
globsyn
Taking People To The Next Level

Prediction Loan Application Status

By

SMITA BANDYOPADHYAY,INSTITUTE OF ENGINEERING & MANAGEMENT,161040110169,10400116078

Table of Contents

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty (Arnab Chakroborty) for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him/her time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

SMITA BANDYOPADHYAY

Project Objective

PROBLEM:

We have the loan application information like the applicant's name, personal details, financial information and requested loan amount and related details and the outcome (whether the application was approved or rejected). Based on this we are going to train a model and predict if a loan will get approved or not.

OBJECTIVE:

1. Our objective from the project is to make use of pandas, matplotlib, & seaborn libraries from python to extract insights from the data & scikit-learn libraries for machine learning.

2 .Secondly, to learn how to hypertune the parameters using grid search cross validation for the machine learning model.

3. In the end, to predict whether the loan applicant can replay the loan or not using voting ensembling techniques of combining the predictions from multiple machine learning algorithms.

HOW TO PLANNING :

The first thing we need to do and before jumping to analyze the data is to understand the problem statement and create a objective. The next step is to identify our independent variables and our dependent variable.

Now it’s the time to make the next big step in our analysis which is splitting the data into training and test sets.

A training set is the subset of the data that we use to train our models but the test set is a random subset of the data which are derived from the training set. We will use the test set to validate our models as un-foreseen data.

In a sparse data like ours, it’s easy to overfit the data. Overfit in simple terms means that the model will learn the training set that it won’t be able to handle most of the cases it has never seen before. Therefore, we are going to score the data using our test set. Once we split the data, we will treat the testing set like it no longer exists.

Project Scope

- Loan default will cause huge loss for the banks, so they pay much attention on this issue and apply various method to detect and predict default behaviors of their customers.
- All the banks are trying to figure out effective business strategies to persuade customers to apply their loans.
- To prevent this situation, banks have to use many methods to predict their customer's behaviors. Machine learning algorithms have a pretty good performance on this purpose.

Data description

The training data set is now supplied to machine learning model, on the basis of this data set the model is trained. Every new applicant details filled at the time of application form acts as a test data set.

Variable Name	Description	Type
Application_ID	Unique Loan Id	Integer
Gender	Male/Female	Character
Maritial_Status	Applicant married(Y/N)	Character
Dependents	Number of dependents	Integer
Education_Qualification	Graduate/Under Graduate	String
Self_Employed	Self Employed(Y/N)	Character
Applicant_Income	Applicant income	Integer
Co_Applicant_Income	Coapplicant income	Integer
Loan_Amount	Loan amount in thousands	Integer
Loan_Amount_Term	Term of loan in months	Integer
Credit_History	Credit history meets guidelines	Integer
Property_Area	Urban/semi urban/Rural	String
Loan_Status	Loan Approved	Character

Model Building

Naive Bayes Classifier:

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Logistic Regression :

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables.

Decision Trees:

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

Nearest Neighbour:

The k-nearest-neighbours algorithm is a classification algorithm, and it is supervised: it takes a bunch of labelled points and uses them to learn how to label other points. To label a new point, it looks at the labelled points closest to that new point (those are its nearest neighbours), and has those neighbours vote, so whichever label the most of the neighbours have is the label for the new point (the "k" is the number of neighbours it checks).

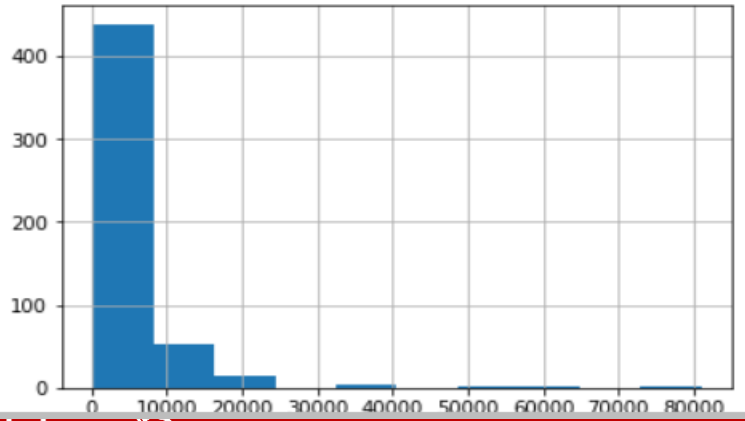
CODE

```
# data head for test
data1.head()
```

