EDA LOAN STATUS PREDICTION

Using Classification Algorithms

Developed by:Sharanya Poshala,Sangeetha Yalla,Manusha Gundu,Shivani Namani

Smart Bridge-Remote Summer Internship Program

1. INTRODUCTION:

In finance, a loan is the lending of money by one or more individuals, organizations, or other entities to other individuals, organizations etc. The recipient (i.e., the borrower) incurs a debt and is usually liable to pay interest on that debt until it is repaid as well as to repay the principal amount borrowed.

The document evidencing the debt (e.g., a promissory note) will normally specify, among other things, the principal amount of money borrowed, the interest rate the lender is charging, and the date of repayment. A loan entails the reallocation of the subject asset for a period of time, between the lender and the borrower.

In India, the number of people applying for the loans gets increased for various reasons in recent years. The bank employees are not able to analyse or predict whether the customer can payback the amount or not (good customer or bad customer) for the given interest rate. The aim of this paper is to find the nature of the client applying for the personal loan. An exploratory data analysis technique is used to deal with this problem.

The result of the analysis shows that short term loans are preferred by majority of the clients and the clients majorly apply loans for debt consolidation. The results are shown in graphs that helps the bankers to understand the client's behavior.

It is difficul for bank employes to collect the data manually .Here to make the process simple & easy . The bankers can get the data with in few minutes. Whether the person is eligible or Not for loan application. Customer first apply for home loan after that company validates the customer eligibility for loan. However doing this manually takes a lot of time. Hence it wants to automate the loan eligibility process (real time) based on customer information So the final thing is to identify the factors/ customer segments that are eligible for taking loan. How will the bank benefit if we give the customer segments is the immediate question that arises?

The solution is, banks would give loans to only those customers that are eligible so that they can be assured of getting the money back.

Hence the more accurate we are in predicting the eligible customers the more beneficial it would be for the banks.

1.1 Overview

In present generation everyone of us have bank accounts ,we can rarely find any one without an account.In lifetime we may atleast take one loan from the bank for any of the need. Due to lot of transactions have been occuring every day huge data volumes are available which represent the customers behavior and the risks around loan are increased.

Data Mining is one of the most motivating and vital area of research with the aim of extracting information from tremendous amount of accumulated data sets. Here a new model for classifying loan risk in banking sector by using Machine Learning concepts. The model has been built using data form banking sector to predict the status of loans. Six algorithms have been used to build the proposed model:RandomForest, Logistic Regression, Decision Tree, SVM, KNN ,NaiveBayes. By using the algorithm a Flask model has been implemented and tested. The results has been discussed and a full comparison between algorithms was conducted. NaiveBayes was selected as best algorithm based on accuracy.

1.2 Purpose

Why are we doing EDA?

Why are we doing all these ? Why can't we do directly modeling the data instead of knowing all these….." Well in some cases we can easily come to conclusion if we just to do EDA. Then there is no necessary for going through next models

Now a days banks are highly associated with risks.we hav seen vijay malya,nirav modi etc who took loan from several banks and flew away without paying back this will result in huge problem to the banks.

So our aim from the project is to develop an application which helps in prediction of loan

status fastly,accurately and in user friendly environment

we used pandas, matplotlib, & seaborn libraries from python to extract the libraries for machine learning for the loan prediction.

2. LITERATURE SURVEY

Data mining is the process of analyzing data from different perspectives and extracting useful knowledge from it. It is the core of knowledge discovery process. The various steps involved in extracting knowledge from raw data . Different data mining techniques include classification, clustering, association rule mining, prediction and sequential patterns, neural networks, regression etc.

Classification is the most commonly applied data mining technique, which employs a set of preclassified examples to develop a model that can classify the population of records at large. Fraud detection and credit risk applications are particularly well suited to classification technique. In classification, a training set is used to build the model as the classifier which can classify the data items into its appropriate classes.A test set is used to validate the model.

2.1 Existing Problem

The previous models have high time complexity and space complexity whereas this model is constrained with the lot of advantages and with a higher accuracy than any other model already proposed. In this model we used Machine learning algorithm named Random Forest which give an accuracy more then the previously predicted problem and there is an user friendly user interface to check loan status. And lot of the previous models haven't included the UI (User interface) which is so friendly and convenient for the users. There are some of the models that are referred for the closure below:

Glorfeld and Hardgrave (2001) proposed a complete and useful systematic way to produce an optimal design of a high performance model for neural network estimating of the Credit value related to applications of commercial loan. The neural network constructed using their design was able to classifying 75% of loan applicants correctly.

Michael D. Johnson, Anders Gustafsson studied broadly the usages of data mining techniques in banking sectors and its related impacts on several operations. They built a prediction model to predict if the customers will pay back the loans which they borrow using the techniques of neural network and classification. Jagielska et al

investigated the abilities of neural networks in classifying loans risk, uncertain logic genetic algorithms, rule stimulation software, he concluded that the genetic approach is more favorably than the neuron fuzzy and rough set methods.

