Loan Approval Prediction

A COURSE PROJECT REPORT
By

Abhi Mukeshumar Patel (RA2011026010078)

Under the guidance of

Dr. Suresh K.

In partial fulfilment for the Course of

18CSE479T - Statistical Machine Learning

in CINTEL

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Chenpalpattu District

NOVEMBER 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project report " **Loan Approval Prediction**" is the bonafide work of **Abhi Mukeshkumar Patel (RA2011026010078)** who carried out the project work under my supervision.

Signature of the Guide	**Signature of the HoD**
Dr. K Suresh	Dr. Annie Uthra
Assistant Professor	**Professor and Head**
Department of CINTEL	Department of CINTEL

About the course:-

18CSC479T - Statistical Machine Learning is three credit course with **L T P C as 3-0-0-3**

Objectives:

The student should be made to:

- Learn the basics of statistical machine learning techniques.
- Learn the basics of build model based on logistic regression and random forest techniques
- Learn basic idea of ideas of probability and work on probabilistic approaches like Naïve Bayes, Bayes Theorem
- Be familiar with knowledge of Kernel functions in practical applications
- Be familiar with knowledge of K-means clustering on real world examples
- Learn PCA and SVD with Scikit-learn

Course Learning Rationale (CLR): The purpose of learning this course is to:

Course Learning Rationale (CLR):	The purpose of learning this course is to:
CLR-1 :	Understand the Fuzzy Logic Basics
CLR-2 :	Gain knowledge on the Machine learning concepts
CLR-3 :	Gain knowledge on Fuzzy based clustering concepts
CLR-4 :	Acquire knowledge on Fuzzy Integrated classification
CLR-5 :	Understanding Neuro-Fuzzy Modeling concepts
CLR-5 :	Acquiring better understanding on Fuzzy logic usage
CLR-6	Understanding the fuzzylogics in Machine learning

Course Learning Outcomes (CLO): At the end of this course, learners will be able to:

Course Learning Outcomes (CLO):	At the end of this course, learners will be able to:
CLO-1 :	Acquire the knowledge on statistical machine learning techniques.
CLO-2 :	Acquire the ability to build model based on logistic regression and random forest techniques
CLO-3 :	Understand the basic ideas of probability and work on probabilistic approaches like Naïve Bayes, Bayes Theorem
CLO-4 :	Apply the knowledge of Kernel functions in practical applications
CLO-5 :	Apply the knowledge of K-means clustering on real world examples
CLO-6 :	Acquire the knowledge on using PCA and SVD with Scikit-learn

Table 1: Internal Mark Split-up:- As per Curriculum

Learning Assessment											
	Bloom's Level of Thinking	Continuous Learning Assessment (50%								Final Examination (50% weightage)	
		CLA – 1 (10%)		CLA – 2 (15%)		CLA – 3 (15%)		CLA – 4 (10%)#			
		Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice	Theory	Practice
Level 1	Remember Understand	*40 %*	-	*30 %*	-	*30 %*	-	*30 %*	-	*30%*	-
Level 2	Apply Analyze	*40 %*	-	*40 %*	-	*40 %*	-	*40 %*	-	*40%*	-
Level 3	Evaluate Create	*20 %*	-	*30 %*	-	*30 %*	-	*30 %*	-	*30%*	-
	Total	100 %		100 %		100 %		100 %		100 %	

 # CLA – 4 can be from any combination of these: Assignments, Seminars, Tech Talks, Mini-Projects, Case-Studies, Self-Study, MOOCs, Certifications, Conf. Paper etc.,

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 Dr.Annapurani Panaiyappan, Professor and Head, Department of Networking and Communications** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty Dr Suresh K.**, Assistant Professor, CINTEL,** for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD Dr. Anni Uthra, Professor, CINTEL** 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.

CONTENT

ABSTRACT

For our project we have decided to experiment, design, and implement a loan prediction problem. We will use the loan prediction dataset to automate the loan eligibility process(real time) based on customer details provided like Education, Loan amount, Credit history, Income etc. Since we have to classify whether the loan will get approved or not so this is a Classification problem which comes under Supervised Machine Learning.

The dataset we will be working on has 615 rows & 13 columns. We will need to preprocess the data, perform data cleaning & feature engineering and finally will be implementing models like Logistic Regression, Decision tree, Random Forest and XGBoost to check the accuracy of each model.

