PREDICTING PERSONAL LOAN APPROVAL USING MACHINE LEARNING

The project submitted to Smart Internz

By
ARAVINTHKUMAR.B (2022K1988)
ARAVINDHAN .S(2022K1987)
ARUNPANDI.M (2022K1989)
BALAKUMAR .B (2022K1990)
YATHIN.K(2022k2027)

Department of Computer Science

GOVERNMENT ARTS AND SCIENCE COLLEGE
MODAKURICHI
(Affiliated to Bharathiar University)
ERODE - 638104

INDEX

Predicting Personal Loan Approval Using Machine Learning

Overview:

INTRODUCTION:

Oneofthemostimportantfactorswhichaffectourcountry’seconomyandfinancialconditionisthecreditsystemgovernedbythebanks.Theprocessof bank credit risk evaluation is recognized at banks across the globe. “As weknow credit risk evaluation isvery crucial, there is a variety of techniques areused for risk level calculation. In addition, credit risk is one of the mainfunctionsofthebankingcommunity.

The prediction of creditdefaulters is one ofthe difficult tasks foranybank.But by forecasting the loan defaulters, the banks definitely may reduce theirloss by reducing their non-profit assets, so that recovery of approved loanscan take place without any loss and it can play as the contributing parameterof the bank statement. This makes the study of this loan approval predictionimportant.MachineLearningtechniquesareverycrucialandusefulinthe predictionofthesetypesofdata.

We will be using classification algorithms such as Decision tree, Randomforest, KNN, ANNand xgboost. We will train and test the data with thesealgorithms. From this best model is selected and saved in pkl format. WewillbedoingflaskintegrationandIBMdeployment.

PURPOSE:

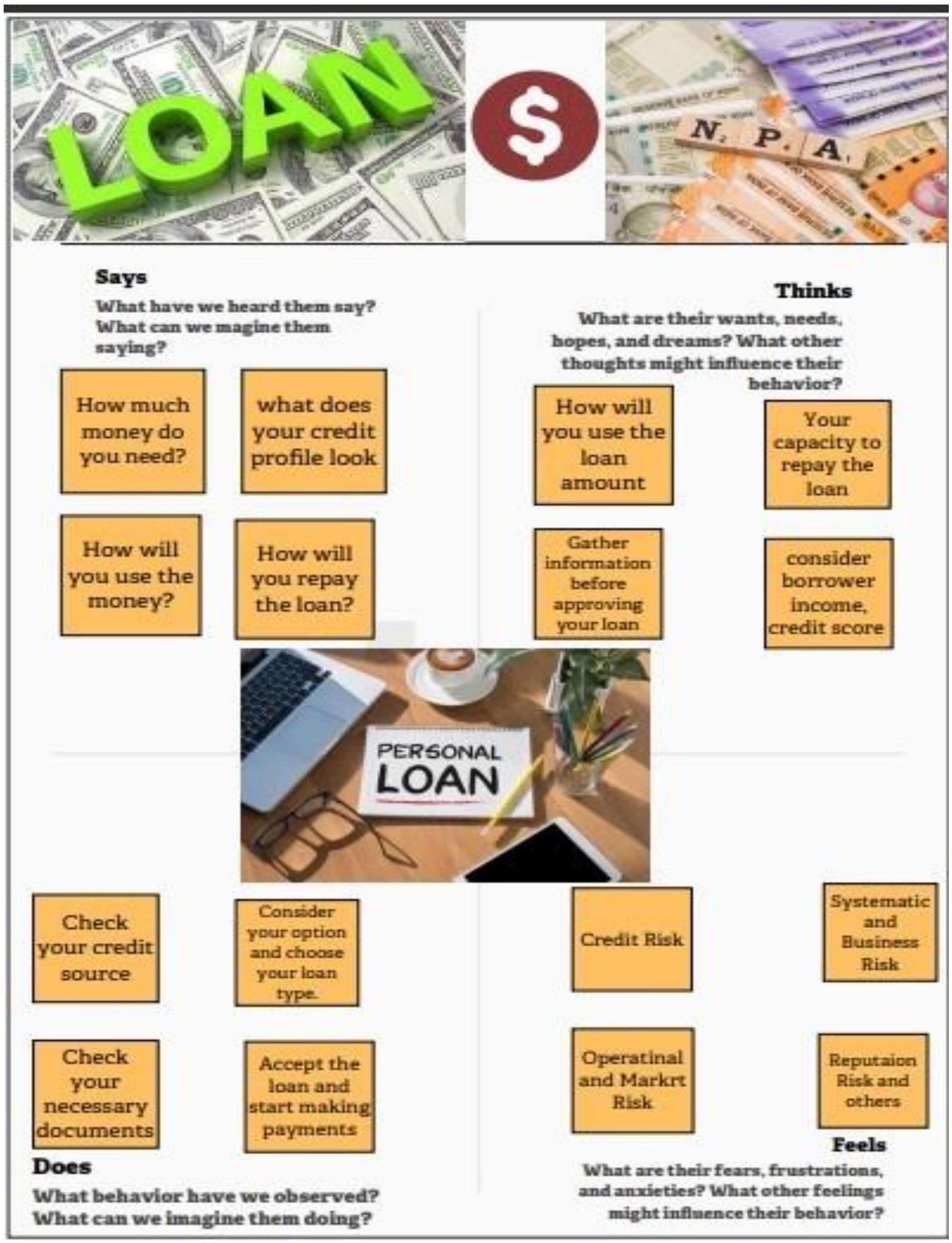
A loan is a sum of money that is borrowed and repaid over a period of time,typically with interest. Thereare various types of loans available to individuals andbusinesses,suchaspersonalloans,mortgages,autoloans,studentloans,business loans and many more. They are offered by banks, credit unions, andotherfinancialinstitutions,andthetermsoftheloan,suchasinterestrate,repayment period,andfees,varydepending onthe lenderandthetype ofloan.

A personal loan is a type of unsecured loan that can be used for a variety ofexpensessuchashomerepairs,medicalexpenses,debtconsolidation,andmore. The loan amount, interest rate, and repayment period vary depending onthe lender and the borrower's credit worthiness. To qualify for a personal loan,borrowerstypicallyneedtoprovideproofofincomeandhaveagoodcreditscore.

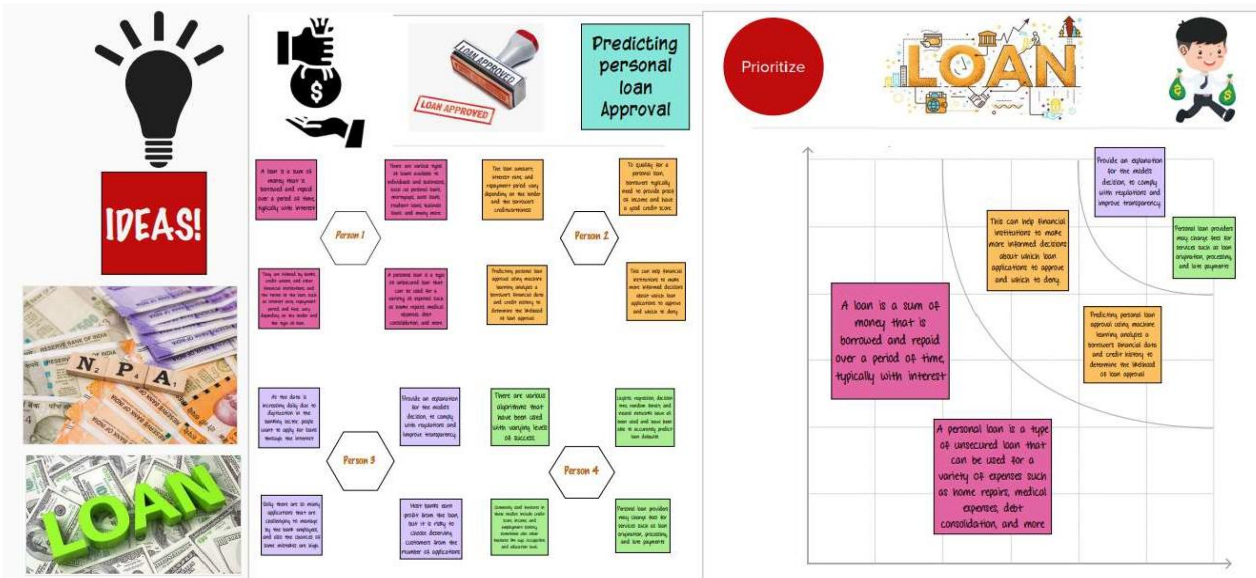
Predicting personal loan approval using machine learning analyses a borrower'sfinancial data and credit history to determine the likelihood of loan approval. Thiscan help financial institutions to make more informed decisions about which loanapplications toapproveandwhichtodeny.