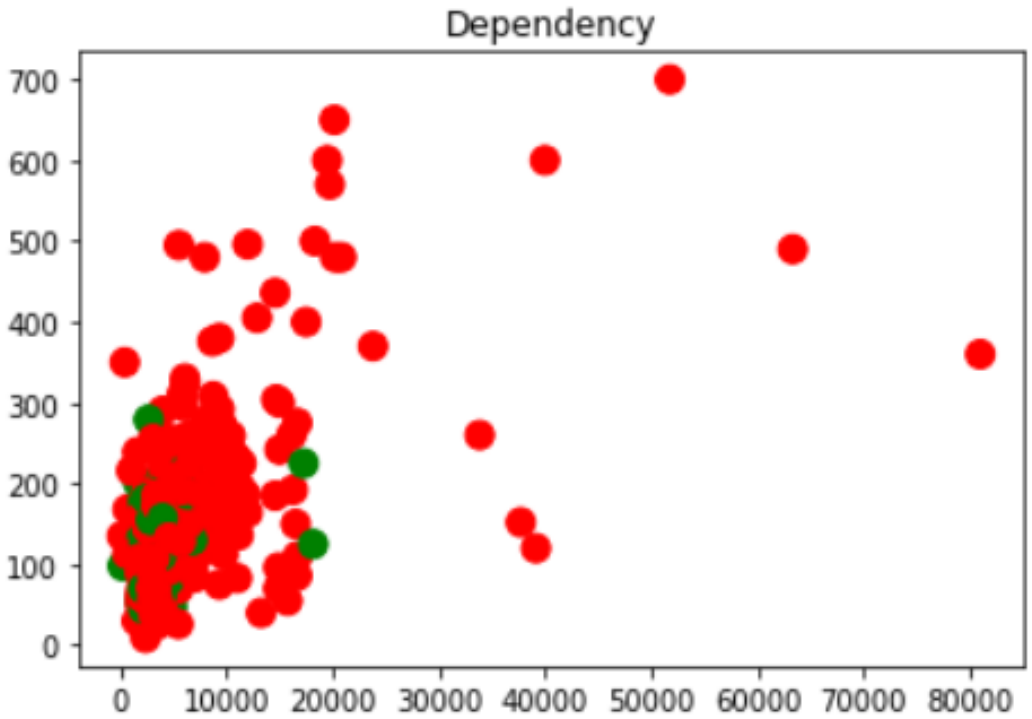
	Application_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	1002	M	No	0	Graduate	No	5849	0	NaN	360.0	1.0	Urban	Y
1	1003	M	Yes	1	Graduate	No	4583	1508	128.0	360.0	1.0	Rural	N
2	1005	M	Yes	0	Graduate	Yes	3000	0	66.0	360.0	1.0	Urban	Y
3	1006	M	Yes	0	Not Graduate	No	2583	2358	120.0	360.0	1.0	Urban	Y
4	1008	M	No	0	Graduate	No	6000	0	141.0	360.0	1.0	Urban	Y

```
# Box Plot for understanding the distributions and to observe the outliers.
%matplotlib inline
# Histogram of variable ApplicantIncome
data['ApplicantIncome'].hist()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c307d2bb70>
```