Title and Introduction

1.1 Motivation and Objective

Machine Learning has become increasingly popular tool in almost every industry out there be it Smartphone Industry, Healthcare or Banks. With Machine Learning we can achieve so much. We will be using machine learning algorithms along with some data analysis techniques in our project.

Our project focus is to use existing customer’s details and analyze it further by applying a few machine learning techniques and predict which future applicant can be approved the loan.

1.2 Problem Statement

Dream Housing Finance Company deals in all home loans. Its customers first apply for home loan after that company validates the customer eligibility for loan**.** Since the process is time taking and tedious, the company decided to automate the loan approval process using machine learning.

1.3 Machine Learning

Machine Learning is an idea to learn from examples and experience, without being explicitly programmed.

“A computer program is said to learn from experience E with some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.” -Tom M. Mitchel

1.3.1 Examples of Machine Learning

There are many examples of machine learning. Here are a few examples of classification problems where the goal is to categorize objects into a fixed set of categories.

Face detection: Identify faces in images (or indicate if a face is present).

Email filtering: Classify emails into spam and not-spam.

Medical diagnosis: Diagnose a patient as a sufferer or non-sufferer of some disease.

Weather prediction: Predict, for instance, whether or not it will rain tomorrow.

Dataset Description

	Loan_ID	Gender	Married	Dependen	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_	Credit_Histor	Property_Area	Loan_Status
2	LP001002	Male	No	0	Graduate	No	5849	0		360	1	Urban	Y
3	LP001003	Male	Yes	1	Graduate	No	4583	1508	128	360	1	Rural	N
4	LP001005	Male	Yes	0	Graduate	Yes	3000	0	66	360	1	Urban	Y
5	LP001006	Male	Yes	0	Not Gradu	No	2583	2358	120	360	1	Urban	Y
6	LP001008	Male	No	0	Graduate	No	6000	0	141	360	1	Urban	Y
7	LP001011	Male	Yes	2	Graduate	Yes	5417	4196	267	360	1	Urban	Y
8	LP001013	Male	Yes	0	Not Gradu	No	2333	1516	95	360	1	Urban	Y
9	LP001014	Male	Yes	3+	Graduate	No	3036	2504	158	360	0	Semiurban	N
10	LP001018	Male	Yes	2	Graduate	No	4006	1526	168	360	1	Urban	Y
11	LP001020	Male	Yes	1	Graduate	No	12841	10968	349	360	1	Semiurban	N
12	LP001024	Male	Yes	2	Graduate	No	3200	700	70	360	1	Urban	Y
13	LP001027	Male	Yes	2	Graduate		2500	1840	109	360	1	Urban	Y
14	LP001028	Male	Yes	2	Graduate	No	3073	8106	200	360	1	Urban	Y
15	LP001029	Male	No	0	Graduate	No	1853	2840	114	360	1	Rural	N
16	LP001030	Male	Yes	2	Graduate	No	1299	1086	17	120	1	Urban	Y
17	LP001032	Male	No	0	Graduate	No	4950	0	125	360	1	Urban	Y
18	LP001034	Male	No	1	Not Gradu	No	3596	0	100	240		Urban	Y
19	LP001036	Female	No	0	Graduate	No	3510	0	76	360	0	Urban	N
20	LP001038	Male	Yes	0	Not Gradu	No	4887	0	133	360	1	Rural	N
21	LP001041	Male	Yes	0	Graduate		2600	3500	115		1	Urban	Y
22	LP001043	Male	Yes	0	Not Gradu	No	7660	0	104	360	0	Urban	N
23	LP001046	Male	Yes	1	Graduate	No	5955	5625	315	360	1	Urban	Y
24	LP001047	Male	Yes	0	Not Gradu	No	2600	1911	116	360	0	Semiurban	N
25	LP001050		Yes	2	Not Gradu	No	3365	1917	112	360	0	Rural	N
26	LP001052	Male	Yes	1	Graduate		3717	2925	151	360		Semiurban	N
27	LP001066	Male	Yes	0	Graduate	Yes	9560	0	191	360	1	Semiurban	Y