2.2 Proposed Solution

Machine Learning (Random Forest):

Using a Decision Tree Regressor has improved our performance, we can further improve the performance by ensembling more trees. Random Forest Regressor trains randomly initialized trees with random subsets of data sampled from the training data, this will make our model more robust.

Random forests are an ensemble model of machine learning with their roots in Decision Trees.These decision trees individually may over fit the data set and thus they come together to form a much stronger model.Numerous decision trees are first built and based on these by performing random sampling of the attributes,a group of decision trees are assembled to form a Random Forest.

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees.After preparing the data, we can fit different models on the training data and compare their performance to choose the algorithm with good performance. As this is a regression problem, we can use RMSE (Root Mean Square Error) and $R²$ score as evaluation metrics.

And also we have created an UI using the Flask for the loan status prediction, this UI will allow the users to predict the loan status very easily and the User interface is user friendly not at least one complication in using the interface, and it can be used just by entering some necessary details into the UI in real time it'll give the predicted value like if the customer is beneficial to take a loan and how often does he pays the loan interest amount to the bank.

Basically this model will give the predicted value when a customer with details will pay the loan back to bank, by just taking some necessary details of the customer in real time, and those details will be collected by bank employee within minutes.

3. THEORETICAL ANALYSIS

While selecting the algorithm that gives an accurate prediction we gone through lot of algorithms which gives the results abruptly accurate and from them we selected only one algorithm for the prediction problem that is RandomForest Regression.Random forest is a Supervised Learning algorithm which uses ensemble learning method for classification and regression.

Random forest is a bagging technique and not a boosting technique. The trees in random forests are run in parallel. There is no interaction between these trees while building the trees.It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

The peculiarity of this problem is collecting the customers details real time and working with the prediction at the same time, so we developed an user interface for the people who'll be accesssing for the loan status prediction.

Accuracy is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test data set. The formula is as follows

Accuracy=TP+TN/TP+TN+FT+FN

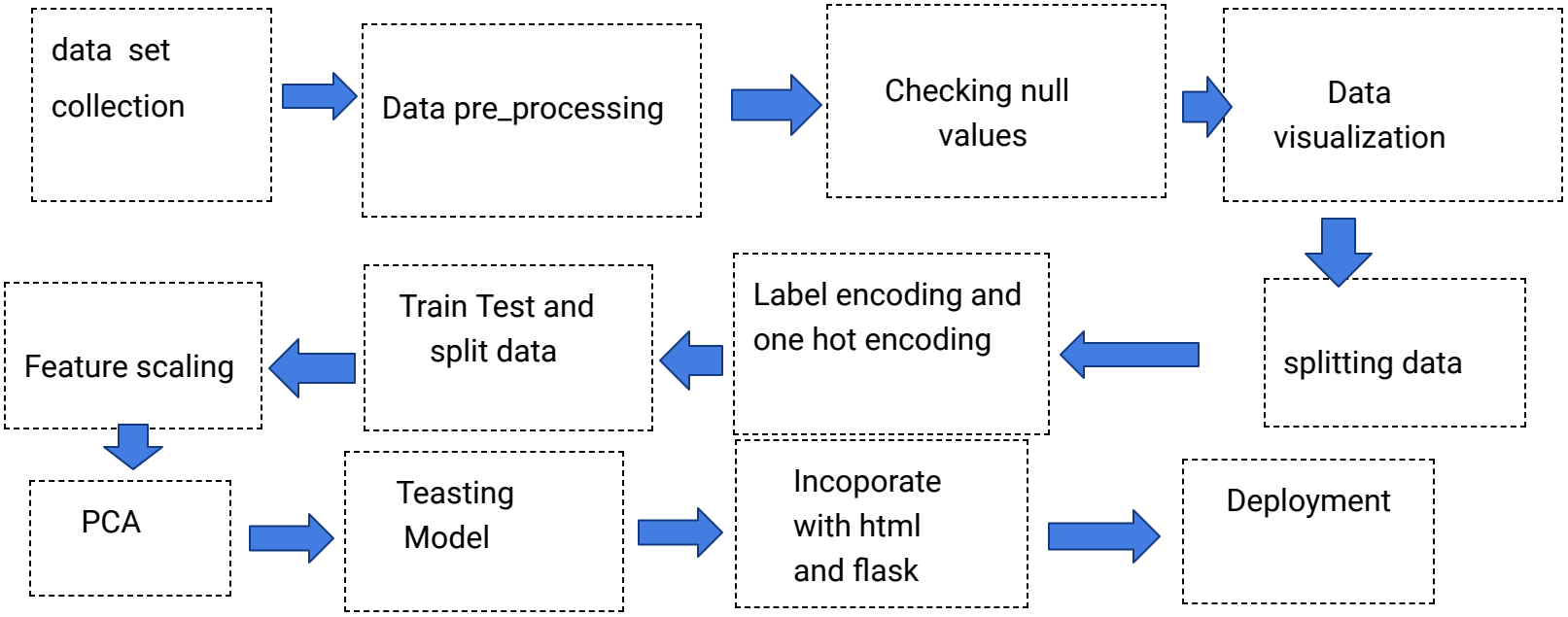
At first we got like lot of worst accuracies because we tried lot of algorithms for the best accurate algorithm , finally after all of that we tried the best suitable algorithm which gives the prediction accurately is Random Forest Classifier and developed it to use as a real time prediction probelm for the loan status prediction.