PROBLEM DEFINITION & DESING THINKING:

1) EmpathyMap:



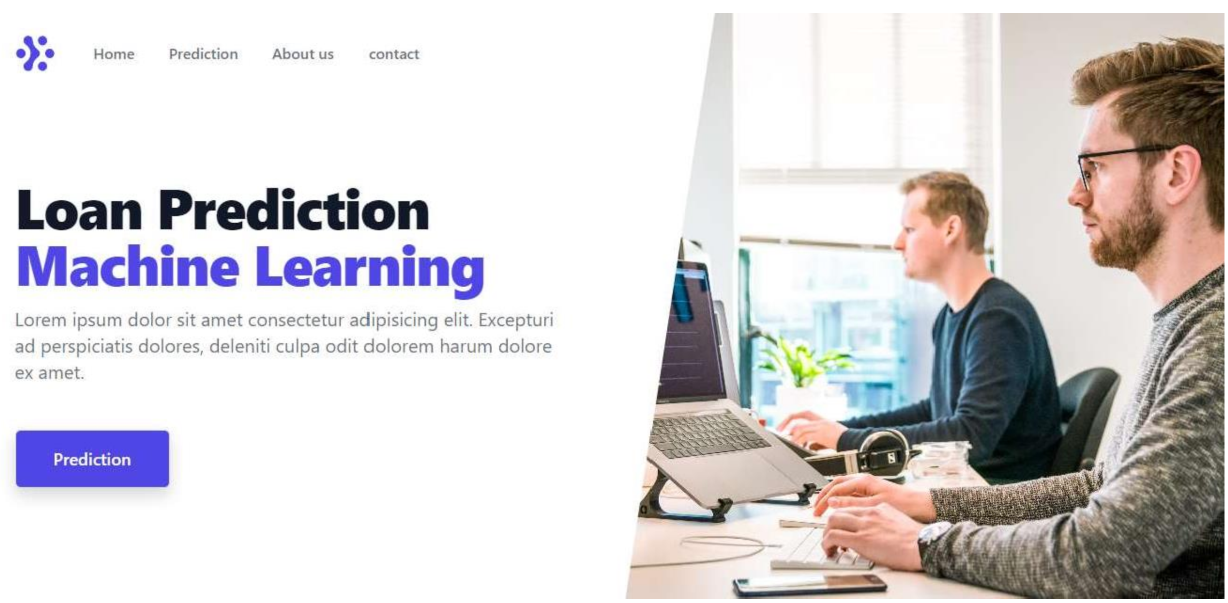
2) Ideation&BrainstormingMap



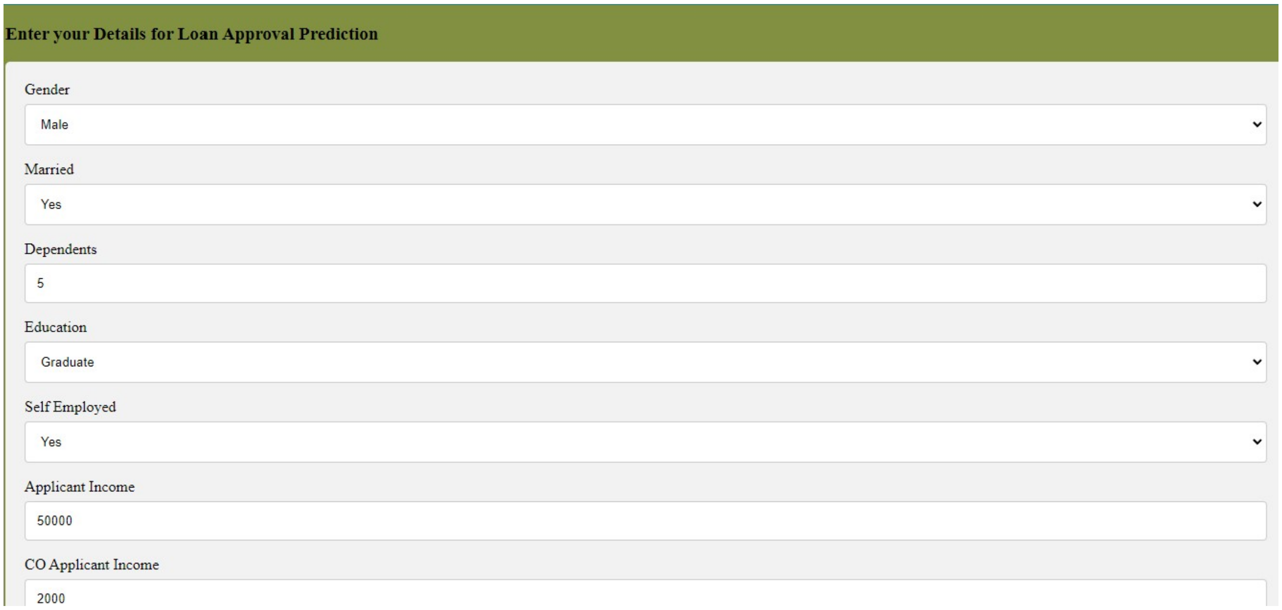
RESULT:

FinalFindingsoftheproject:Ho

mepageoftheproject:



FillingtheFormforLoanPrediction:



Outputpage:

ADVANTAGES&DISADVANTAGES:

ADVANTAGES:

Flexibility:Abankloanallowsonetorepayasperconvenienc ϵ aslongasthe instalmentsareregularandtimely.Unlikeanoverdraftwhereallthecreditis deductedingo,oraconsutiliւ mercreditcardwherethemaximumlimitcannotbe sedinonego.

CostEffectiveness:Whenitcomestointerestrates,bankloansareusuallythe cheapestoptioncompare d tooverdraftandcreditcard.

- ProfitRetention:Whenyouraisefundsthroughequityyouhav ϵ toshareprofits

withshareholders.However,inabankloanraisedfinanceyoudonothavetoshareprof
itswiththebank.

- BenefitofTax:Governmentmakestheinterestpayableontheloanatax-deductibleitemwhentheloanhasbeentakenforbusinesspurpose.

DISADVANTAGES:

HardPrerequisite:Sincebigfinancefromabankisbasedoncollateral,mostyoungbusinesseswillfindithardtofinancetheoperationsbasedonbankloan.

- IrregularPaymentAmounts:Overalongdurationpaybackviamonthlyinstalmentmight witnessvariationintherateofinterest.ThismeansthattheEMIwillnotbeconstant,ratherit willchangeaspertheinfluenceofthemarketontheinterestapplicable.

APPLICATIONS:

APredictionModelusesdatamining,statisticsandprobabilitytoforecastanoutcome.Everymodelhassomevariablesknownaspredictorsthatarelikelytoinfluencefutureresults.Thedatathatwascollectedfromvariousresourcesthenastatisticalmodelismade.Itcan useasimplelinearequationorasophisticatedneuralnetworkmappedusingacomplexsoftware.Asmoredatabecomesavailablethemodelbecomesmorerefinedandtheerrordecreasesmeaningthenit'llbeabletopredictwiththeleastriskandconsumingaslesstimeas itcan.ThePredictionModelhelpsthebanksbyminimizingtheriskassociatedwiththeloan approvalsystemandhelpstheapplicantbydecreasingthetimetakenintheprocess.The mainobjectiveoftheProjectistocomparetheLoanPredictionModelsmadeimplemented usingvariousalgorithmsandchoosethebestoneoutofthemthatcanshortentheloanapprovaltimeanddecreasetheriskassociatedwithit.Itisdonebypredictingiftheloancanbegiventothatpersononthebasisofvariousparameterslikecreditscore,income,age,marital status,gender,etc.

Thepredictionmodelnotonlyhelpstheapplicantbutalsohelpsthebankbyminimizingtheriskandreducingthenumberofdefaulters.

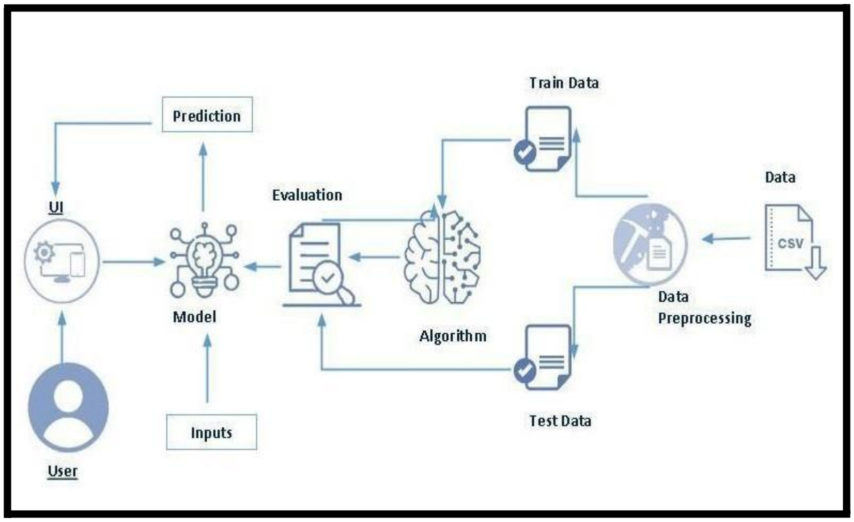
Inthepresentscenario,aloanneedstobeapprovedmanuallybyarepresentativeofthebankwhichmeansthatpersonwillberesponsibleforwhetherthepersoniseligiblefortheloanornotandalsocalculatingtheriskassociatedwithit.Asitisdonebyahumanitisatimeconsumingprocessandissusceptibletoerrors.Iftheloanisnotrepaid,thenitaccountsasalosstothebankandbanksearnmostoftheirprofitsbytheinterestpaidtothem.Ifthebankslosetoomuchmoney,thenitwillresultinabankingcrisis.Thesebankingcrisisaffectstheeconomyofthecountry.Soitisveryimportantthattheloanshouldbeapprovedwiththeleastamountoferrorinriskcalculationwhiletakingupastheleasttimepossible.Soaloanpredictionmodelisrequiredthatcanpredictquicklywhethertheloancanbepassedornotwiththeleastamountofriskpossible.