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncor	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area
2	LP001015	Male	Yes	0	Graduate	No	6756.756757	0	33.78378378	60	1	Urban
3	LP001022	Male	Yes	1	Graduate	No	3076	1500	126	360	1	Urban
4	LP001031	Male	Yes	2	Graduate	No	5000	1800	208	360	1	Urban
5	LP001035	Male	Yes	2	Graduate	No	2340	2546	100	360		Urban
6	LP001051	Male	No	0	Not Gradu	No	3276	0	78	360	1	Urban
7	LP001054	Male	Yes	0	Not Gradu	Yes	2165	3422	152	360	1	Urban
8	LP001055	Female	No	1	Not Gradu	No	2226	0	59	360	1	Semiurban
9	LP001056	Male	Yes	2	Not Gradu	No	3881	0	147	360	0	Rural
10	LP001059	Male	Yes	2	Graduate		13633	0	280	240	1	Urban
11	LP001067	Male	No	0	Not Gradu	No	2400	2400	123	360	1	Semiurban
12	LP001078	Male	No	0	Not Gradu	No	3091	0	90	360	1	Urban
13	LP001082	Male	Yes	1	Graduate		2185	1516	162	360	1	Semiurban
14	LP001083	Male	No	3+	Graduate	No	4166	0	40	180		Urban
15	LP001094	Male	Yes	2	Graduate		12173	0	166	360	0	Semiurban
16	LP001096	Female	No	0	Graduate	No	4666	0	124	360	1	Semiurban
17	LP001099	Male	No	1	Graduate	No	5667	0	131	360	1	Urban
18	LP001105	Male	Yes	2	Graduate	No	4583	2916	200	360	1	Urban
19	LP001107	Male	Yes	3+	Graduate	No	3786	333	126	360	1	Semiurban
20	LP001108	Male	Yes	0	Graduate	No	9226	7916	300	360	1	Urban
21	LP001115	Male	No	0	Graduate	No	1300	3470	100	180	1	Semiurban
22	LP001121	Male	Yes	1	Not Gradu	No	1888	1620	48	360	1	Urban
23	LP001124	Female	No	3+	Not Gradu	No	2083	0	28	180	1	Urban
24	LP001128		No	0	Graduate	No	3909	0	101	360	1	Urban
25	LP001135	Female	No	0	Not Gradu	No	3765	0	125	360	1	Urban
26	LP001149	Male	Yes	0	Graduate	No	5400	4380	290	360	1	Urban
27	LP001153	Male	No	0	Graduate	No	0	24000	148	360	0	Rural

The dataset we will be working on has 615 rows & 13 columns. We will need to preprocess the data, perform data cleaning & feature engineering and finally will be implementing models like Logistic Regression, Decision tree, Random Forest and XGBoost to check the accuracy of each model.

Modules Description

Process:

- Hypothesis Generation
- Understanding the Data
- Exploratory Data Analysis (EDA)
    - Univariate analysis
    - Bivariate analysis
- Missing Value and Outlier Treatment
- Feature Engineering
- Model Building
    - Logistic Regression
    - Decision Tree
    - Random Forest
    - XGBoost

Software Requirements:

- Python
- Anaconda (JUPYTER Notebook)

LIBRARIES USED:

- NumPy
- SciKit Learn
- Pandas
- MatplotLib
- Seaborn
- XGBoost

1.2 Hypothesis Generation

This is the first and foremost step which is performed even before looking at the data. It involves understanding the problem in detail by brainstorming as many factors as we can.

Below are some of the factors which I think can affect the Loan Approval (dependent variable for this loan prediction problem):

- **Salary**: Applicants with high income should have more chances of loan approval.
- **Previous history**: Applicants who have repaid their previous debts should have higher chances of loan approval.
- **Loan amount**: Loan approval should also depend on the loan amount. If the loan amount is less, chances of loan approval should be high.
- **Loan term**: Loan for less time period and less amount should have higher chances of approval.
- **EMI**: Lesser the amount to be paid monthly to repay the loan, higher the chances of loan approval.

1.3 Reading the data

Train file will be used for training the model, i.e. our model will learn from this file. It contains all the independent variables and the target variable.

