# 

# LOAN APPLICATION STATUS PREDICTION

Submitted by:

Poojashree D S

# 

# Problem Definition

# The aim of the project is to predict the status of the loan application. Company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application forms. 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.

The purpose of this study is to provide comprehensive research and to develop a model to predict loan defaults. This kind of model becomes inevitable as the issue of bad loans are very much critical in the financial sector especially in micro financing banks of various underdeveloped and developed countries. To cope up with this problem a comprehensive literature review was done to study the significant factors that lead to this issue. Moreover, these reviewed studies were critically focused towards applying data mining techniques for the prediction and classification of the loan defaults.

###### 

# This research investigates briefly about techniques involved in machine learning algorithms for acquiring accurate prediction levels of loan application status.

# The major contribution of the current study is to present a model of status prediction that helps companies to automate the loan eligibility process based on customer detail provided.

**Data Analysis**

We will go step by step for building a machine learning algorithm for the prediction of loan defaulters based on certain variables present in the dataset. Our main goal is to correctly identify defaulter's (True positives) so that the lending club can decide whether a person is fit for sanctioning a loan or not in the future.

This dataset contains a total of 614 applicants and 13 attributes, coming from credit history, loan amount, their income, dependents etc. Out of 614 applicants 422 applicants have been granted the loan, which demonstrates the data set is slightly unbalanced. The target variable for assessment is ‘Loan\_status’ .

Independent Variables:

* Loan\_ID
* Gender
* Married
* Dependents
* Education
* Self\_Employed
* ApplicantIncome
* CoapplicantIncome
* Loan\_Amount
* Loan\_Amount\_Term
* Credit History
* Property\_Area
* The main goal is to develop a machine learning model to predict the status of the loan application. I will use mainly Python, Pandas, and Scikit-Learn libraries for this implementation.

****

From the above graph we can see that the dataset is slightly unbalanced, about 2/3rd of applicants have been granted the loan. The attributes explains as follows

Loan ID -> As the name suggests each person should have a unique loan ID.

Gender -> In general it is male or female. No offence for not including the third gender.

Married -> Applicant who is married is represented by Y and not married is represented as N. The information regarding whether the applicant who is married is divorced or not has not been provided. So we don’t need to worry about all these.

Dependents -> the number of people dependent on the applicant who has taken loan has been provided.

Education -> It is either non -graduate or graduate. The assumption I can make is “ The probability of clearing the loan amount would be higher if the applicant is a graduate”.

Self\_Employed -> As the name suggests Self Employed means , he/she is employed for himself/herself only. So freelancers or having their own business might come in this category. An applicant who is self employed is represented by Y and the one who is not is represented by N.

Applicant Income -> Applicant Income suggests the income by Applicant.So the general assumption that i can make would be “The one who earns more have a high probability of clearing loan amount and would be highly eligible for loan ”

Co Applicant income -> this represents the income of a co-applicant. I can also assume that “ If co applicant income is higher , the probability of being eligible would be higher “

Loan Amount -> This amount represents the loan amount in thousands. One assumption I can make is that “ If Loan amount is higher , the probability of repaying would be lesser and vice versa”

Loan\_Amount\_Term -> This represents the number of months required to repay the loan.

Credit\_History -> When I googled it , I got this information. A credit history is a record of a borrower’s responsible repayment of debts. It suggests → 1 denotes that the credit history is good and 0 otherwise.

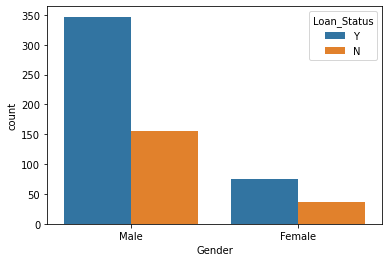
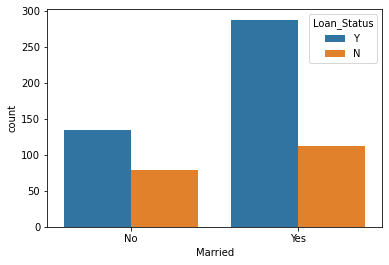
Property\_Area -> The area where they belong to is my general assumption as nothing more is told. Here it can be three types. Urban or Semi Urban or Rural