FUTURESCOPE:

Thetwomostpressingissuesinthebankingsectorare:Howriskyistheborrower.Shouldwelendtotheborrowergiventherisk?Theresponsetothefirstquestiondictatestheborrower'sinterestrate.Interestrate,amongotherthings(suchastimevalueofmoney),teststheriskinessoftheborrower,i.e.thehighertheinterestrate,theriskiertheborrower.Wewillthendecidewhethertheapplicantissuitablefortheloanbasedontheinterestrate.Lenders(investors)makeloanstocreditorsinreturnfortheguaranteeofinterest-bearingrepayment.Thatis,thelenderonlymakesareturn(interest)iftheborrowerrepaystheloan.However,whetherheorshedoesnotrepaytheloan,thelenderlosesmoney.Banksmakeloanstocustomersinexchangefortheguaranteeofrepayment.Somewoulddefaultontheirdebts,unabletorepaythemforanumberofreasons.Thebankretainsinsurancetominimizethepossibilityoffailureinthecaseofadefault.Theinsuredsumcancoverthewholeloanamountorjustaportionofit.Bankingprocessesusemanualprocedurestodeterminewhetherornotaborrowerissuitableforaloanbasedonresults.Manualprocedureswereamostlyeffective,buttheywereinsufficientwhentherewerealargenumberofloanapplications.Atthattime,makingadecisionwouldtakealongtime.Asaresult,theloanpredictionmachinelearningmodelcanbeusedtoassessacustomer'sloanstatusandbuildstrategies.Thismodelextractsandintroducestheessentialfeaturesofaborrowerthatinfluencethecustomer'sloanstatus.Finally,itproducestheplanned performance(loanstatus).Thesereportsmakeabankmanager'sjobsimplerandquicker.

APPENDIX:

TechnicalArchitecture:



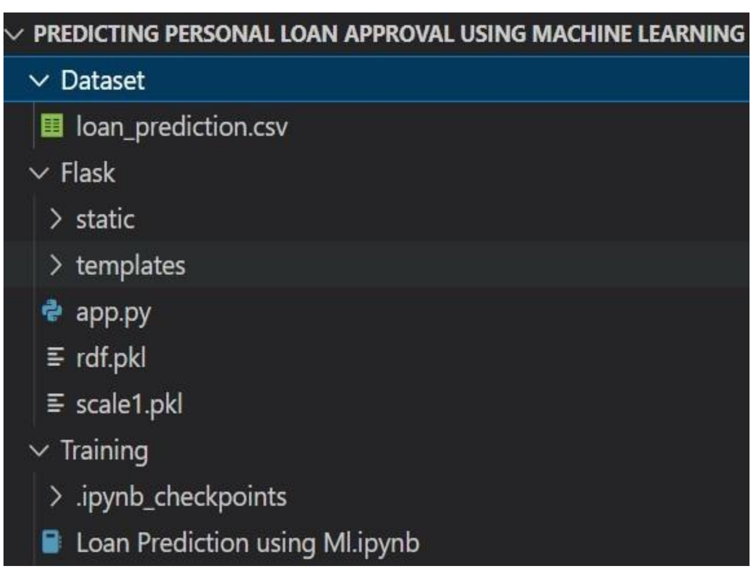
ProjectFlow:

- UserinteractswiththeUItoenter theinput.
- Enteredinputisanalysedbythemodelwhichisintegrated.
  - Oncemodelanalysestheinputthepredictionissho wcasedontheUI.
  - Toaccomplishthis,wehavetocompletealltheactivi tieslisted below,
- DefineProblem/ProblemUnderstanding
  - Specifythebusinessproblem
  - Businessrequirements
  - SocialorBusinessImpact.
- DataCollection&Preparation
  - Collectthedataset
  - DataPreparation

- ExploratoryDataAnalysis
  - Descriptivestatistical
  - VisualAnalysis
- ModelBuilding
  - Training themodelinmultiplealgorithms
  - Testingthemodel
- PerformanceTesting&HyperparameterTuning
  - Testingmodelwithmultipleevaluationmetrics
  - Comparingmodelaccuracybefore&afterapplyinghyperparametertuning
- ModelDeployment
  - Savethebestmodel
  - IntegratewithWebFramework
- ProjectDemonstration&Documentation
  - RecordexplanationVideoforprojectendtoendsolution
  - ProjectDocumentation-Stepbystepprojectdevelopmentprocedure

ProjectStructure:

Createthe Projectfolderwhichcontainsfilesasshownbelow



- WearebuildingaflaskapplicationwhichneedsHTMLpagesstoredinthe templatesfolderandapythonscriptapp.pyforscripting.

- rdf.pklisoursavedmodel.Furtherwewillusethismodelforflaskintegration.
- Trainingfolder contains amodel trainingfile.

DefineProblem/ProblemUnderstandingSpecifyth

ebusinessproblem

Businessrequirements

Thebusinessrequirementsforamachinelearningmodeltopredictperso nalloanapprovalincludetheabilitytoaccurately predictloanapprovalbasedonapplicantinformation,Minimisethenumberoffalse positives(approvedloansthatdefault)andfalsenegatives(rejectedloansthatwou ldhavebeensuccessful).Provideanexplanationforthemodel'sdecision,tocompl ywithregulationsandimprovetransparency.

SocialorBusinessImpact.

SocialImpact:-
Personalloanscanstimulateeconomicgrowthbyprovidingindividualswiththefu ndstheyneedtomakemajorpurchases, startbusinesses,orinvest intheireducation.

BusinessModel/Impact:-
Personalloanprovidersmaychargefeesforservicessuchasloanorigination,proce ssing,andlatepayments.Advertisingthebrandawareness andmarketingtoreachouttopotentialborrowerstogeneraterevenue.

DataCollection&Preparation:

ML depends heavily on data. It is the most crucial aspect that makesalgorithm training possible. So this section allows you to download therequireddataset.

Collectthedataset:

Therearemanypopularopensourcesforcollectingthedata.Eg:kaggle.com,UCIrepository, etc.

Inthisprojectwehaveused.csvdata.Thisdataisdownloadedfromkaggle.com.

Pleaserefertothelinkgivenbelowtodownloadthedataset.

Link: https://www.kaggle.com/datasets/altruistdelhite04/loan-prediction-problem-dataset

Asthedatasetisdownloaded.Letusreadandunderstand thedataproperlywiththehelpofsomevisualisationtechniquesandsomeanalysingtechniques.

Note:Thereareanumberoftechniquesforunderstandingthedata.Butherewehaveusedsomeofit.Inanadditionalway,youcanusemultipletechniques.

Importingthelibraries:

Importthenecessarylibrariesasshownintheimage.(optional)Herewehave usedvisualisationstyleasfivethirtyeight.

```
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sklearn
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import RandomizedSearchCV
import imblearn
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, f1_score
```

ReadtheDataset:

Ourdatasetformatmightbein .csv,excelfiles,.txt,.json,etc.Wecanreadthedatasetwiththehelpofpandas.

Inpandaswehaveafunctioncalledread_csv()toreadthedataset.Asaparameterwehave togivethedirectoryofthecsvfile.

```python
#importing the dataset which is in csv file
data = pd.read_csv('loan_prediction.csv')
data
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	L
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	3
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	3
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	3
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	3
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	3
...	...	...	...	...	...	...	...	...	...	..
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	3
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	1
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	3
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	3
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	3

614 rows × 13 columns

DataPreparation:

Aswehaveunderstoodhowthedatais,let'spre-processthe collecteddata.

Thedownloaddatasetisnotsuitablefortrainingthemachinelearningmodelasitmigh thavesomuchrandomnesssoweneedtocleanthedatasetproperlyinordertofetchg oodresults.Thisactivityincludesthefollowingsteps.

- Handlingmissingvalues
- Handlingcategoricaldata
- HandlingImbalanceData

Note:Thesearethegeneralstepsofpre-processingthedatabeforeusingitformachinelearning.Dependingonthecondition ofyourdataset,youmayormaynothavetogothroughallthesesteps.Handlingmis singvalues:

- Let'sfindtheshape ofourdatasetfirst.Tofind theshapeofourdata,thedf.shapemethodisused.Tofindthedatatype,df.info()f unctionisused.

```
data.info()

<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
```

- For checkingthe nullvalues,df.isnull()functionisused.Tosumthose nullvaluesweuse.sum()function.Fromthebelowimagewefoundthattherearenonu llvaluespresentinourdataset.Sowe canskiphandlingthemissingvaluesstep.

```
#finding the sum of null values in each column
data.isnull().sum()
```

```
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
```

- Fromtheabovecodeofanalysis,we caninferthatcolumns suchasgender,married,dependents,selfemployed,loanamount,loanamountterm andcredithistory arehavingthemissingvalues,weneedtotreattheminarequiredway.