Test file contains all the independent variables, but not the target variable. We will apply the model to predict the target variable for the test data.

```
train=pd.read_csv("train.csv")

test=pd.read_csv("test.csv")
```

1.4 Understanding the Data

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)

```
# Print data types for each variable
train. Types
Loan_ID                  object
Gender                   object
Married                  object
Dependents               object
Education                object
Self_Employed            object
ApplicantIncome           int64
CoapplicantIncome       float64
LoanAmount              float64
Loan_Amount_Term        float64
Credit_History          float64
Property_Area            object
Loan_Status              object
type: object
```

We can see there are three format of data types:

- object: Object format means variables are categorical. Categorical variables in our dataset are: Loan_ID, Gender, Married, Dependents, Education, Self_Employed, Property_Area, Loan_Status
- int64: It represents the integer variables. ApplicantIncome is of this format.
- float64: It represents the variable which have some decimal values involved. They are also numerical variables. Numerical variables in our dataset are: CoapplicantIncome, LoanAmount, Loan_Amount_Term, and Credit_History

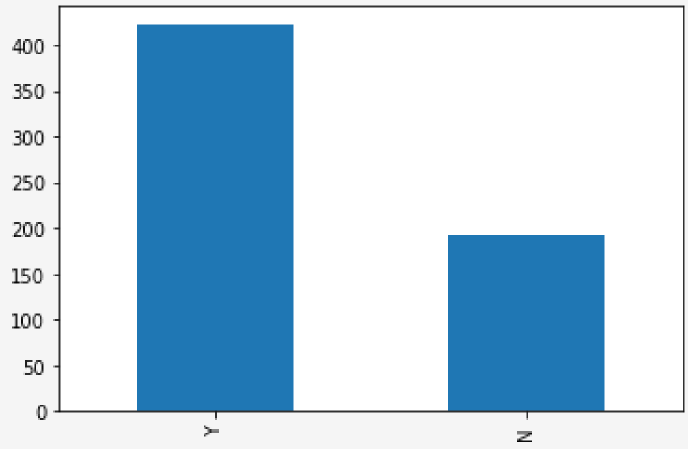
1.5 Univariate Analysis

In this we examine each variable individual. For categorical features (Loan_ID, Gender, Married, Dependents etc.) frequency table or bar plots will be used which will calculate the number of each category in a particular variable. For numerical features, probability density plots can be used to look at the distribution of the variable.

```
train['Loan_Status'].value_counts(normalize=True)
```

```
Y    0.687296
N    0.312704
Name: Loan_Status, dtype: float64
```

```
train['Loan_Status'].value_counts().plot.bar()
```



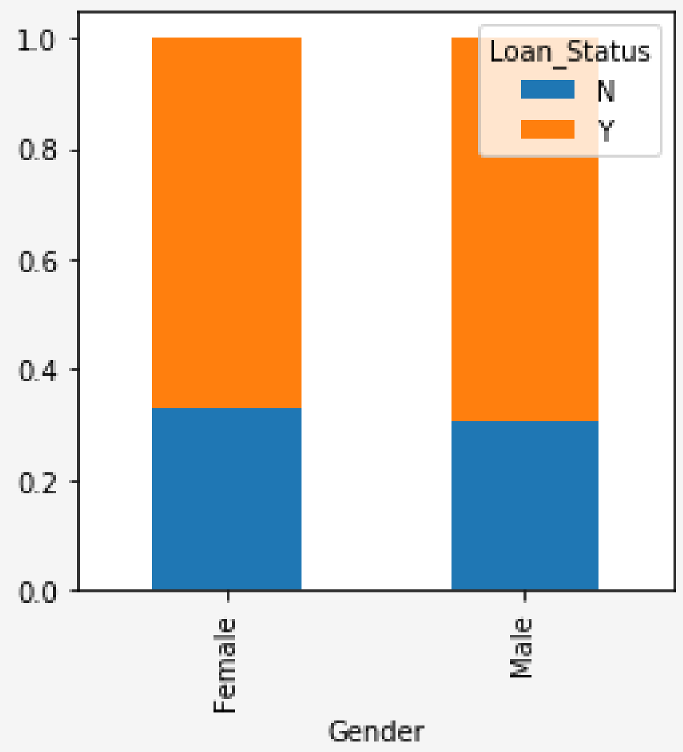
The loan of 422(around 69%) people out of 614 was approved.

1.6 Bivariate Analysis

After looking at every variable individually in univariate analysis, we will now explore them again with respect to the target variable.

```
Gender=pd.crosstab(train['Gender'],train['Loan_Status'])
```

```
Gender.div(Gender.sum(1).astype(float), axis=0).plot(kind="bar",
stacked=True, figsize=(4,4))
```



It can be inferred that the proportion of male and female applicants is more or less same for both approved and unapproved loans.