In statistics, a receiver operating characteristic (ROC), is a two dimensional graphical plot that illustrates the performance of a binary classifier system. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. ROC curve can intuitively represent the performance of classifier.

FPR=FP/FP+TN

TPR=TP/TP+FP

3.1 Block Diagram:



3.2 SOFTWARE DESIGNING

- Jupyter Notebook Environment
- Spyder Ide
- Machine Learning Algorithms
- Python (pandas, numpy, matplotlib, seaborn, sklearn)
- HTML
- Flask

We developed this loan status prediction by using the Python language which is a interpreted and high level programming language and usng the Machine Learning algorithms.

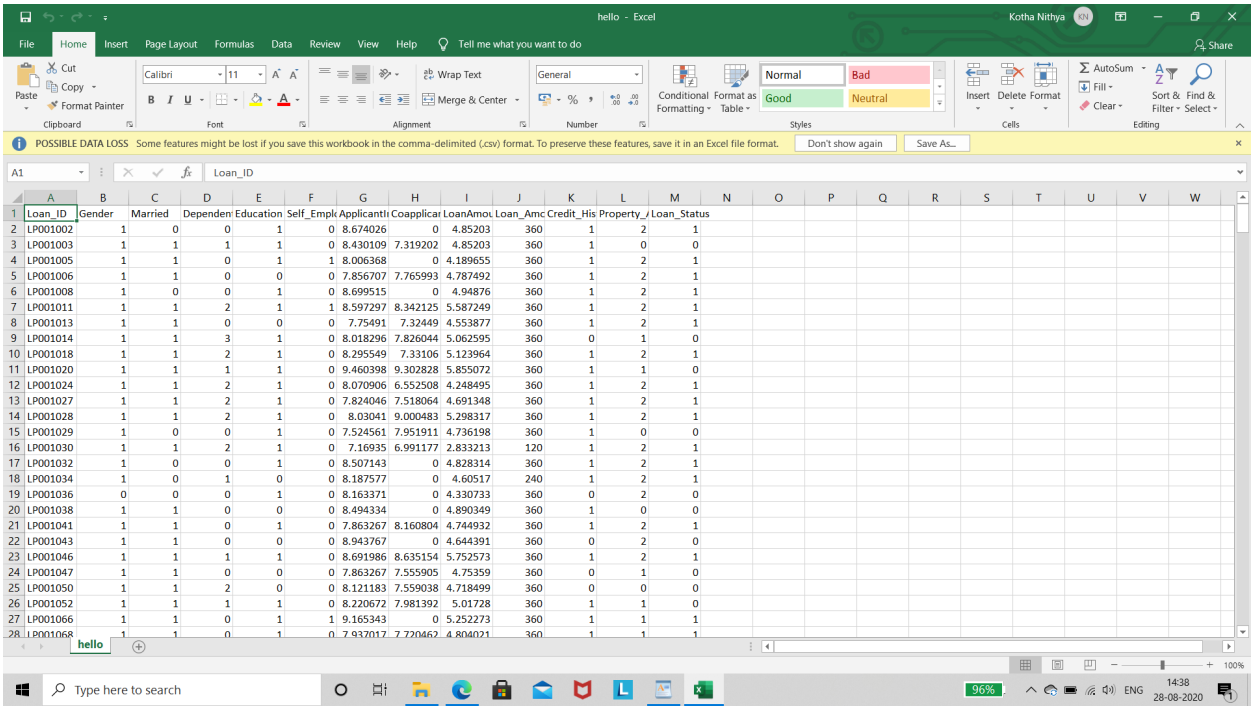
For coding we used the Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language.

For creating an user interface for the prediction we used the Flask. It is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a webpage is HTML by creating the templates to use in th functions of the Flask and HTML.

4. EXPERIMENTAL INVESTIGATION

In this paper, the dataset we used is derived from kaggle website

It contains 615 original loan data of users with 13 attributes. After that, the missing values are filled in by means of mode,mean median interpolation, and the duplicate or meaningless attributes are deleted, finally we have retained to 13 attributes.



Description about the Data Columns:

It's very useful to know about the data columns before getting in to the actual problem for avoiding confusion at a later state. Now let us understand the data columns first so that we will get a glance.

There are altogether 13 columns in our data set. Of them Loan_Status is the response variable and rest all are the variables /factors that decide the approval of the loan or not.