```
data['Gender'] = data['Gender'].fillna(data['Gender'].mode()[0])

data['Married'] = data['Married'].fillna(data['Married'].mode()[0])

#replacing + with space for filling the nan values
data['Dependents']=data['Dependents'].str.replace('+','')

data['Dependents'] = data['Dependents'].fillna(data['Dependents'].mode()[0])

data['Self_Employed'] = data['Self_Employed'].fillna(data['Self_Employed'].mode()[0])

data['LoanAmount'] = data['LoanAmount'].fillna(data['LoanAmount'].mode()[0])

data['Loan_Amount_Term'] = data['Loan_Amount_Term'].fillna(data['Loan_Amount_Term'].mode()[0])

data['Credit_History'] = data['Credit_History'].fillna(data['Credit_History'].mode()[0])
```

- Wewillfillinthemissingvaluesin thenumericdatatypeusingthemeanvalueofthatparticularcolumnandcategoricaldatatypeusingthemostrepeatedvalue.

HandlingCategoricalValues:

Aswecanseeourdatasethascategoricaldatawemustconvertthecategoricaldatatointegerencodingorbinaryencoding.

Toconvertthecategoricalfeaturesintonumericalfeatures weuseencodingtechniques.Thereareseveraltechniquesbutinourprojectweareusingmanualencodingwiththehelpoflistcomprehension.

- Inourproject,Gender,married,dependents,self-employed,co-applicantsincome,loanamount,loanamount term,credithistoryWithlistcomprehensionencodingis done.

```
#changing the datype of each float column to int
data['Gender']=data['Gender'].astype('int64')
data['Married']=data['Married'].astype('int64')
data['Dependents']=data['Dependents'].astype('int64')
data['Self_Employed']=data['Self_Employed'].astype('int64')
data['CoapplicantIncome']=data['CoapplicantIncome'].astype('int64')
data['LoanAmount']=data['LoanAmount'].astype('int64')
data['Loan_Amount_Term']=data['Loan_Amount_Term'].astype('int64')
data['Credit_History']=data['Credit_History'].astype('int64')
```

HandlingImbalanceData:

Data Balancingisoneofthemostimportantstep,whichneedto beperformedforclassificationmodels,becausewhenwetrainourmodelonimbalanceddataset,wewillgetbiassedresults,whichmeansourmodelisabletopredict onlyoneclasselement

ForBalancingthedataweareusingtheSMOTEMethod.

SMOTE:Syntheticminorityoversamplingtechnique,whichwillcreatenewsynthetic datapointsforunderclassaspertherequirementsgivenbyususingKNNmethod.

```
#Balancing the dataset by using smote
from imblearn.combine import SMOTETomek
```

```
smote = SMOTETomek(0.90)
```

```
C:\Users\HP\AppData\Roaming\Python\Python39\site-packages\imblearn\utils\_validation.py:587: FutureWarning: Pass sampling_strategy=0.9
keyword args. From version 0.9 passing these as positional arguments will result in an error
  warnings.warn(
```

```
#dividing the dataset into dependent and independent y and x respectively
y = data['Loan_Status']
x = data.drop(columns=['Loan_Status'],axis=1)
```

```
#creating a new x and y varialbles for the balnced set
x_bal,y_bal = smote.fit_resample(x,y)
```

```
#printing the values of y before balancing the data and after
print(y.value_counts())
print(y_bal.value_counts())
```

```
1    422
0    192
Name: Loan_Status, dtype: int64
1    351
0    308
Name: Loan_Status, dtype: int64
```

Fromtheabovepicture,wecaninferthat,previouslyyourdatasethad492class1,an d192classitems,afterapplyingsmotetechniqueonthedatasetthesizehasbeen changedforminorityclass.

ExploratoryDataAnalysis:

Descriptivestatistical:

Descriptiveanalysisisto studythebasicfeaturesofdatawiththestatisticalprocess.Here pandashasaworthyfunctioncalleddescribe.Withthisdescribefunctionwecanun derstandtheunique,topandfrequentvaluesofcategoricalfeatures.Andwecanfin dmean,std,min,maxandpercentilevaluesofcontinuousfeatures.

```
data.describe()
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

Visualanalysis:

Visualanalysisistheprocessof usingvisualrepresentations, such

ascharts,plots,

andgraphs,toexploreandunderstanddata.Itisawaytoquicklyidentifypatterns,t

rends,andoutliersinthedata,whichcanhelptogaininsightsandmakeinformed

decisions.

Univariateanalysis:

Insimplewords,univariateanalysisisunderstandingthedatawithasinglefeature.He

rewehavedisplayedtwodifferentgraphssuchasdistplotandcountplot.

- TheSeabornpackageprovidesawonderfulfunctiondistplot.Withthehelpofdist plot,wecanfindthedistributionofthefeature.Tomakemultiplegraphsinasinglepl ot,weuse subplot.

```python
#plotting the using distplot
plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(data['ApplicantIncome'], color='r')
plt.subplot(122)
sns.distplot(data['Credit_History'])
plt.show()
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be remo
ved in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot`
(an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be remo
ved in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot`
(an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Inourdatasetwehavesomecategoricalfeatures.Withthecountplotfunction,wear egoingtocounttheuniquecategoryinthosefeatures.Wehavecreatedadummydat aframewithcategoricalfeatures.Withforloopandsubplotwehaveplottedthisbelo wgraph.

- Fromtheplotwecametoknow,Applicantsincomeisskewedtowardsleftside,wh ereascredithistoryiscategoricalwith1.0and0.0

Countplot:-

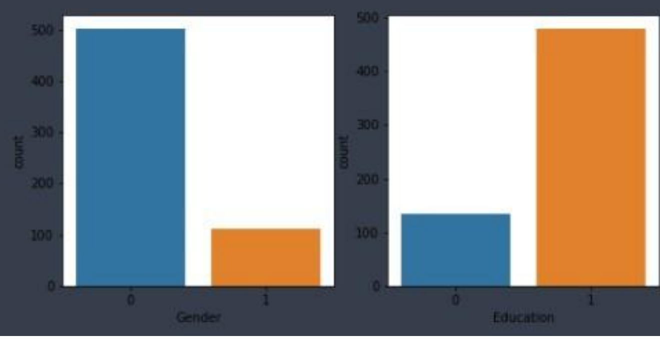
Acountplotcanbethoughtofasahistogramacrossacategorical,insteadofquanti tative,variable.ThebasicAPIandoptionsareidentical tothoseforbarplot(),soyoucancomparecountsacrossnestedvariables.

Fromthegraphwecaninferthat,genderandeducationisacategoricalvariableswi th2categories,fromgendercolumnwecaninferthat0- categoryishavingmoreweightagethancategory- 1,whileeducationwith0,itmeansnoeducationisaunderclasswhen compared withcategory-1,whichmeanseducated.

Bivariateanalysis:

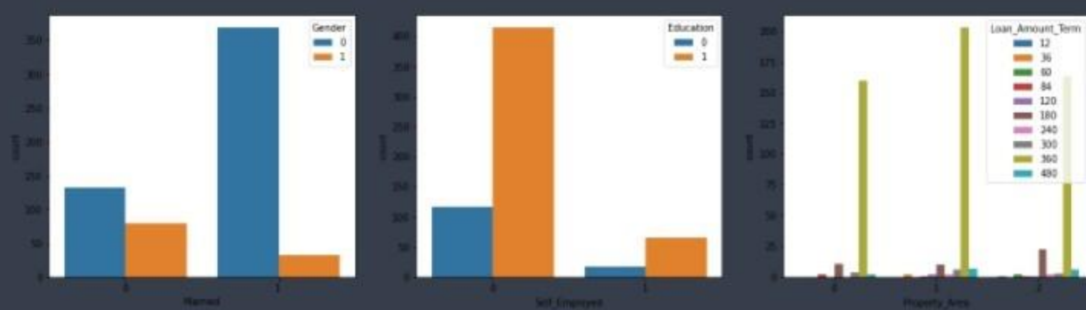
```
#plotting the count plot
plt.figure(figsize=(18,4))
plt.subplot(1,4,1)
sns.countplot(data['Gender'])
plt.subplot(1,4,2)
sns.countplot(data['Education'])
plt.show()
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From ve
rsion 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an e
rror or misinterpretation.
  warnings.warn(
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From ve
rsion 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an e
rror or misinterpretation.
  warnings.warn(
```



```
#visualising two columns against each other
plt.figure(figsize=(20,5))
plt.subplot(131)
sns.countplot(data['Married'], hue=data['Gender'])
plt.subplot(132)
sns.countplot(data['Self_Employed'], hue=data['Education'])
plt.subplot(133)
sns.countplot(data['Property_Area'], hue=data['Loan_Amount_Term'])
```

```
error or misinterpretation.
  warnings.warn(
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From v
ersion 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  warnings.warn(


<AxesSubplot:xlabel='Property_Area', ylabel='count'>
```



Fromtheabovegraphwecaninfertheanalysissuchas

- Segmenting thegendercolumnandmarriedcolumnbasedonbargraphs
- SegmentingtheEducationandSelf-employedbasedon bargraphs,fordrawinginsightssuchaseducatedpeopleareemployed.
- Loanamounttermbasedonthepropertyareaofapersonholding

Multivariateanalysis:

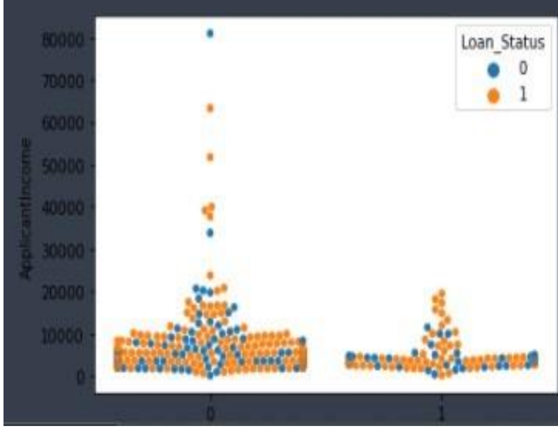
Insimplewords,multivariateanalysisistofind

therelationbetweenmultiplefeatures.Herewehaveusedaswarmplotfro

mtheseabornpackage.