1.7 Missing value and Outlier Treatment

- **Missing value Imputation**

After exploring all the variables in our data, we can now impute the missing values and treat the outliers because missing data and outliers can have adverse effect on the model performance.

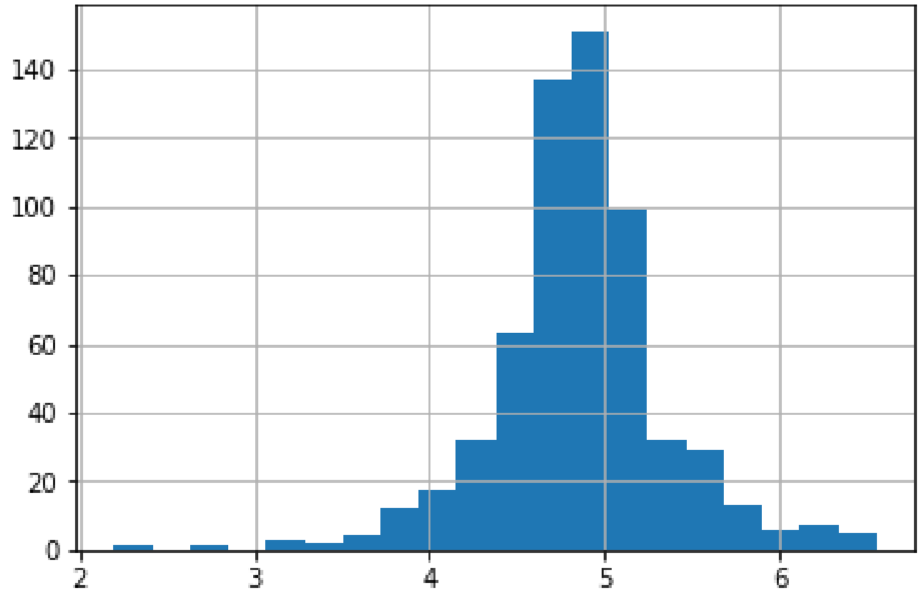
- **For numerical variables:** imputation using mean or median
- **For categorical variables:** imputation using mode

• Outlier Treatment

As we saw earlier in univariate analysis, LoanAmount contains outliers so we have to treat them as the presence of outliers affects the distribution of the data.

Due to these outliers bulk of the data in the loan amount is at the left and the right tail is longer. This is called right skew-ness. One way to remove the skew-ness is by doing the log transformation. As we take the log transformation, it does not affect the smaller values much, but reduces the larger values. So, we get a distribution similar to normal distribution.

```
train ['LoanAmount_log'] = np.log(train['LoanAmount'])
train['LoanAmount_log'].hist(bins=20)
test['LoanAmount_log'] = np.log(test['LoanAmount'])
```



Now the distribution looks much closer to normal and effect of extreme values has been significantly subsided.

1.8 Feature Engineering

Based on the domain knowledge, we can come up with new features that might affect the target variable. We will create the following three new features:

- **Total Income** - As discussed during bivariate analysis we will combine the Applicant Income and Coapplicant Income. If the total income is high, chances of loan approval might also be high.
- **EMI** - EMI is the monthly amount to be paid by the applicant to repay the loan. Idea behind making this variable is that people who have high EMI's might find it difficult to pay back the loan. We can calculate the EMI by taking the ratio of loan amount with respect to loan amount term.
- **Balance Income** - This is the income left after the EMI has been paid. Idea behind creating this variable is that if this value is high, the chances are high that a person will repay the loan and hence increasing the chances of loan approval.

```
train['Total_Income']=train['ApplicantIncome']+train['CoapplicantIncome']

test['Total_Income']=test['ApplicantIncome']+test['CoapplicantIncome']
```

We will add rest of features in similar way.

```
train['EMI']=train['LoanAmount']/train['Loan_Amount_Term']

test['EMI']=test['LoanAmount']/test['Loan_Amount_Term']

train['Balance Income']=train['Total_Income']-(train['EMI']*1000) # Multipl
y with 1000 to make the units equal

test['Balance Income']=test['Total_Income']-(test['EMI']*1000)
```

1.9 Model Building

After creating new features, we can continue the model building process. So we will start with logistic regression model and then move over to more complex models like Random Forest and XGBoost.