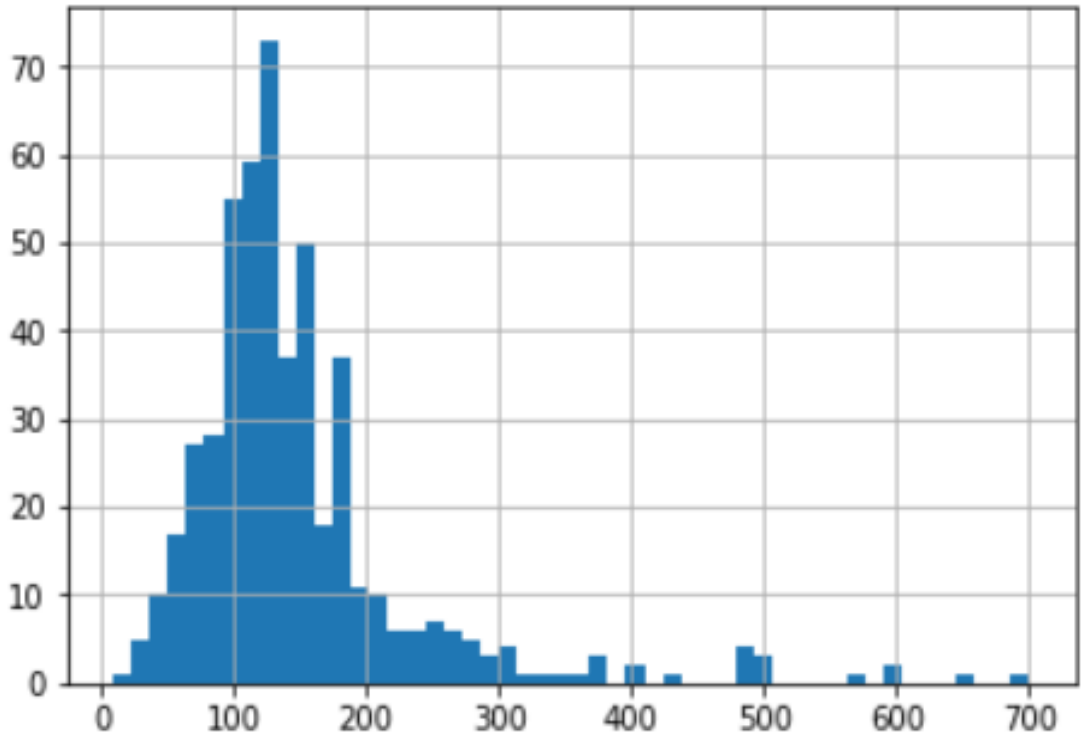


```
colors = colors_edu =  {'Graduate':'r','Not Graduate':'g'}
plt.scatter(data['ApplicantIncome'],data['LoanAmount'],c=data['Education'].apply(lambda x:colors[x]),s=100)
plt.title('Dependency')
plt.legend
plt.show()
```



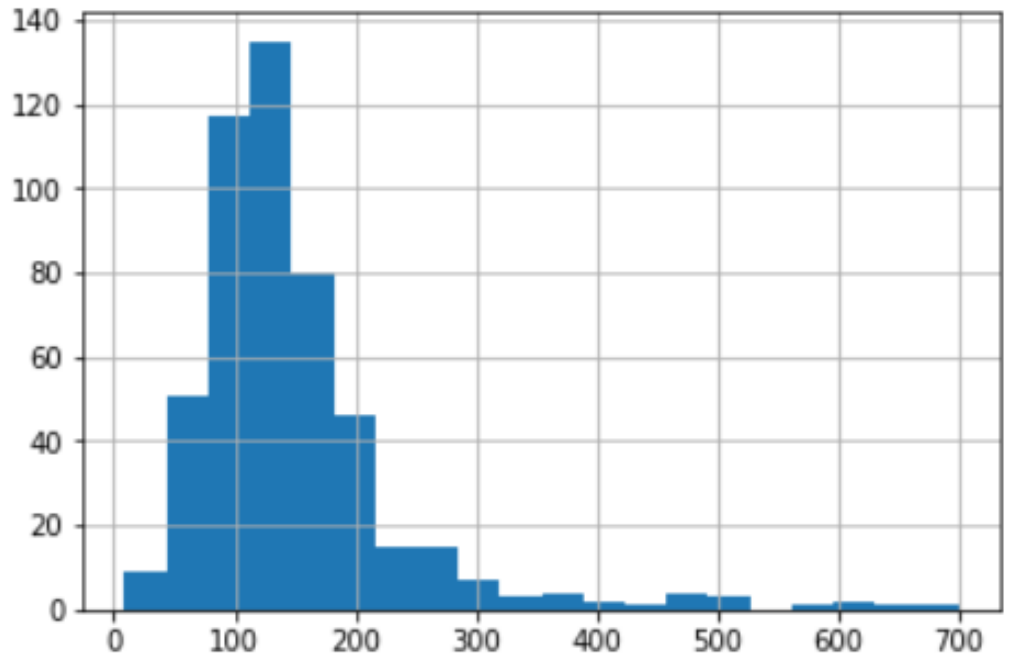
```
# Histogram of variable LoanAmount
data['LoanAmount'].hist(bins=50)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c307b66780>
```