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 regarding 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 freelancer or having a 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 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 no represented by N.

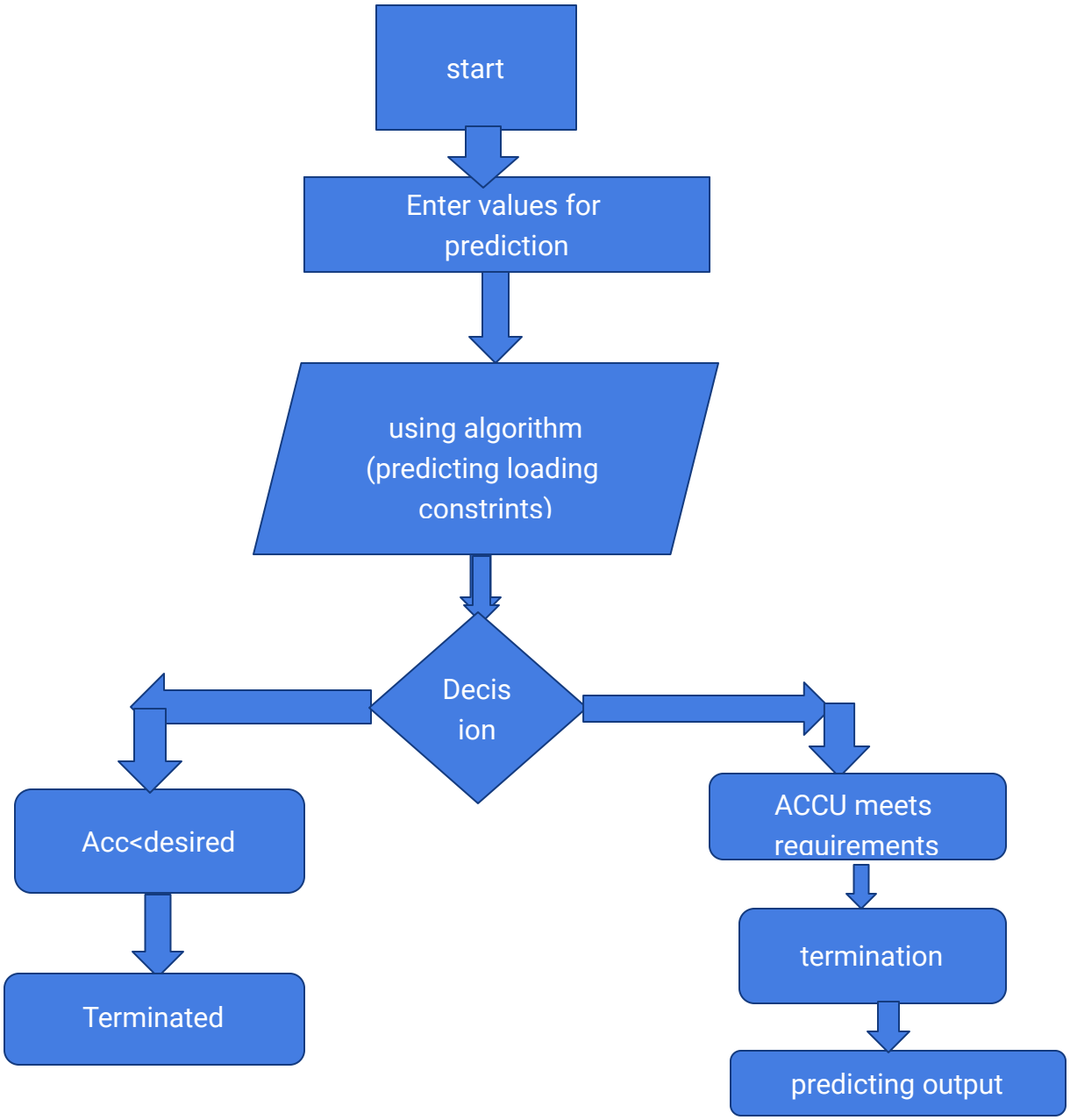
Exploratory Data Analysis

By looking at the columns description in the above paragraph, we can make many assumptions like:

- The one whose salary is more can have a greater chance of loan approval.
- The one who is graduate has a better chance of loan approval.
- Married people would have a upper hand than unmarried people for loan approval .

- The applicant who has less number of dependents have a high probability for loan approval.
- The lesser the loan amount the higher the chance for getting loan.