We will build the following models.

- Logistic Regression
- Decision Tree
- Random Forest
- XGBoost

Let's prepare the data for feeding into the models.

```
X = train.drop('Loan_Status',1)

y = train.Loan_Status      # Save target variable in separate dataset

#loan id don't have an effect on the outcome

train=train.drop('Loan_ID',axis=1)

test=test.drop('Loan_ID',axis=1)



"""Sklearn requires the target variable in a separate dataset.

so, we will drop our target variable from the train dataset and save it
in another dataset."""

x = train.drop('Loan_Status',1)

y = train.Loan_Status    # Save target variable in separate dataset



#As logistic regression takes only the numerical values as input, we ha
ve to change every categorical variable to continious
```

```
x=pd.get_dummies(x)

train=pd.get_dummies(train)

test=pd.get_dummies(test)



#we will use train_test_split function of sklearn to validate our predi
ctions

from sklearn.model_selection import train_test_split

x_train,x_cv,y_train,y_cv = train_test_split(x,y, test_size=0.3, random
_state=123)
```

In Train Test Split validation we split the training data into training and validating data as to evaluate our model. Since we don' have the outcome for Test data so we can't check its accuracy after predicting the outcome as we don't have original outcomes to compare with.

Train Test Split is a validation technique used to solve the above issue and hence evaluate our model.

Logistic Regression

Here we build our model using the popular Logistic Regression. Logistic regression is an estimation of Logit function. Logit function is simply a log of odds in favor of the event.

```
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score

LR=LogisticRegression()

LR.fit(x_train,y_train)




pred_cv=LR.predict(x_cv)
```

```
accuracy_score(y_cv,pred_cv)
```

OUTPUT : 0.7783783783783784

We got an accuracy score as 0.778 for this model.

- ## Decision Tree

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables.

Decision trees use multiple algorithms to decide to split a node in two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that purity of the node increases with respect to the target variable.

```
from sklearn import tree

DT = tree.DecisionTreeClassifier()

DT.fit(x_train, y_train)

pred_cv = DT.predict(x_cv)

accuracy_score(y_cv,pred_cv)
```

OUTPUT: 0.6702702702702703

We got an accuracy score of 0.6702 for this model.

- ## Random Forest

Random Forest is a tree based bootstrapping algorithm wherein a certain no. of weak learners (decision trees) are combined to make a powerful prediction model.

For every individual learner, a random sample of rows and a few randomly chosen variables are used to build a decision tree model.

Final prediction can be a function of all the predictions made by the individual learners.

```
from sklearn.ensemble import RandomForestClassifier

RF = RandomForestClassifier()

RF.fit(x_train,y_train)

pred_cv=RF.predict(x_cv)

accuracy_score(y_cv,pred_cv)
```

OUTPUT: 0.7297297297297297

We got an accuracy score of 0.729 for this model.

- ## XGBoost

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

```
from xgboost import XGBClassifier

XG= XGBClassifier()

XG.fit(x_train,y_train)

pred_cv=XG.predict(x_cv)

accuracy_score(y_cv,pred_cv)
```

OUTPUT: 0.7675675675675676

We got an accuracy score of 0.767 close to Logistic Regression.

```
In [1360]: pred_test = LR.predict(test)

In [1361]: df=pd.DataFrame(pred_test)#converting the output to pandas dataframe

In [1364]: test.head(1)
```

Out[1364]:

	Dependents	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	LoanAmount_log	Total_Income	Total_Income_log	
0	0.0	6756.756757	0	33.783784	60.0	1.0	3.519981	6756.756757	8.818298	563.0