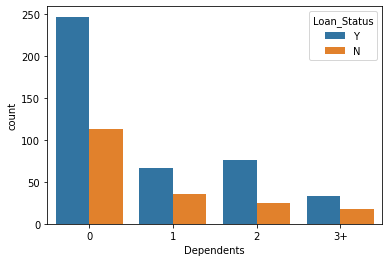
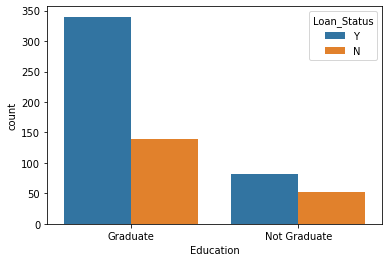
Loan\_Status -> If the applicant is eligible for loan it’s yes represented by Y else it’s not represented by N.

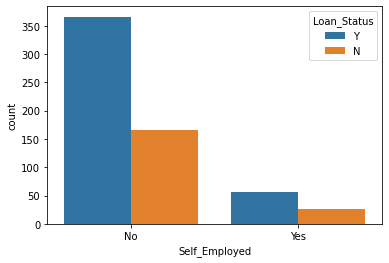
**Exploratory Data Analysis**

As the aim of this experiment is to **identify patterns** that can predict loan\_status, I will be focusing mainly on the Loan\_Staus portion of the dataset for the exploratory analysis.

Loan\_Status is the target variable. The below charts represent how much Loan\_Status depends on the feature variables.

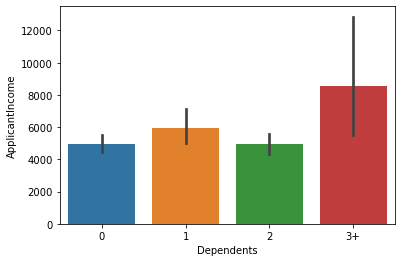
 



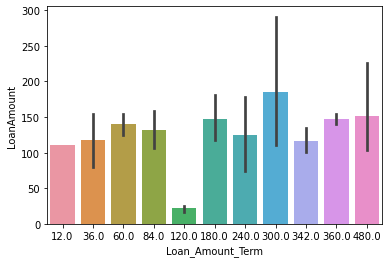
* More applicant Loan\_status is Yes
* More Men applicant present Then women
* More married applicant are present in the Dataset and married applicant are more likely to get approval of loan than unmarried
* Majority of applicants are zero dependents and are more likely to get loan approval.
* 5/6th of applicants are graduates and graduates have a higher proportion of loan approval.
* 5/6th applicants are not self employed
* Self employed applicants are more likely to get loan approval.



The above graph shows the Loan \_Status of applicant Based on his monthly income, if applicant monthly income is more than 5000 he is more likely to get a loan.



If applicants have more number of dependents there is a possibility of having more income which increases the chance of applicants to get a loan.



Loan amount of the applicant is dependent on the Loan\_Amount\_Term,in this case if the term is 300 applicants get a larger amount of loan.

**Pre-processing**

Data Preprocessing includes data cleaning in which we manually remove unwanted columns. In this particular project we remove customerID as it is not required.

First we have import pandas to load dataset which is present in the csv format , the following code is used to read the file

df = pd.read\_csv('loan\_prediction.csv')

The dataset consists of null values. I replaced the null values with their mean values and their mode . Mean is nothing but the average value whereas median is nothing but the central value and mode the most occurring value. Replacing the categorical variable by mode makes some sense.

The skewness in the data is removed using the log transformation.

Data Preprocessing includes data cleaning in which we manually remove unwanted columns. In this particular project we remove customerID as it is not required.

For handling categorical variables, there are many methods like One Hot Encoding or Dummies. In one hot encoding method we can specify which categorical data needs to be converted . However as in my case, as I need to convert every categorical variable in to numerical, I have used get\_dummies method

Then I remove columns which are giving the same information.

### Split train and test data

The dataset was split into 70% for training and 30% for testing. The training set will be used to generate the model the chosen algorithms will use when exposed to new data**.** The test set is the final dataset I will touch to measuremodel performance based on some metrics.To split,train and test data i imported train test split.