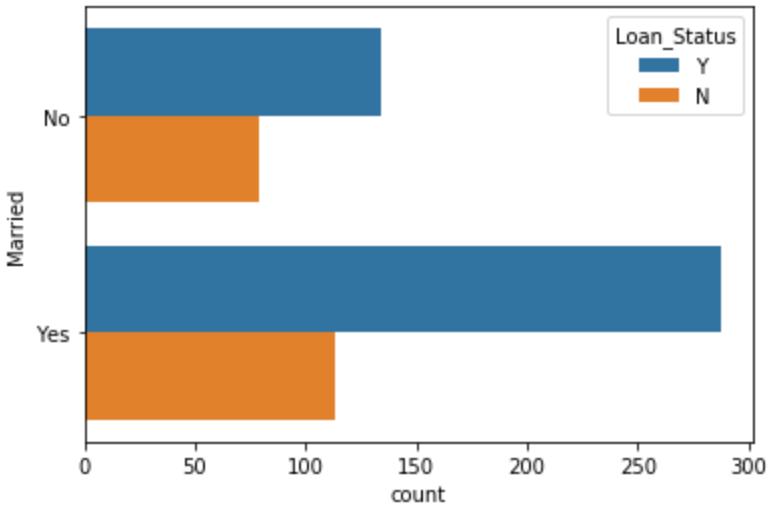
5. FLOWCHART



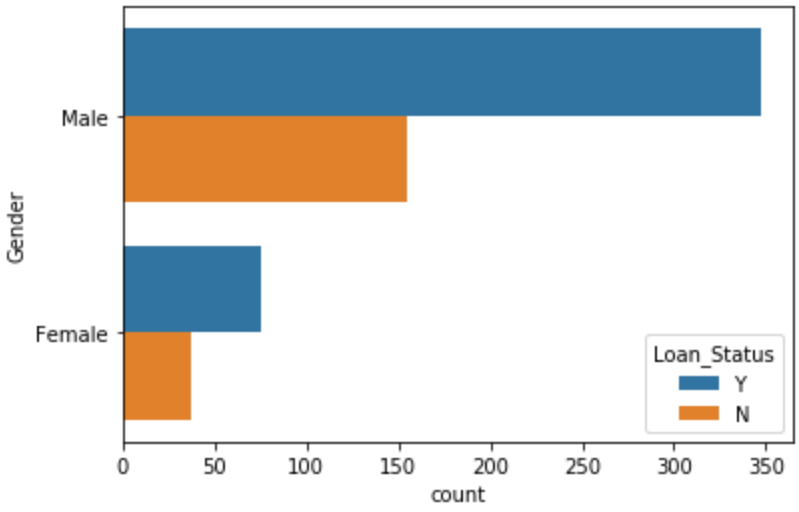
6. RESULT

In this paper, the Random Forest algorithm is used to predict its performance, and compared with another six machine learning methods namely the decision tree, the logistic regression, KNN, Naive Bayes and the SVM. The obtained results are displayed in Table below. The results show that, the performance of Random Forest is higher than that of Logistic Regression, Naive Bayes, SVM and decision tree,KNN .The ROC curve of the prediction model based Random Forest are all above 0.72, indicating that the model has strong ability of generalization.

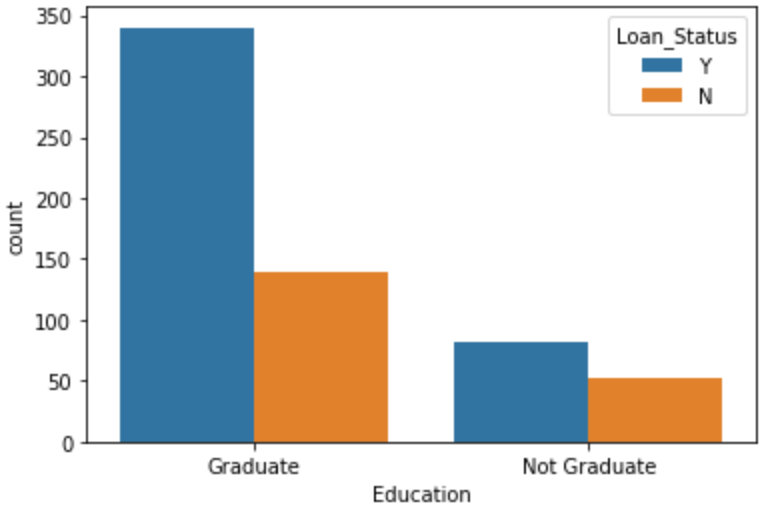
#married people tend to take more loan

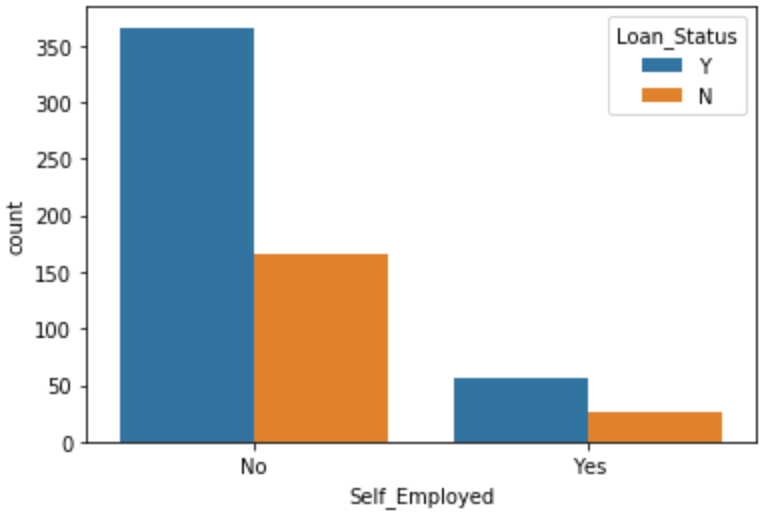


more males are on loan than females.