1 rows × 22 columns

```
In [1365]: df.head(1)
```

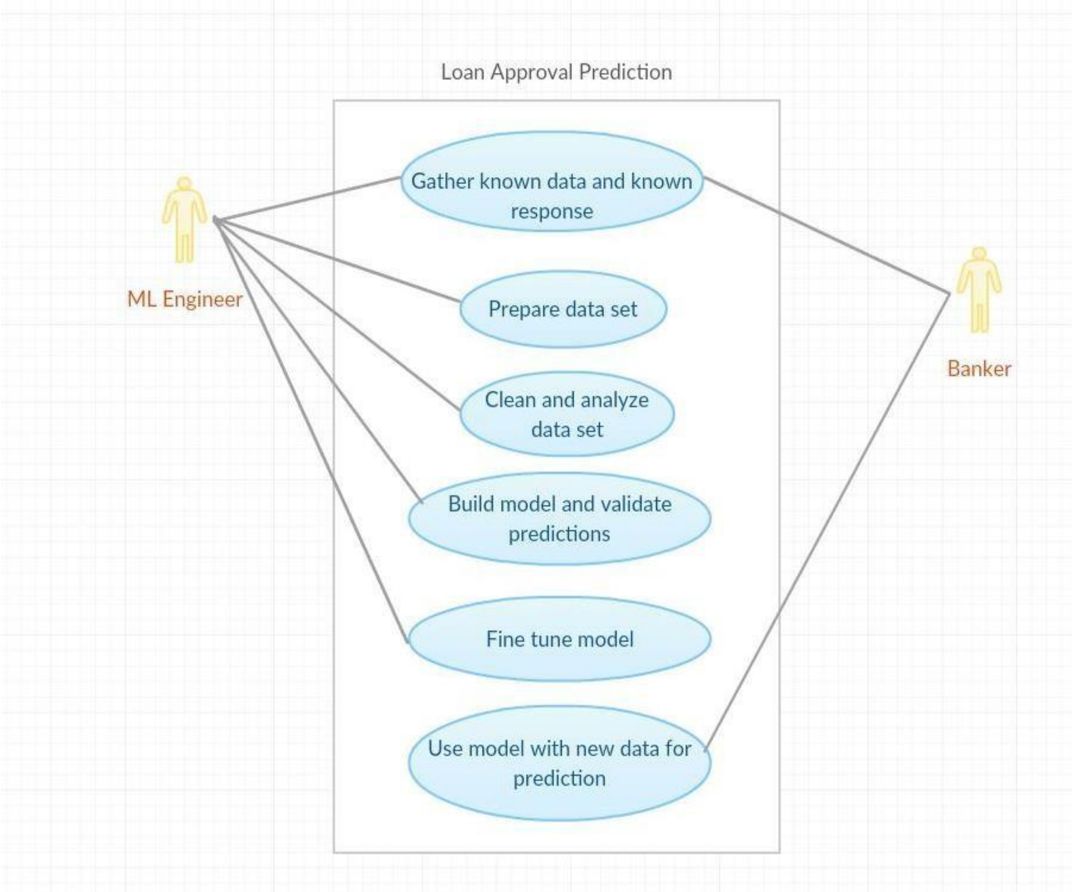
Out[1365]:

	0
0	1

Results and Disscussion

We observe that Logistic Regression score is the highest and hencewe will predict our Test data using Logistic Regression

Use Case Diagram



CONCLUSION and FUTURE SCOPE

Conclusion

1. Out of all the classification algorithms used on the dataset, **Logistic Regression** algorithm gives the best overall prediction accuracy.
2. **Credit History, Balance Income, EMI, Property Area** are the most important factors for predicting the class of the loan applicant (whether the applicant would be‘approved’ or ‘not’).

3. In near future this module of prediction can be integrated with the module of automated processing system. The system is trained on old training dataset, in future software can be made such that new testing data can be used after certain time

4. We can train the XGBoost model using grid search to optimize its hyper parameters and improve the accuracy.

Future Scope

- Time Series Analysis can be done using the Loan data of several years, for prediction of the approximate time, when the client can default.
- Future analysis can be done on predicting the approximate Interest rates that the loan applicant is expected to be charged as per his profile, if his loan is approved. This can be useful for loan applicants, since some banks approve loans, but give very high interest rates to the customer.
- An app with proper UI can be built, which will take various inputs from the user like name, address, loan amount, loan duration, etc. and give a prediction of whether their loan application can be approved by the banks or not based on their inputs along with an approximate interest rates.

BIBLIOGRAPHY

1. Machine Learning Courses

1.2 Coursera(Andrew NG)

1.3 YouTube(SimpliLearn, Edureka)

2. Machine Learning Repositories

2.2 UCI

2.3 Kaggle

3. Websites

3.2 www.kdnuggets.com

3.3 www.analyticsvidhya.com

3.4 www.machinelearningmastery.com