**from** **sklearn.model\_selection** **import** train\_test\_split

train\_test\_split will split the data into 70% for training and 30% for testing.

**Scaling of the data**

Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units.I used the standard scalar to scale the data, then I started to build the model by splitting the model to train and test.

**Balancing the Data**

The data is highly imbalanced.it can reduce the accuracy score or model may give biased output so to avoid this we have to balance the data.

I use SMOTE to balance the data. It increases the number of cases to balance the data so the model is not biased.

As the data cleaning and data structuring are done, we will be going to our next section which is nothing but Model Building.

**Building Machine Learning Models**

For performance assessment of the chosen models, various metrics are used. I used accuracy score, confusion matrix and classification report.

· Accuracy score is the ratio of number of correct predictions to the total number of input samples.

· A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

The classification report visualizer displays the precision, recall, F1, and support scores for the model.

In this experiment, I applied eight different ML algorithms to analyze and compare the Accuracy score obtained by each of them. Those are listed below:

* LogisticRegression
* DecisionTreeClassifier
* SVC
* RandomForestClassifier
* KNeighborsClassifier
* AdaBoostClassifier
* GradientBoostingClassifier

I have tried all the above models but RandomForest give best Accuracy score

**from** **sklearn.ensemble** **import** RandomForestClassifier

rf = RandomForestClassifier()

rf.fit(x\_train,y\_train)

predr = rf.predict(x\_test)

print('accuracy score :',)

print(accuracy\_score(y\_test,predr))

print(confusion\_matrix(y\_test,predr))

print(classification\_report(y\_test,predr))

accuracy score :

0.7914691943127962

[[82 25]

[19 85]]

precision recall f1-score support

0 0.81 0.77 0.79 107

1 0.77 0.82 0.79 104

accuracy 0.79 211

macro avg 0.79 0.79 0.79 211

weighted avg 0.79 0.79 0.79 211

In the Random forest model True positive values are 82 and True negative values are 82, which shows it is a pretty good model.

## Cross validation:

I applied cross-validation method which is a technique that partitions the data into subsets, training the data on a subset and use the other subset to evaluate the model’s performance.

The top performance is given by the RandomForestClassifier, and the difference between the Accuracy Score and cross validation is very less so this is not due to oversampling or undersampling. But there is still room for optimization.

scr = cross\_val\_score(rf,x,y,cv=5)

print('coss validation score is',scr.mean())

coss validation score is 0.8176105945336716

**Hyper Parameter Tuning**

To improve overall performance I tuned classifiers parameters using GridSearchCV for RandomForestClassifier,I used parameters max\_featues and n\_estimators to tune the model.

The model Accuracy score shifted from 0.79 to 0.81.

fimal\_mod =RandomForestClassifier(n\_estimators=500,max\_features='sqrt')

fimal\_mod.fit(x\_train,y\_train)

predict= fimal\_mod.predict(x\_test)

accu = accuracy\_score(y\_test,predict)

print(accu)

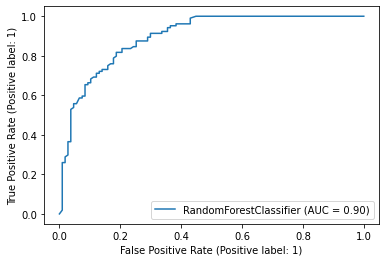
0.8104265402843602

This means the implementing ML model based on RandomForestClassifier delivers 81% of accuracy while predicting Loan\_status.

**Saving the model**

I saved the model for future use in joblib as ‘loanrf1.pkl’ and loaded the saved model and saved the prediction values in the csv file.

**AUC-ROC Curve for the Model**

****

The AUC ROC score for this model is 0.791

# 

# 

# Conclusion

The aim of the project is building and comparing several applicant loan application status.

In EDA we remove all the null values present and replace them with the mean and mode of the column

Using the visualization we are able to conclude the relation between the data

Finally able to build the model and cross validate it, balances it and hyper tuned the model it gives better accuracy score

And lastly saved the model in joblib

No algorithm will predict churn with 100% accuracy. There will always be a trade-off between precision and recall. That's why it's important to test and understand the **strengths and weaknesses**of each classifier and get the best out of each.

This model helps companies to automate the loan eligibility process based on customer detail provided.