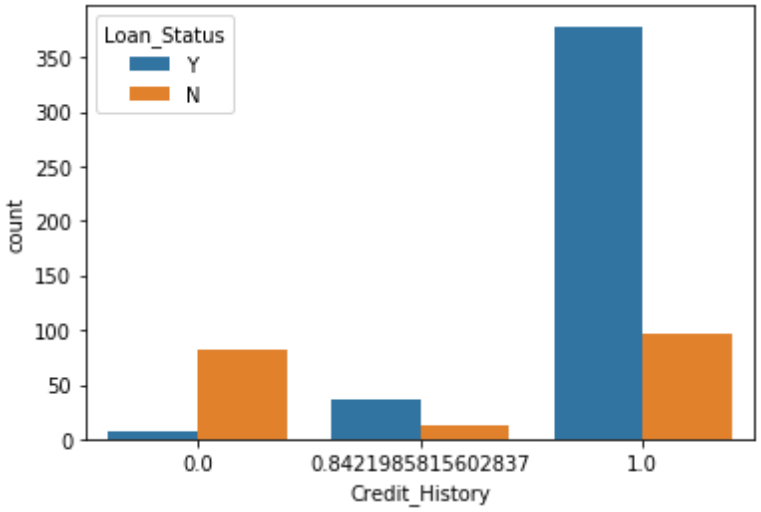
#educated people tend to take loan than uneducated



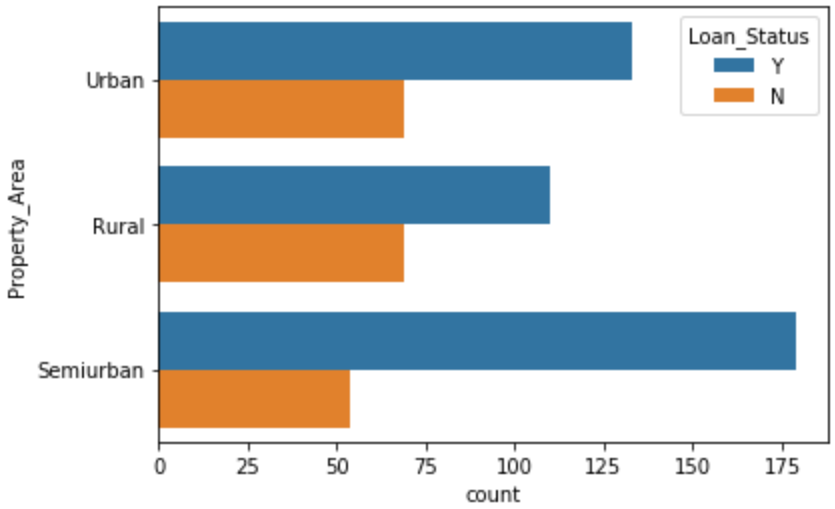


#The category of those that take loans is less of self-employed people.

#That's those are not self-employed probably salalary earners obtain more loan



#According to the credit history, greater number of people pay back their loans.



#urban obtain more loan, folowed by semiUrban and then rural.

sno	algorithm	Accuracy	ROC Curve
1	logistic regression	62%	0.62
2	Decision tree classifier	63%	0.63
3	Random forest classifier	74%	0.72
4	KNN	66%	0.65
5	Naive Bayes	67%	0.65
6	SVM	46%	0.50

7. ADVANTAGES AND DISADVANTAGES

Advantages:

● Easy and simple User Interface for the bank people who is going to evaluate the customer loan status.

● Random Forest give the accurate result of the prediction upto 74% which is the algorithm we used for prediction.

● It is widely used for managing risks in the banking industry.

● It is composed using the HTML and Python for the web usage in real time.

● It can work in real time and predict as soon as the necessary details for prediction are given to the model.

Disadvantages:

● Gives only 74% accuracy for the loan status.

● It could not work anywhere like an web-application, if one is using other should be quiet.

● Needs more than a single value for the prediction.

8. APPLICATIONS

● It is widely used for managing risks in the banking industry. Bank executives need to know the credibility of customers they are dealing with in real time.

● To have an idea of customer relationship cycle such as customer acquisition, increasing

value of the customer and customer retention.

● It is one of the most widely used areas of data mining in the banking industry. The consumer behavior with reference to product, price and distribution channel can be analyzed by the marketing department.