```
#visulaized based gender and income what would be the appplication status
sns.swarmplot(data['Gender'],data['ApplicantIncome'], hue = data['Loan_Status'])
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\HP\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 67.1% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\Users\HP\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 33.0% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)


<AxesSubplot:xlabel='Gender', ylabel='ApplicantIncome'>
```



FromtheabovegraphweareplottingtherelationshipbetweentheGender,applicantsinco

meandloanstatusoftheperson.

Now,thecode wouldbenormalising thedatabyscalingittohavea

similarrangeofvalues,andthensplittingthatdataintoatrainingsetandatestsetfortr

ainingthemodelandtestingitsperformance,respectively.

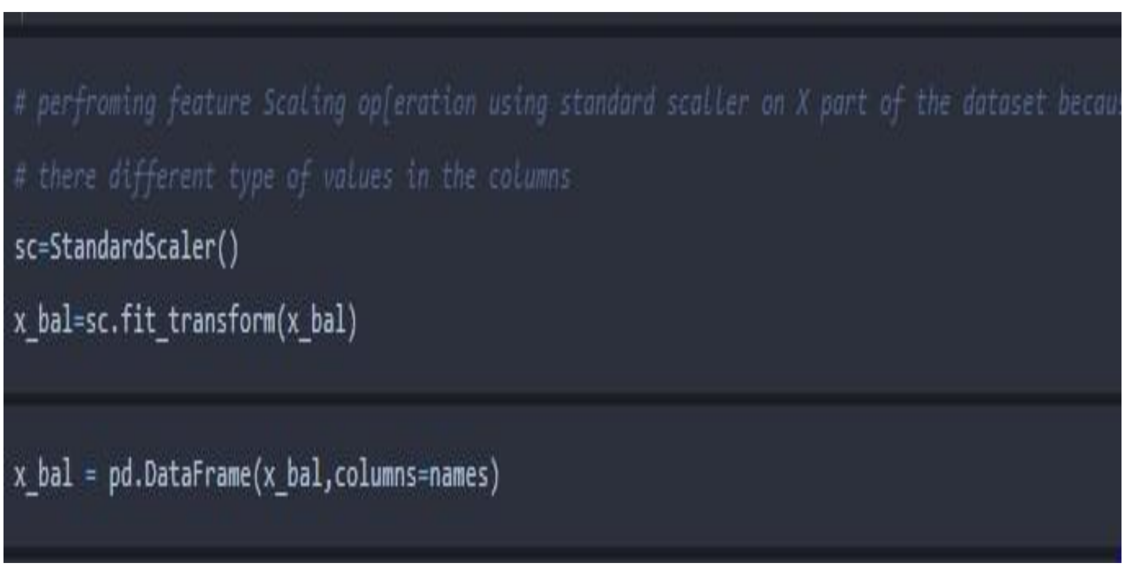
ScalingtheData:

Scalingisonetheimportantprocess,wehavetoperformonthedataset,becauseofd

atameasuresindifferentrangescanleadstomisleadinprediction

ModelssuchasKNN,Logisticregressionneedscaleddata,astheyfollowdistanceba

sedmethodandGradientDescentconcept.



```
# perfroming feature Scaling op[eration using standard scaller on X part of the dataset becau
# there different type of values in the columns
sc=StandardScaler()
x_bal=sc.fit_transform(x_bal)

x_bal = pd.DataFrame(x_bal,columns=names)
```

Wewillperformscalingonlyon theinputvalues.Oncethedatasetisscaled,itwillbeconvertedintoanarrayandweneedtoconvertitbacktoadataframe.

Splittingdataintotrainandtest:

Nowlet’ssplittheDatasetintotrainandtestsets

Changes:firstsplitthedatasetintoxandyandthensplitthedataset

Herexandyvariablesarecreated.On xvariable,dfispassedwithdroppingthetargetvariable.Andonytargetvariableispassed.Forsplittingtrainingandtestingdataweareusingthetrain_test_split()functionfromsklearn.Asparameters,wearepassingx,y,test_size,random_state.

```
#splitting the dataset in train and test on balnmced datasew
X_train, X_test, y_train, y_test = train_test_split(
     x_bal, y_bal, test_size=0.33, random_state=42)
```

<u>ModelBuilding:</u>

Activity1:Training the model in multiple algorithms

Nowourdataiscleanedandit’stimetobuildthemodel.Wecantrainourdataondifferentalgorithms.Forthisprojectweareapplyingfourclassificationalgorithms.Thebe stmodelissavedbasedonitsperformance.

Decision tree model:

AfunctionnameddecisionTreeiscreatedandtrainandtestdataarepassedastheparameters.Insidethefunction,DecisionTreeClassifieralgorithmisinitialisedandtrainingdataispassedtothemodelwiththe.fit()function. Testdataispredictedwith.predict()functionandsavedinanewvariable.Forevaluatingthemodel,aconfusionmatrixandclassificationreportisdone.

```
def decisionTree(x_train, x_test, y_train, y_test):
    dt=DecisionTreeClassifier()
    dt.fit(x_train,y_train)
    yPred = dt.predict(x_test)
    print('***DecisionTreeClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
```

Random forest model:

AfunctionnamedrandomForestiscreatedandtrainandtestdataarepassedastheparameters.Insidethefunction,RandomForestClassifieralgorithmisinitialisedandtrainingdataispassedtothemodelwith.fit()function.Testdataispredictedwith.predict()functionandsavedinanewvariable.Forevaluatingthemodel,aconfusionmatrixandclassificationreportisdone.

```
def randomForest(x_train, x_test, y_train, y_test):
    rf = RandomForestClassifier()
    rf.fit(x_train,y_train)
    yPred = rf.predict(x_test)
    print('***RandomForestClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
```

KNN model:

AfunctionnamedKNNiscreatedandtrainandtestdataarepassedastheparameters.Insid ethefunction,KNeighborsClassifieralgorithmisinitialisedandtrainingdataispassedtothe modelwith

.fit() function. Test data is predicted with .predict() function and saved in newvariable. For evaluating the model, confusion matrix and classification reportisdone.

```
def KNN(x_train, x_test, y_train, y_test):
    knn = KNeighborsClassifier()
    knn.fit(x_train,y_train)
    yPred = knn.predict(x_test)
    print('***KNeighborsClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
```

Xg boost model:

A function named xgboost is created and train and test data are passed astheparameters.Insidethefunction,GradientBoostingClassifieralgorithmis initialisedandtrainingdataispassedtothemodelwith.fit()function.Test

dataispredictedwith.predict()functionandsavedinnewvariable.Forevaluati

ngthemodel,confusionmatrixandclassificationreportisdone.

```python
def xgboost(x_train, x_test, y_train, y_test):
    xg = GradientBoostingClassifier()
    xg.fit(x_train,y_train)
    yPred = xg.predict(x_test)
    print('***GradientBoostingClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))
```

ANN model:

Building andtrainingAnArtificialNeuralNetwork(ANN)using

theKeraslibrarywithTensorFlowasthebackend.TheANNisinitialisedasaninstan

ceoftheSequentialclass,whichisalinearstackoflayers.Then,theinputlayerandtw

ohiddenlayersareaddedtothemodelusingtheDenseclass,wherethenumberofun

itsandactivationfunctionarespecified.TheoutputlayerisalsoaddedusingtheDens

eclasswithasigmoidactivationfunction.ThemodelisthencompiledwiththeAdamo

ptimizer,binarycross-

entropylossfunction,andaccuracymetric.Finally,themodelisfittothetrainingdata

withabatchsizeof100,20%validationsplit,and100epochs.

ANN

```
# Importing the Keras libraries and packages
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
[226] # Initialising the ANN
classifier = Sequential()
```

```
[227] # Adding the input layer and the first hidden layer
classifier.add(Dense(units=100, activation='relu', input_dim=11))
```

```
# Adding the second hidden layer
classifier.add(Dense(units=50, activation='relu'))
```

```
[229] # Adding the output layer
classifier.add(Dense(units=1, activation='sigmoid'))
```

