used for data science and data analytics.

4.1 Architecture Design

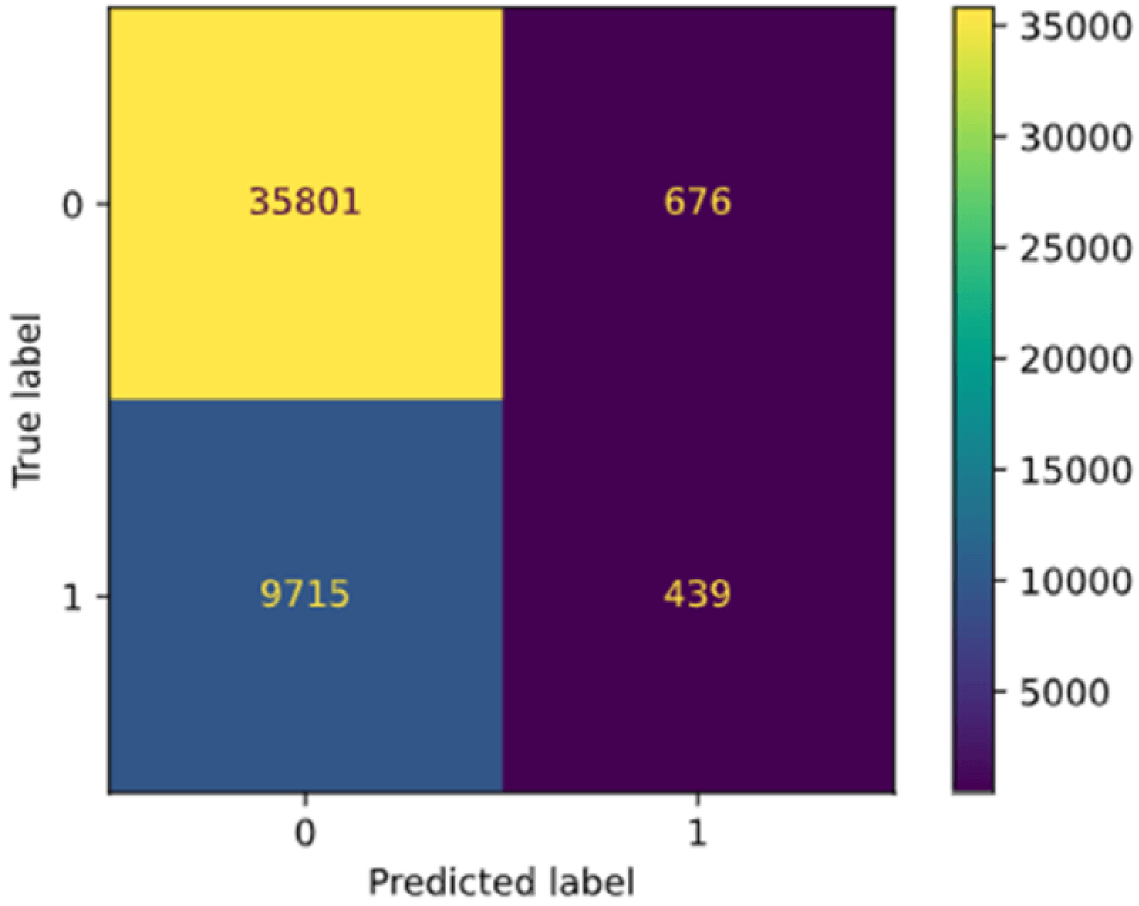


4.2 Algorithm used

Support Vector Machine (SVM)