● Due to tremendous growth in data the banking industry deals with, analysis and transformation ofthe data into useful knowledge has become a task beyond human ability.

● So we use Machine Learning Algorithms to analyze the data and propose what banks and loan lending companies need to achieve their needs.

9. CONCLUSION

In this paper, the Random Forest algorithm is adopted to build a UI model for predicting loan status compared with other six algorithms of logistic regression, KNN, Naive Bayes, decision tree and support vector machine. The experiment shows that the Random Forest algorithm performs outstanding than the other six algorithms in the prediction of loan default and has strong ability of generalization. There is no definitive guide of which algorithms to use given any situation. What may work on some data sets

may not necessarily work on others. Therefore, always evaluate methods using cross validation to get a reliable estimates.

10. FUTURE SCOPE

In future the Random Forest algorithm can be applied on other data sets available for loan approvals to further investigate its accuracy. A rigorous analysis of other machine learning algorithms other than these six can also be done in future to investigate the power of machine learning algorithms for loan status prediction. In further study, we will try to conduct experiments on larger data sets or try to tune the model so as to achieve the state -of-art performance of the model and a great UI support system making it complete web application model.

11. BIBLIOGRAPHY

- Briceno Ortega,Ana Cecilia and Frances Bell. “Online social lending: borrower generated content [C].”AMCIS 2008 Proceedings, 2008. 380
- Malekipirbazari M , Aksakalli V . Risk assessment in social lending via random forests[J]. Expert Systems with Applications, 2015, 42(10):4621-4631.
- Selvamuthu, D., Kumar, V.& Mishra, A. Financ Innov(2019)5:16 https://doi.org/10.1186/s40854-019- 0131-7.
- Zhong, X. and Enke , D. Financ Innov (2019) 5: 24 https://doi.org/10.1186/s40854-019- 0138-0
- Yao X , Crook J , Andreeva G . Support vector regression for loss given default modelling[J]. European Journal of Operational Research, 2015, 240(2):528-538.
- Emekter, R., Tu, Y., Jirasakuldech, B., Lu, M., 2015. Evaluating credit risk and loan performance in online Peer-to-Peer (P2P) lending. Appl. Econ. 47, 54–70.
- Kvamme,H. et al. (2018), Predicting Mortgage Default Using Convolutional Neural Networks; Expert Systems With Applications, 102, pp.207-217
- K. Aleum and S.B. Cho, “An ensemble semi-supervised learning method for predicting defaults in social lending,” Eng. Appl. Artif. Intell., vol. 81, pp. 193–199, May 2019
- Challa, M.L., Malepati, V. & Kolusu, S.N.R. Financ Innov (2018) 4: 24. https://doi.org/10.1186/s40854-018- 0107-z

APPENDIX

APP.PY

```
from flask import Flask,render_template,request

import pickle

model = pickle.load(open('loan1.pkl','rb'))
app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html')


@app.route('/prediction',methods=['POST'])
def prediction():
    '''    #
    For rendering results on HTML GUI
    '''
    #x_test = [[str(x) for x in request.form.values()]]
    a = request.form['Gender']
    if (a == "Male"):
        a = 1
    if (a == "Female"):
        a = 0

    b = request.form['Married']
    if (b == "Yes"):
        b = 1
    if (b == "No"):
        b = 0

    c = request.form['Dependents']
```

```
d = request.form['Education']
if (d == "educated"):
   d = 0
if (d == "uneducated"):
   d = 1

e = request.form['Self_Employed']
if (e == "yes"):
   e = 1
if (e == "no"):
   e = 0

f = request.form['ApplicantIncome']
g = request.form['CoapplicantIncome']
h = request.form['LoanAmount']
i = request.form['Loan_Amount_Term']

j = request.form['Credit_History']
if (j == "yes"):
   j = 1
if (j == "no"):
   j = 0

k = request.form['Property_Area']
if (k == "urban"):
   k = 2
if (k == "semiurban"):
   k = 1
if (k == "rural"):
   k = 0

total = [[int(a),int(b),int(c),int(d),int(e),float(f),float(g),float(h),float(i),int(j),int(k)]]
prediction_1 = model.predict(total)
print(prediction_1)
output=prediction_1[0]
if(output==1):
   pred=" Congratulations Loan Granted"
```

```
    else:
        pred="Sorry Loan rejected"
    return render_template('index.html', y=str(pred))

'''@app.route('/predict_api',methods=['POST'])
def predict_api():

    For direct API calls trought request

    data = request.get_json(force=True)
    prediction = model.y_predict([np.array(list(data.values()))])

    output = prediction[0]
    return jsonify(output)
    '''

if __name__ == "__main__":
    app.run(debug=True)
```

HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>

  <title>Loan prediction </title>
   <link rel = "icon" href = "{{url_for('static', filename='favicon.png')}}"
        type = "image/x-icon">
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
                                                    <link                    rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css">
                                                    <link                    rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.mi
```