```
[230] # Compiling the ANN
classifier.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
# Fitting the ANN to the Training set
model_history = classifier.fit(X_train, y_train, batch_size=100, validation_split=0.2, epochs=100)
```

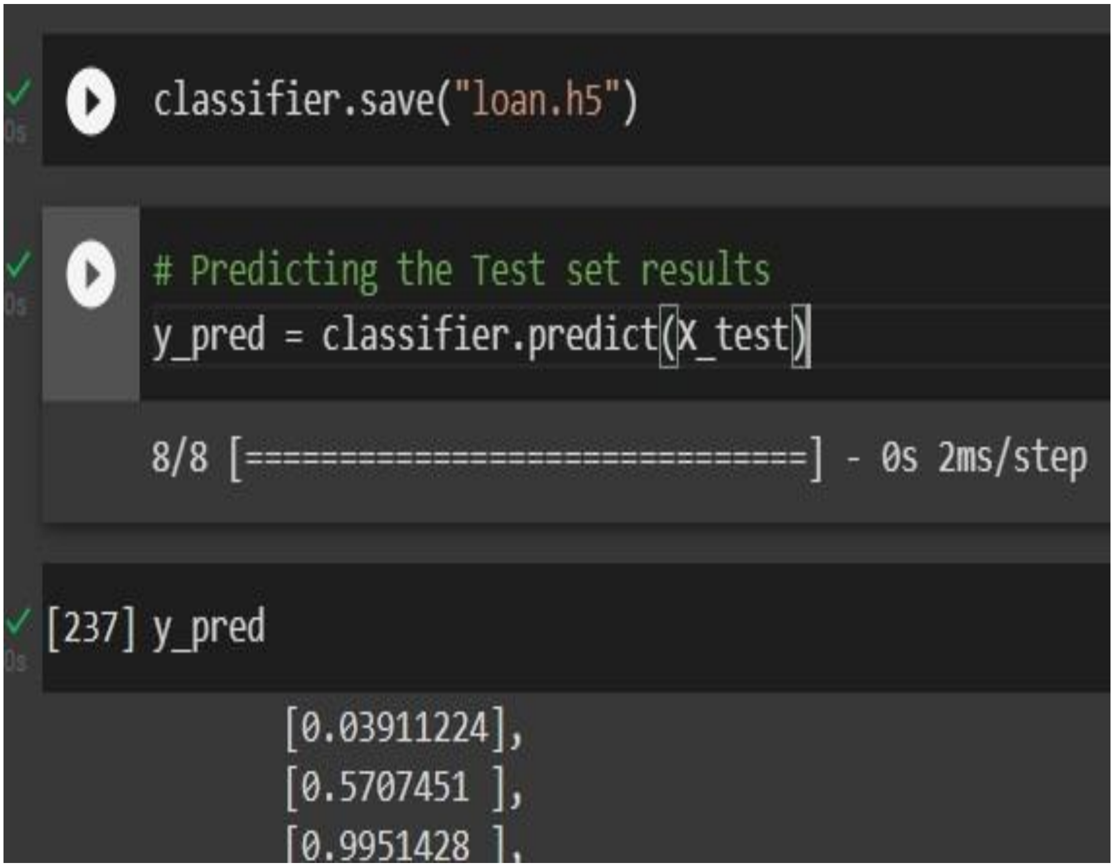
```
Epoch 72/100
4/4 [==============================] - 0s 11ms/step - loss: 0.4286 - accuracy: 0.7824 - val_loss: 0.7493 - val_accuracy: 0.6703
Epoch 73/100
4/4 [==============================] - 0s 12ms/step - loss: 0.4252 - accuracy: 0.8017 - val_loss: 0.7592 - val_accuracy: 0.6703
Epoch 74/100
4/4 [==============================] - 0s 12ms/step - loss: 0.4244 - accuracy: 0.8017 - val_loss: 0.7638 - val_accuracy: 0.6703
Epoch 75/100
4/4 [==============================] - 0s 11ms/step - loss: 0.4222 - accuracy: 0.7989 - val_loss: 0.7577 - val_accuracy: 0.6703
Epoch 76/100
4/4 [==============================] - 0s 14ms/step - loss: 0.4200 - accuracy: 0.7934 - val_loss: 0.7586 - val_accuracy: 0.6703
Epoch 77/100
4/4 [==============================] - 0s 11ms/step - loss: 0.4181 - accuracy: 0.7989 - val_loss: 0.7657 - val_accuracy: 0.6703
```

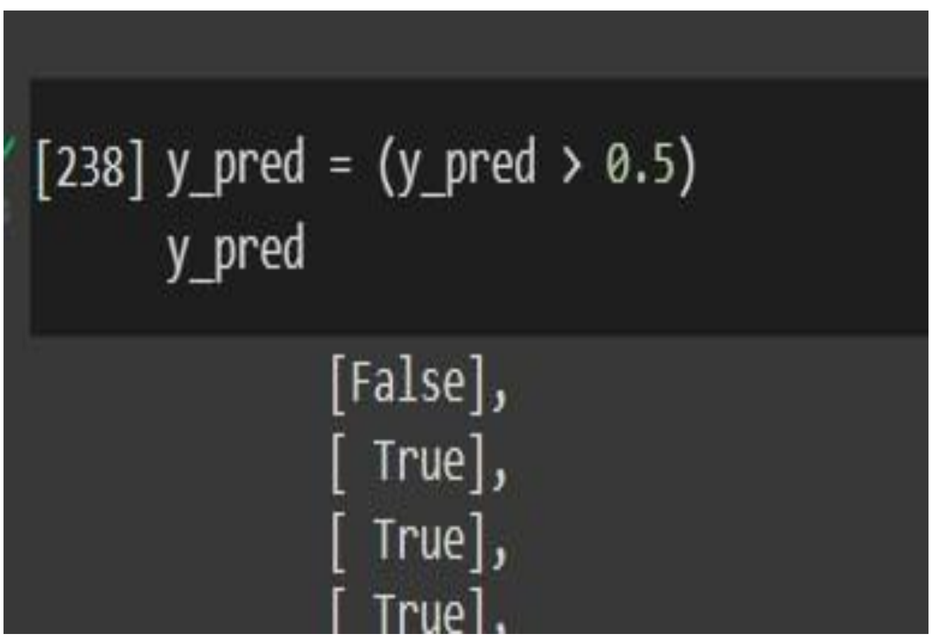
```
Epoch 95/100
4/4 [==============================] - 0s 17ms/step - loss: 0.3877 - accuracy: 0.8292 - val_loss: 0.8256 - val_accuracy: 0.6593
Epoch 96/100
4/4 [==============================] - 0s 13ms/step - loss: 0.3858 - accuracy: 0.8292 - val_loss: 0.8253 - val_accuracy: 0.6593
Epoch 97/100
4/4 [==============================] - 0s 13ms/step - loss: 0.3858 - accuracy: 0.8347 - val_loss: 0.8260 - val_accuracy: 0.6593
Epoch 98/100
4/4 [==============================] - 0s 12ms/step - loss: 0.3841 - accuracy: 0.8430 - val_loss: 0.8382 - val_accuracy: 0.6593
Epoch 99/100
4/4 [==============================] - 0s 12ms/step - loss: 0.3817 - accuracy: 0.8347 - val_loss: 0.8357 - val_accuracy: 0.6593
Epoch 100/100
4/4 [==============================] - 0s 11ms/step - loss: 0.3805 - accuracy: 0.8430 - val_loss: 0.8368 - val_accuracy: 0.6593
```

Testing the model:

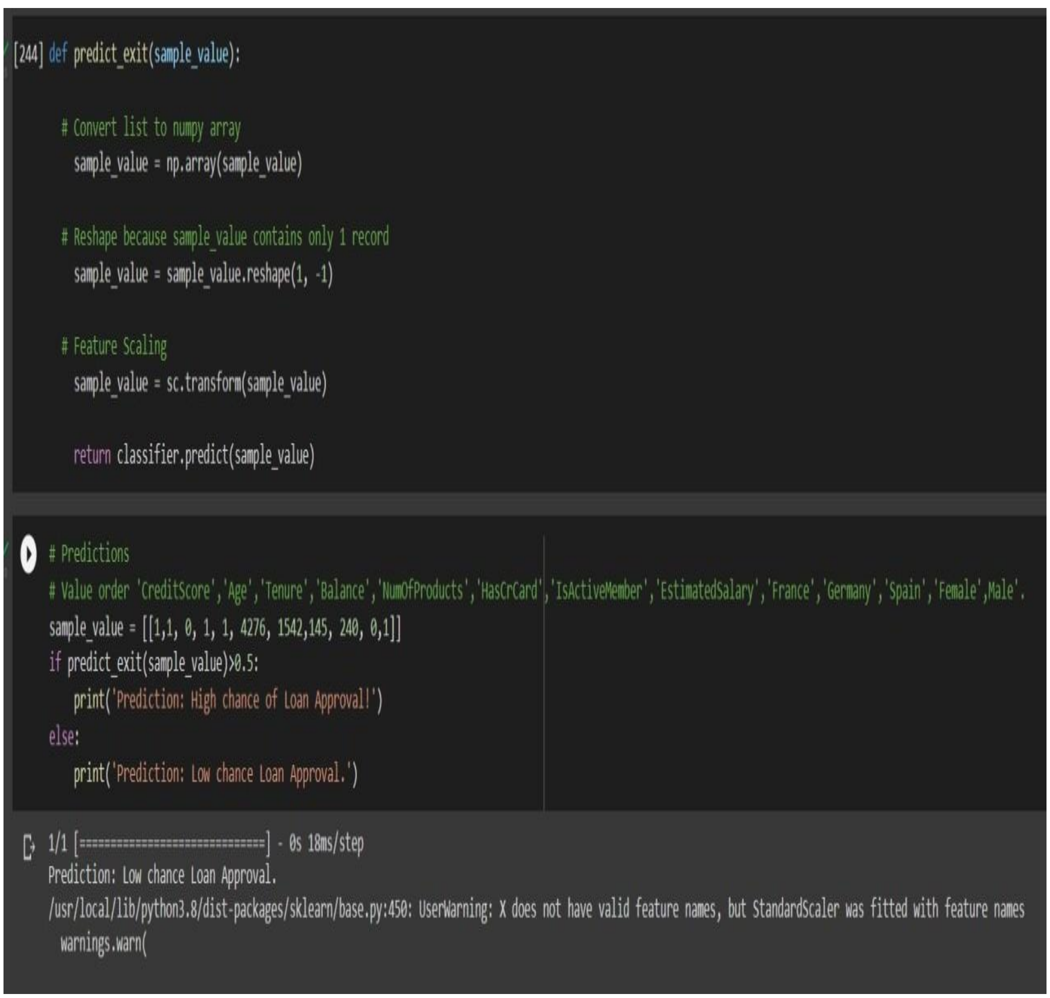
```
+ Code  + Text