5. Results and discussion

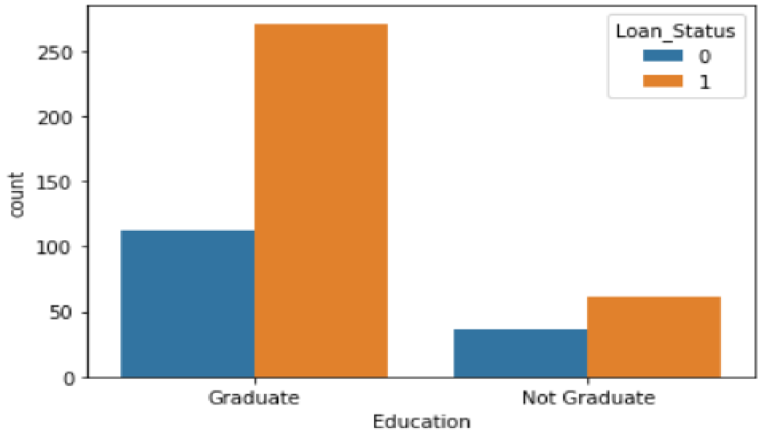
5.1 Confusion Matrix



5.2 Measures of data set

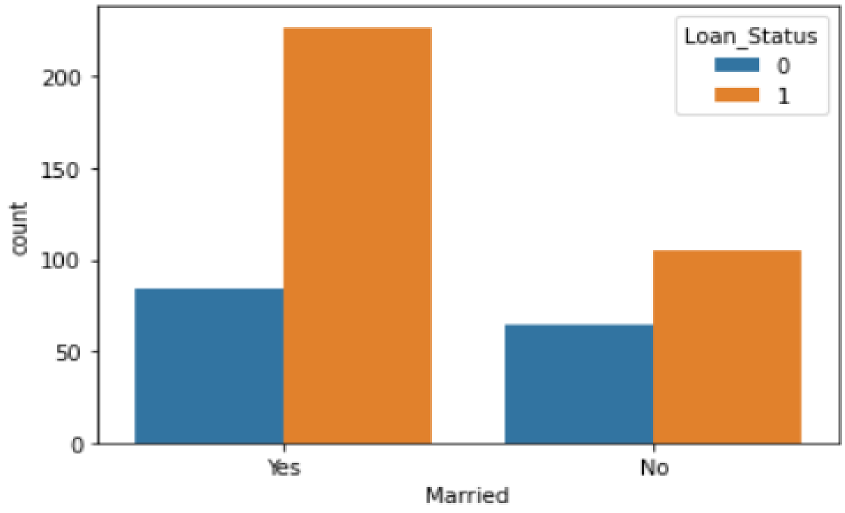
5.2.1 Education & Loan Status

sns.countplot(x='Education',hue='Loan_Status',data=loan_dataset)