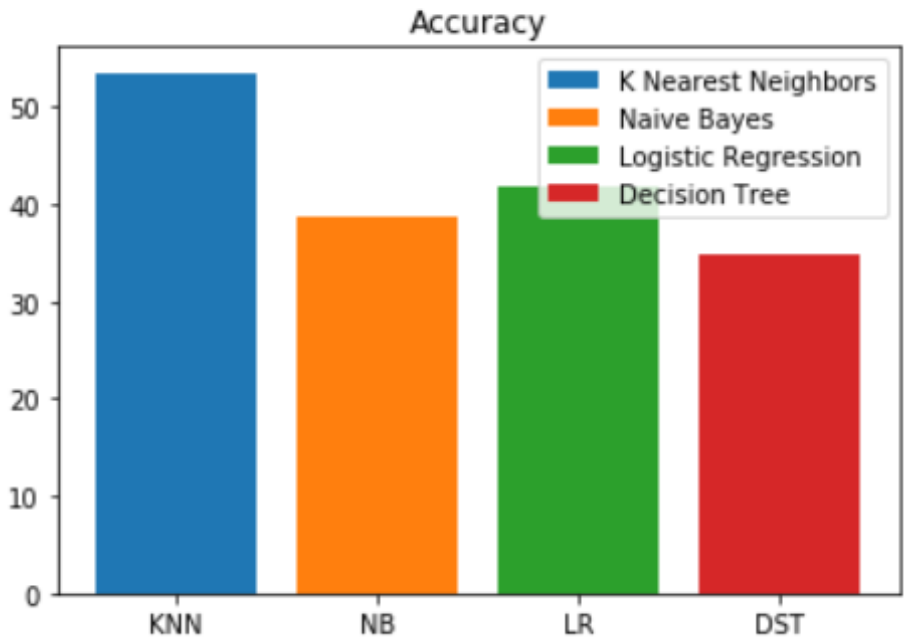
```
# Add both ApplicantIncome and CoapplicantIncome to TotalIncome
data['TotalIncome'] = data['ApplicantIncome'] + data['CoapplicantIncome']

# Looking at the distribtion of TotalIncome
data['LoanAmount'].hist(bins=20)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c307d2bcc0>
```



```
# Box plotting for the Accuracy of all algorithms
plt.bar('KNN',[pre],label='K Nearest Neighbors')
plt.bar('NB',[pred],label='Naive Bayes')
plt.bar('LR',[Lf],label='Logistic Regression')
plt.bar('DST',[Df],label='Decision Tree')
plt.title("Accuracy")
plt.legend()
plt.show
```

```
<function matplotlib.pyplot.show(*args, **kw)>
```



Future Scope of Improvements

- We will understand the various regression, classification and other machine learning algorithms and we'll come to know when to use them.
- We can combine multiple models with by boosting or stacking.
- We can communicate more visually and effectively with matplotlib and seaborn.
- We will use more features to make it more effective.

THANK YOU…