[147] #Gender Married Dependents  Education Self_Employed ApplicantIncome CoapplicantIncome LoanAmount  Loan_Amount_Term  Credit_History  Property_Area
      dtr.predict([[1,1, 0, 1, 1, 4276, 1542,145, 240, 0,1]])

/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
array([0])

[149] #Gender Married Dependents  Education Self_Employed ApplicantIncome CoapplicantIncome LoanAmount  Loan_Amount_Term  Credit_History  Property_Area
      rfr.predict([[1,1, 0, 1, 1, 4276, 1542,145, 240, 0,1]])

/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
array([1])

      #Gender Married Dependents  Education Self_Employed ApplicantIncome CoapplicantIncome LoanAmount  Loan_Amount_Term  Credit_History  Property_Area
      knn.predict([[1,1, 0, 1, 1, 4276, 1542,145, 240, 0,1]])

/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but KNeighborsClassifier was fitted with feature names
  warnings.warn(
array([1])

[153] #Gender Married Dependents  Education Self_Employed ApplicantIncome CoapplicantIncome LoanAmount  Loan_Amount_Term  Credit_History  Property_Area
      xgb.predict([[1,1, 0, 1, 1, 4276, 1542,145, 240, 0,1]])

/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but GradientBoostingClassifier was fitted with feature names
  warnings.warn(
array([1])
```

InANNwefirsthavetosave themodeltothetesttheinputs





Thiscodedefinesafunctionnamed"predict_exit"whichtakesinasample_valueas aninput.Thefunctionthenconvertstheinputsample_valuefromalisttoanumpyarray.Itreshapesthesample_valuearrayasit containsonlyone record.Then,itappliesfeaturescalingtothereshapedsample_valuearrayusingascalerobject'sc'thatshouldhavebeenpreviouslydefinedandfitted.Finally,thefunctionreturnsthepredictionoftheclassifieron the scaledsample_value.



```
[244] def predict_exit(sample_value):

      # Convert list to numpy array
      sample_value = np.array(sample_value)

      # Reshape because sample_value contains only 1 record
      sample_value = sample_value.reshape(1, -1)

      # Feature Scaling
      sample_value = sc.transform(sample_value)

      return classifier.predict(sample_value)
```

```
# Predictions
# Value order 'CreditScore','Age','Tenure','Balance','NumOfProducts','HasCrCard','IsActiveMember','EstimatedSalary','France','Germany','Spain','Female',Male'.
sample_value = [[1,1, 0, 1, 1, 4276, 1542,145, 240, 0,1]]
if predict_exit(sample_value)>0.5:
    print('Prediction: High chance of Loan Approval!')
else:
    print('Prediction: Low chance Loan Approval.')
```

```
1/1 [==============================] - 0s 18ms/step
Prediction: Low chance Loan Approval.
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
```

```
def compareModel(X_train,X_test,y_train,y_test):
  decisionTree(X_train,X_test,y_train,y_test)
  print('-'*100)
  RandomForest(X_train,X_test,y_train,y_test)
  print('-'*100)
  XGB(X_train,X_test,y_train,y_test)
  print('-'*100)
  KNN(X_train,X_test,y_train,y_test)
  print('-'*100)
```

```
# Predictions
# Value order 'CreditScore','Age','Tenure','Balance','NumOfProducts','HasCrCard','IsActiveMember','EstimatedSalary','France','Germany','Spain','Female',Male'.
sample_value = [[1,0, 1, 1, 1, 45, 14,45, 240, 1,1]]
if predict_exit(sample_value)>0.5:
    print('Prediction: High chance of Loan Approval!')
else:
    print('Prediction: Low chance of Loan Approval.')
```

```
1/1 [==============================] - 0s 50ms/step
Prediction: High chance of Loan Approval!
/usr/local/lib/python3.8/dist-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
```

PerformanceTesting&HyperparameterTuning:

Testingmodelwithmultipleevaluationmetrics:

Multipleevaluationmetricsmeansevaluatingthemodel'sperformanceonatest setusingdifferentperformancemeasures.Thiscanprovideamorecomprehensiv eunderstandingofthemodel'sstrengthsandweaknesses.

Weareusingevaluationmetricsforclassificationtasksincludingaccuracy,precision,recal l,supportandF1-score.

Compare the model:

Forcomparingtheabovefourmodels,thecompareModelfunctionisdefined.

```
compareModel(X_train,X_test,y_train,y_test)
```

```
1.0
0.7822222222222223
Decision Tree
Confusion_Matrix
[[83 24]
 [25 93]]
Classification Report
              precision    recall  f1-score   support

           0       0.77      0.78      0.77       107
           1       0.79      0.79      0.79       118

    accuracy                           0.78       225
   macro avg       0.78      0.78      0.78       225
weighted avg       0.78      0.78      0.78       225

-----------------------------------------------------------------
```

```
-----------------------------------------------------------------
1.0
0.8088888888888889
Random Forest
Confusion_Matrix
[[ 78  29]
 [ 14 104]]
Classification Report
              precision    recall  f1-score   support

           0       0.85      0.73      0.78       107
           1       0.78      0.88      0.83       118

    accuracy                           0.81       225
   macro avg       0.81      0.81      0.81       225
weighted avg       0.81      0.81      0.81       225

-----------------------------------------------------------------
```

```
0.933920704845815
0.8222222222222222
XGBoost
Confusion_Matrix
[[ 78  29]
 [ 11 107]]
Classification Report
              precision    recall  f1-score   support

           0       0.88      0.73      0.80       107
           1       0.79      0.91      0.84       118

    accuracy                           0.82       225
   macro avg       0.83      0.82      0.82       225
weighted avg       0.83      0.82      0.82       225
```

```
0.7665198237885462
0.6666666666666666
KNN
Confusion_Matrix
[[60 47]
 [28 90]]
Classification Report
              precision    recall  f1-score   support

           0       0.68      0.56      0.62       107
           1       0.66      0.76      0.71       118

    accuracy                           0.67       225
   macro avg       0.67      0.66      0.66       225
weighted avg       0.67      0.67      0.66       225
```

```
yPred = classifier.predict(X_test)
print(accuracy_score(y_pred,y_test))
print("ANN Model")
print("Confusion_Matrix")
print(confusion_matrix(y_test,y_pred))
print("Classification Report")
print(classification_report(y_test,y_pred))
```

```
8/8 [==============================] - 0s 4ms/step
0.6844444444444444
ANN Model
Confusion_Matrix
[[63 44]
 [27 91]]
Classification Report
              precision    recall  f1-score   support

           0       0.70      0.59      0.64       107
           1       0.67      0.77      0.72       118

    accuracy                           0.68       225
   macro avg       0.69      0.68      0.68       225
weighted avg       0.69      0.68      0.68       225
```

Aftercallingthefunction,theresultsofmodelsaredisplayedasoutput.Fromthefive modelsXgboostisperformingwell.Fromthebelowimage,Wecanseetheaccuracy ofthemodel.Xgboostisgivingtheaccuracyof93.39% withtrainingdata,82.2%accuracyforthetestingdata.

```
from sklearn.model_selection import cross_val_s

# Random forest model is selected

rf = RandomForestClassifier()
rf.fit(x_train,y_train)
yPred = rf.predict(x_test)

f1_score(yPred,y_test,average='weighted')
```

Comparingmodelaccuracybefore&afterapplyinghyperparametertuni

ng:

EvaluatingperformanceofthemodelFromsklearn,cross_val_scoreisused to evaluate

thescoreofthemodel.Ontheparameters,wehavegivenrf(modelname),x,y,cv(as5f

olds).Ourmodelisperformingwell.So,wearesavingthemodelbypickle.dump().

Note: Tounderstandcrossvalidation,refer~~to~~thislink

```
0.9691629955947136
0.8222222222222222
Random Forest
Confusion_Matrix
[[ 77  30]
 [ 10 108]]
Classification Report
              precision    recall  f1-score   support

           0       0.89      0.72      0.79       107
           1       0.78      0.92      0.84       118

    accuracy                           0.82       225
   macro avg       0.83      0.82      0.82       225
weighted avg       0.83      0.82      0.82       225

[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
```

Model Deployment:

Savethebestmodel:

Savingthebest modelafter comparingitsperformanceusingdifferentevaluationmetricsmeansselectingthe modelwiththehighestperformanceandsavingitsweightsandconfiguration.Thisc anbeusefulinavoidingtheneedtoretrainthemodeleverytimeit isneededandalsotobeabletouseitinthefuture.