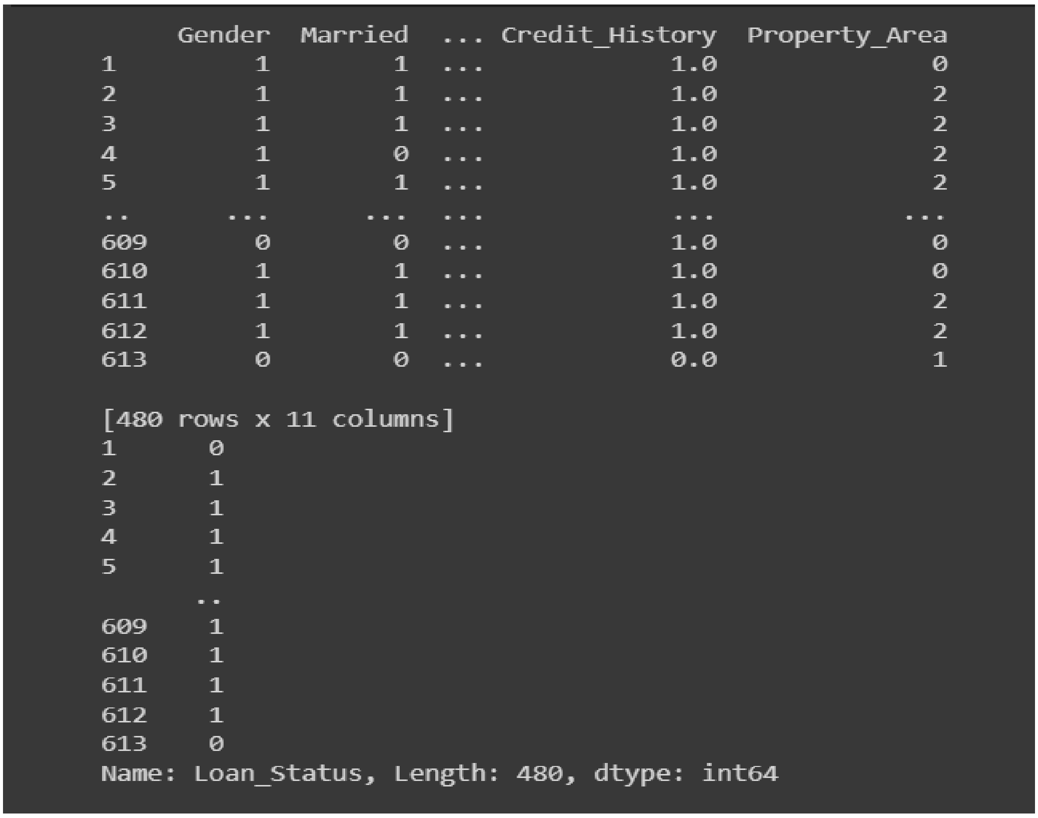
5.2.2 Marital status & Loan Status

sns.countplot(x='Married',hue='Loan_Status',data=loan_dataset)



5.3 Splitting the features and target

Train test split



```
     Gender  Married  ... Credit_History  Property_Area
1         1        1  ...            1.0              0
2         1        1  ...            1.0              2
3         1        1  ...            1.0              2
4         1        0  ...            1.0              2
5         1        1  ...            1.0              2
..      ...      ...  ...            ...            ...
609       0        0  ...            1.0              0
610       1        1  ...            1.0              0
611       1        1  ...            1.0              2
612       1        1  ...            1.0              2
613       0        0  ...            0.0              1

[480 rows x 11 columns]
1      0
2      1
3      1
4      1
5      1
      ..
609    1
610    1
611    1
612    1
613    0
Name: Loan_Status, Length: 480, dtype: int64
```