```
n.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></scri
pt>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>

<style>
@import url(https://fonts.googleapis.com/css?family=Source+Sans+Pro:200,300);
body{
   background: #50a3a2;
background: -webkit-linear-gradient(top left, #50a3a2 0%, #53e3a6 100%);
background: -moz-linear-gradient(top left, #50a3a2 0%, #53e3a6 100%);
background: -o-linear-gradient(top left, #50a3a2 0%, #53e3a6 100%);
background: linear-gradient(to bottom right, #50a3a2 0%, #53e3a6 100%);
}
body{
 font-family: 'Source Sans Pro', sans-serif;
 color: black;
 font-weight: 300;
 position: centre;

 ::-webkit-input-placeholder { /* WebKit browsers */
  font-family: 'Source Sans Pro', sans-serif;
    color:    black;
  font-weight: 300;
 }
 :-moz-placeholder { /* Mozilla Firefox 4 to 18 */
  font-family: 'Source Sans Pro', sans-serif;
   color:    black;
   opacity:  1;
   opacity:  1;
  font-weight: 300;
 }
 ::-moz-placeholder { /* Mozilla Firefox 19+ */
```

```
    font-family: 'Source Sans Pro', sans-serif;
     color:    black;
     opacity:  1;
    font-weight: 300;
  }
  :-ms-input-placeholder { /* Internet Explorer 10+ */
    font-family: 'Source Sans Pro', sans-serif;
     color:    black;
    font-weight: 300;
  }
}
.container{
  background: #50a3a2;
background: -webkit-linear-gradient(top left, #50a3a2 0%, #53e3a6 100%);
background: -moz-linear-gradient(top left, #50a3a2 0%, #53e3a6 100%);
background: -o-linear-gradient(top left, #50a3a2 0%, #53e3a6 100%);
background: linear-gradient(to bottom right, #50a3a2 0%, #53e3a6 100%);
}
.container{
  box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);
  max-width: 340px;
  margin: 0 auto;
  margin-top: 50px;
  padding: 20px 0;
  position: middle;
  /*height: 100%;*/
  text-align: center;

  header{
    font-size: 40px;
    transition-duration: 1s;
    transition-timing-function: ease-in-put;
    font-weight: 200;
  }
}
```

```
.MAIN button
{
height:30px;
width:200px;
margin-left:60px;
background-color:blue;
}

.MAIN b{
font-size:20px;
font-weight:800px;
text-align:center;
font-family: 'Source Sans Pro', sans-serif;
margin-left:20px;
}
</style>
</head>

<body>

<div class="container">

<nav style="background-color: rgba(32, 153, 100, 0.3);" class="navbar navbar-light light-blue lighten-4">

</nav>
<br>
    <h2 style="color:black;">Loan approval Prediction </h2>
        <p style="color:darkslategray;padding-left: 40px;padding-right: 40px;">it's a very important process for banking organizations. The system approved or reject the loan applications.</p>

  <br>
<div class="header">
<h1>EDA Loan Status Prediction</h1>
</div>
<div class="MAIN">
```

```
<form action = "{{url_for('prediction')}}" method = "post">
<br>
<label for = "Gender">Gender</label>
<select name = "Gender">
<option value = "Male">Male</option>
<option value = "Female">Female</option>
</select>
</br>
<br>
<label for = "Married">Married</label>
<select name = "Married">
<option value = "Yes">Yes</option>
<option value = "No">No</option>
</select>
</br>
<p>Dependents<span><input type = "number" name = "Dependents"/></span></p>
<br>
<label for = "Education">Education</label>
<select name = "Education">
<option value = "-Select-">-Select</option>
<option value = "educated">educated</option>
<option value = "uneducated">uneducated</option>
</select>
</br>
<br>
<label for = "Self_Employed">Self_Employed</label>
<select name = "Self_Employed">
<option value = "yes">yes</option>
<option value = "no">no</option>
</select>
</br>
<br>
<p> ApplicantIncome <span><input type = "float" name = "ApplicantIncome"/></span></p>
<p> CoapplicantIncome <span><input type = "float" name = "CoapplicantIncome"/></span></p>
<p> LoanAmount <span><input type = "float" name = "LoanAmount"/></span></p>
```

```
<p> Loan_Amount_Term <span><input type = "number" name = "Loan_Amount_Term"/></span></p>
</br>
<br>
<label for = "Credit_History">Credit_History</label>
<select name = "Credit_History">
<option value = "yes">yes</option>
<option value = "no">no</option>
</select>
</br>
<br>
<label for = "Property_Area">Property_Area</label>
<select name = "Property_Area">
<option value = "urban">urban</option>
<option value = "semiurban">semiurban</option>
<option value = "rural">rural</option>
</select>
</br>

<br>
<button type="submit" >SUBMIT</button></br>
</form>
<b>{{y}}</b>

</div>
</body>
</html>
```