```
#saviung the model by using pickle function
pickle.dump(model,open('rdf.pkl','wb'))
```

IntegratewithWebFramework:

In this section, we will be building a web application that is integrated to themodelwe built. A UIisprovidedfortheuseswhere he hastoenterthevalues for predictions. The enter values are given to the saved model andpredictionisshowcasedontheUI.

Thissectionhasthefollowingtasks

- BuildingHTMLPages
- Buildingserversidescript
- Runthewebapplication

BuildingHtmlPages:

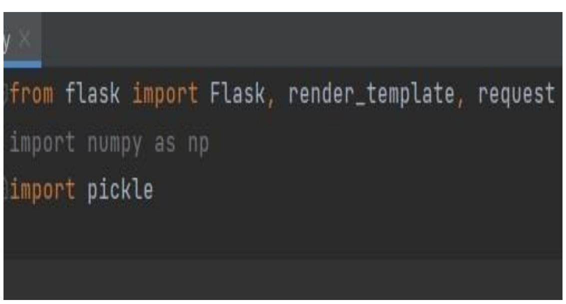
ForthisprojectcreatetwoHTMLfilesnamely

- home.html
- predict.html

andsavethemin thetemplatesfolder.

BuildPythoncode:

Importthelibraries



```
from flask import Flask, render_template, request
import numpy as np
import pickle
```

Loadthesavedmodel.Importingtheflaskmoduleintheprojectismandatory.Anobj

ectofFlaskclassisourWSGIapplication.Flaskconstructortakes thename of thecurrentmodule

(name___)asargument.

```python
app = Flask(__name__)
model = pickle.load(open(r'rdf.pkl', 'rb'))
scale = pickle.load(open(r'scale1.pkl','rb'))
```

RenderHTMLpage:

```python
@app.route('/') # rendering the html template
def home():
    return render_template('home.html')
```

whichwehavecreatedearlier.

Intheaboveexample,'/'URL isboundwiththehome.htmlfunction.Hence,whenthehomepageofthewebserverisopenedinthebrowser,thehtmlpagewillberendered.Wheneveryouenterthevaluesfromthe htmlpagethevaluescanberetrievedusingPOST Method

HerewewillbeusingadeclaredconstructortoroutetotheHTMLpage

.

RetrievesthevaluefromUI:

```python
@app.route('/submit',methods=["POST","GET"])# route to show the predictions in a web UI
def submit():
    #  reading the inputs given by the user
    input_feature=[int(x) for x in request.form.values() ]
    #input_feature = np.transpose(input_feature)
    input_feature=[np.array(input_feature)]
    print(input_feature)
    names = ['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'ApplicantIncome',
       'CoapplicantIncome','LoanAmount','Loan_Amount_Term','Credit_History','Property_Area']
    data = pandas.DataFrame(input_feature,columns=names)
    print(data)

    #data_scaled = scale.fit_transform(data)
    #data = pandas.DataFrame(,columns=names)



     # predictions using the loaded model file
    prediction=model.predict(data)
    print(prediction)
    prediction = int(prediction)
    print(type(prediction))

    if (prediction == 0):
       return render_template("output.html",result ="Loan will Not be Approved")
    else:
       return render_template("output.html",result = "Loan will be Approved")
     # showing the prediction results in a UI
```

HereweareroutingourapptopredictOfunction.Thisfunctionretrievesallthevalues

fromtheHTMLpageusingPostrequest.Thatisstoredinanarray.Thisarrayispasse

dtothemodel.predict()

function.Thisfunctionreturnstheprediction.Andthispredictionvaluewillberender

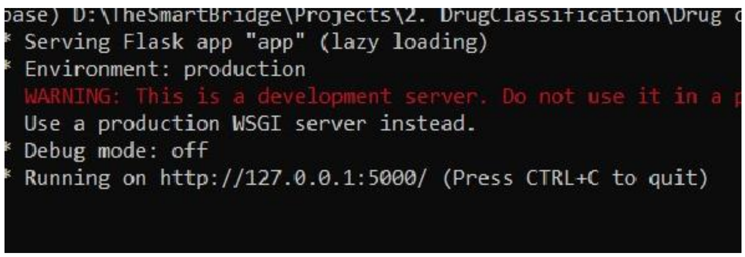
edtothetextthatwehavementionedinthesubmit.htmlpageearlier.

MainFunction:

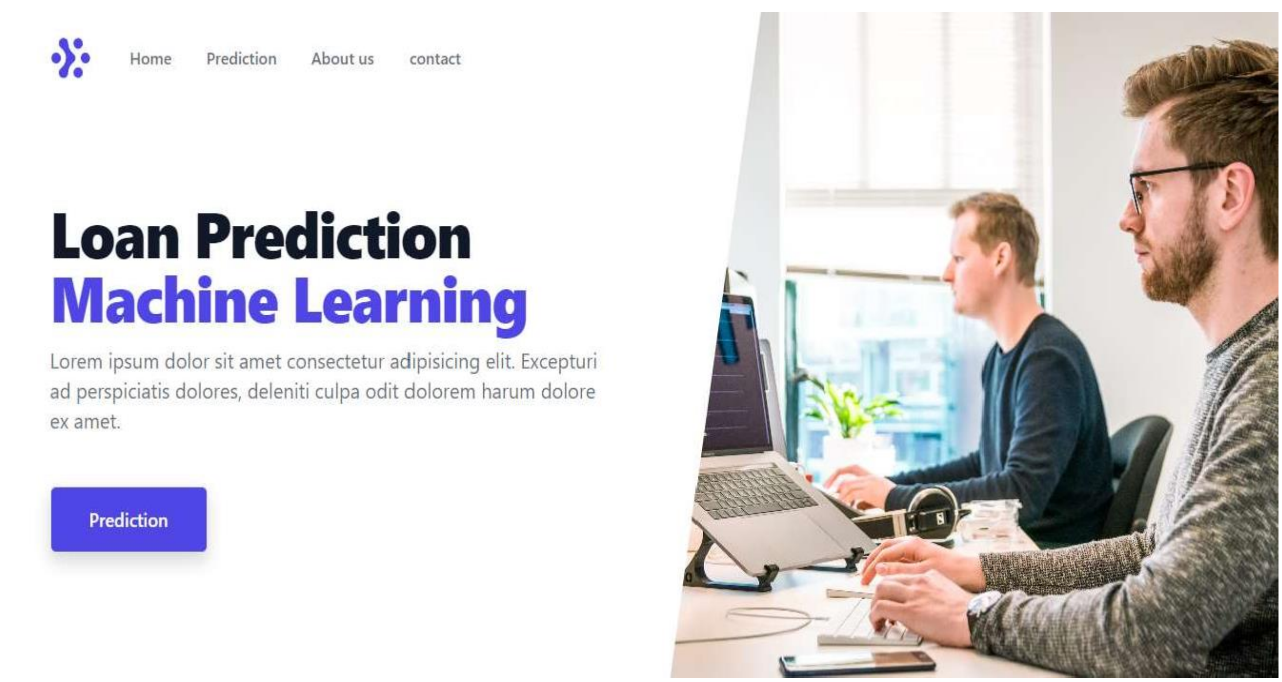
```python
if __name__=="__main__":

    # app.run(host='0.0.0.0', port=8000,debug=True)    # running the app
    port=int(os.environ.get('PORT',5000))
    app.run(debug=False)
```

Runthewebapplication:

- Openanacondapromptfromthestartmenu
- Navigatetothe folderwhereyourpythonscriptis.
- Nowtype“pythonapp.py”command
- Navigatetothelocalhostwhereyoucanviewyourwebpage.
- Clickonthepredictbuttonfromthetop leftcorner,entertheinputs,clickonthesubmitbutton,andseetheresult/predictionontheweb.



Now,Gothewebbrowserandwritethelocalhosturl(http://127.0.0.1:5000)togetthebelowresult



Now,whenyouclickonclickmetopredictthebuttonfromthebanneryouwillget

redirectedtothepredictionpage.

Input1-Now,theuserwillgiveinputstogetthepredicted resultafterclickingonto

thesubmitbutton.

Youwillgettheoutput.

CONCLUSION:

The predictive models based on Decision Tree and Random Forest, give the accuracy as80.945%,93.648%and 83.388%whereasthecross-validation isfound tobe 80.945%,72.213%and80.130%respectively.Thisshowsthatforthegivendataset,theaccuracyof modelbasedondecisiontreeishighestbutrandomforestisbetteratgeneralizationeventhoughit’ scrossvalidationisnotmuchhigherthanothermodels.