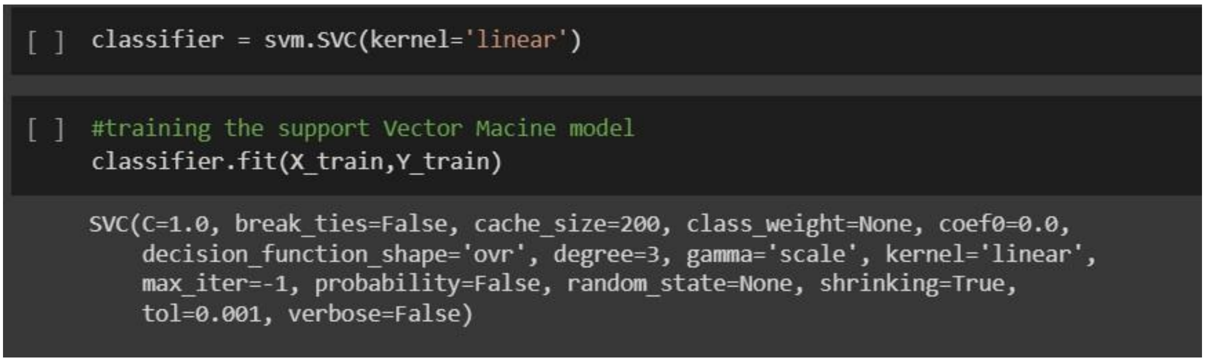
5.4 Model Analysis

Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane

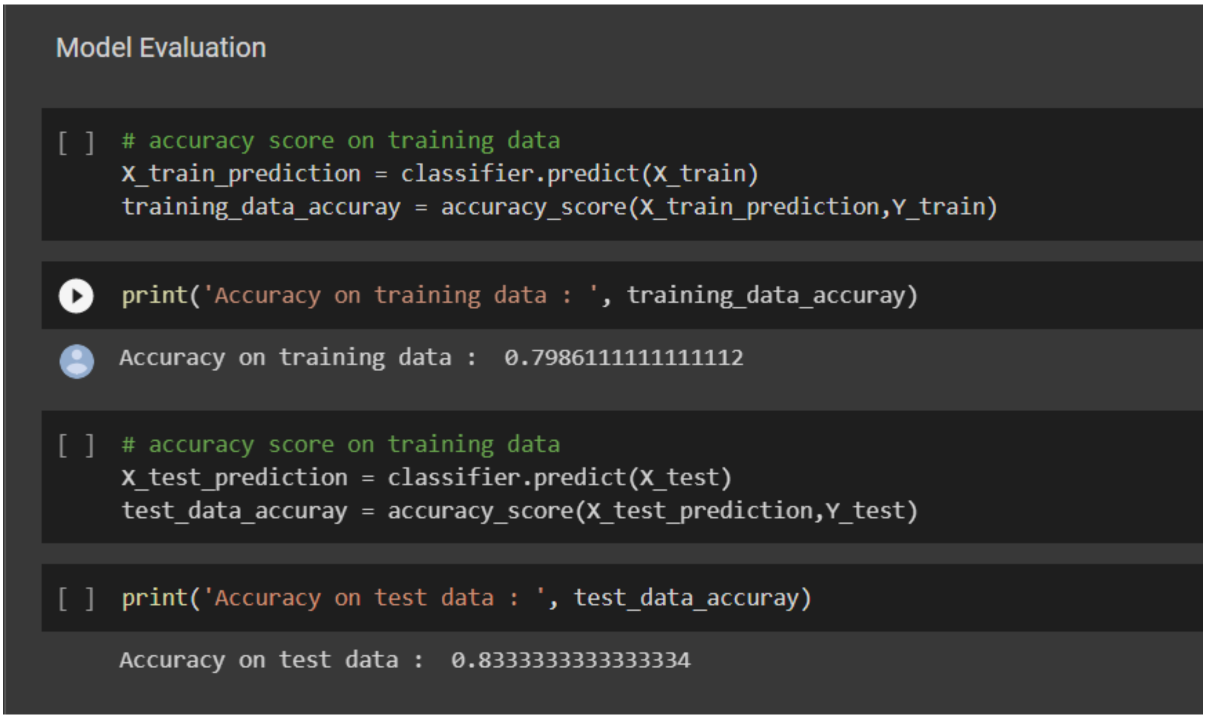


```
[ ] classifier = svm.SVC(kernel='linear')
```

```
[ ] #training the support Vector Macine model
    classifier.fit(X_train,Y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

5.5 Model Evaluation



Model Evaluation

```
[ ] # accuracy score on training data
    X_train_prediction = classifier.predict(X_train)
    training_data_accuray = accuracy_score(X_train_prediction,Y_train)
```

```
print('Accuracy on training data : ', training_data_accuray)
```

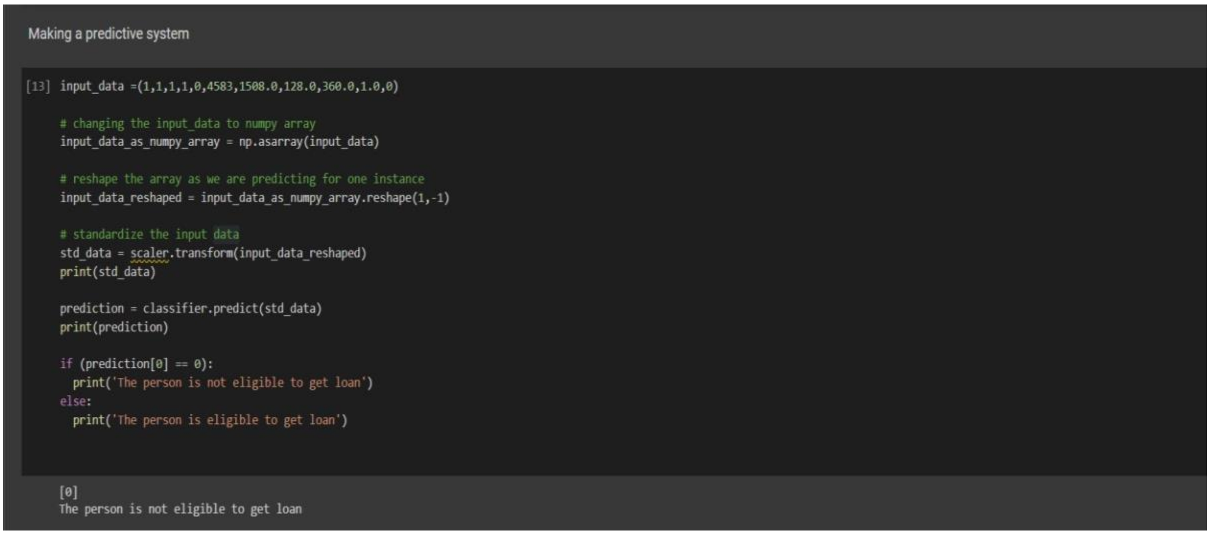
```
Accuracy on training data :  0.7986111111111112
```

```
[ ] # accuracy score on training data
    X_test_prediction = classifier.predict(X_test)
    test_data_accuray = accuracy_score(X_test_prediction,Y_test)
```

```
[ ] print('Accuracy on test data : ', test_data_accuray)
```

```
Accuracy on test data :  0.8333333333333334
```

5.6 Making a predictive system



Making a predictive system

```
[13] input_data =(1,1,1,1,0,4583,1508.0,128.0,360.0,1.0,0)

     # changing the input_data to numpy array
     input_data_as_numpy_array = np.asarray(input_data)

     # reshape the array as we are predicting for one instance
     input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

     # standardize the input data
     std_data = scaler.transform(input_data_reshaped)
     print(std_data)

     prediction = classifier.predict(std_data)
     print(prediction)

     if (prediction[0] == 0):
       print('The person is not eligible to get loan')
     else:
       print('The person is eligible to get loan')
```

```
     [0]
     The person is not eligible to get loan
```

6. Conclusion

This model is built to predict if a person is eligible to get loan by svm model.

7. Appendix

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score

# loading the dataset to pandas DataFrame
loan_dataset = pd.read_csv('/content/dataset.csv')
```

```
loan_dataset.head()

# number of rows and columns
loan_dataset.shape

# statistical measures
loan_dataset.describe()

# number of missing values in each column
loan_dataset.isnull().sum()

# dropping the missing values
loan_dataset = loan_dataset.dropna()

# number of missing values in each column
loan_dataset.isnull().sum()

# label encoding
loan_dataset.replace({"Loan_Status":{'N':0,'Y':1}},inplace=True)

# printing the first 5 rows of the dataframe
loan_dataset.head()

# Dependent column values
loan_dataset['Dependents'].value_counts()

# replacing the value of 3+ to 4
loan_dataset = loan_dataset.replace(to_replace='3+', value=4)

# dependent values
loan_dataset['Dependents'].value_counts()

# education & Loan Status
```

```
sns.countplot(x='Education',hue='Loan_Status',data=loan_dataset)

# marital status & Loan Status
sns.countplot(x='Married',hue='Loan_Status',data=loan_dataset)

# convert categorical columns to numerical values
loan_dataset.replace({'Married':{'No':0,'Yes':1},'Gender':{'Male':1,'Fe
male':0},'Self_Employed':{'No':0,'Yes':1},

'Property_Area':{'Rural':0,'Semiurban':1,'Urban':2},'Education':{'Grad
uate':1,'Not Graduate':0}},inplace=True)

loan_dataset.head()

# separating the data and label
X = loan_dataset.drop(columns=['Loan_ID','Loan_Status'],axis=1)
Y = loan_dataset['Loan_Status']
X_train, X_test,Y_train,Y_test =
train_test_split(X,Y,test_size=0.1,stratify=Y,random_state=2)

print(X.shape, X_train.shape, X_test.shape)

classifier = svm.SVC(kernel='linear')

#training the support Vector Macine model
classifier.fit(X_train,Y_train)

# accuracy score on training data
X_train_prediction = classifier.predict(X_train)
training_data_accuray = accuracy_score(X_train_prediction,Y_train)

print('Accuracy on training data : ', training_data_accuray)

# accuracy score on training data
X_test_prediction = classifier.predict(X_test)
```

```
test_data_accuray = accuracy_score(X_test_prediction,Y_test)

print('Accuracy on test data : ', test_data_accuray)

input_data =(1,1,1,1,0,4583,1508.0,128.0,360.0,1.0,0)

# changing the input_data to numpy array
input_data_as_numpy_array = np.asarray(input_data)

# reshape the array as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

# standardize the input data
std_data = scaler.transform(input_data_reshaped)
print(std_data)

prediction = classifier.predict(std_data)
print(prediction)

if (prediction[0] == 0):
  print('The person is not eligible to get loan')
else:
  print('The person is eligible to get loan')
```

******