NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY
[IJIERT] ISSN: 2394-3696
Website: ijiert.org
VOLUME 6, ISSUE 12, Dec. -2019

A NEW WAY OF PREDICTING THE LOAN APPROVAL PROCESS USING ML TECHNIQUES

Chaitanya Krishna Suryadevara
Department of Information Systems Wilmington University
chaitanyakrishnawork123@gmail.com

Abstract

Loans account for a large portion of bank profits. Despite the fact that many people are looking for loans. Finding a legitimate applicant who will return the loan is difficult. There may be many errors made when selecting the real applicant when the process is done manually. As a result, we are creating a machine learning-based loan prediction system that will choose the qualified applicants on its own. Both the applicant and the bank staff will benefit from this. There will be a significant reduction in the loan sanctioning period of time. In this research, we use different machine learning techniques to predict the loan data.

Keywords: loan, bank, legitimate applicant, loan prediction, Machine Learning, Vector system.

Introduction

A loan is the foundation of a bank's operations. The majority of the bank's profits are derived directly from the money made from the loans. Even when the bank authorizes the loan following a lengthy verification and testimonial process, there is no guarantee that the chosen hopeful is the appropriate hopeful. When done manually, this operation requires additional time. We are able to foretell if a specific hopeful is secure or not, and the entire testimonial procedure is mechanized using machine literacy.Credit risk is the chance that the loan won't be paid back on time or at all; liquidity risk is the chance that too many deposits will be withdrawn too quickly, leaving the bank cash-strapped; and interest rate risk is the chance that interest rates on bank loans will be too low to generate enough revenue for the bank. For bank clients as well as potential borrowers, loan prognostic is quite beneficial.

The goal of this project is to provide a quick, easy, and immediate technique for choosing qualified applications. The bank may gain in a number of ways, such as by imposing a deadline for applicants to check and confirm whether or not their loan will be approved.This forecasting approach might be useful in that it allows bankers to concentrate more on valuable assets rather than on unqualified applicants. The applicant's loan application process will take less time as a result. "Results for a certain Loan Id can be sent to other bank departments so they can handle applications in a suitable manner. This facilitates the completion of other formalities by all other departments.

Methodology

We have used different classification algorithms to find out which one best predicts the loan status with most accuracy.

Random Forest:In Supervised Machine Learning, Random Forest is a well-known learning technique that works well for Regression & Classification applications.It creates random forests and then searches through them for solutions.It is an ensemble learning method where a sizable number of classifiers are employed to address a challenging issue.To avoid overfitting difficulties, random forest considers each tree for prediction rather than just one.

Support Vector Machine (SVM):A popular supervised machine learning algorithm is SVM. The SVM classifier is currently the most popular classifier.SVM has proven to have a wide variety of excellent skills, especially in classification issues.

Logistic Regression (LR):Logistic regression, which belongs to the supervised learning approach, is one of the most popular machine learning algorithms.The sample size for LR should be large.It is employed for categorical target variable prediction. It does not produce 0 or 1, but rather returns the probability value.

Data Collection:

A training set and a testing set are created using the dataset gathered for loan failure prediction customers. The 80/20 rule is typically used to separate the training set from the testing set. Forecasting for the test set is done using the data model that was developed using a decision tree and applied to the training set. the following characteristics:

- Loan_id-Unique loan id
- Gender-Male / female
- Married-Applicant Married(Y/N)
- Dependents-Number of dependents
- Education-Applicant education(graduate/undergraduate)
- Self_employed-Self employed(Y/N)
- Applicant income-Applicant income
- Co-Application income-Co_application 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)

```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split, RandomizedSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import accuracy_score, roc_auc_score
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import GridSearchCV
from sklearn import metrics

import matplotlib.pyplot as plt
!pip install xgboost
import xgboost as xgb
from xgboost import XGBClassifier
```

```
loan_train = pd.read_csv('./loan-train.csv')
print(loan_train.shape) # (614, 13)
loan_train.head()
```

Preprocessing: The collected data may contain missing values that may lead to inconsistency. To gain better results data needs to be preprocessed and so it'll better the effectiveness of the algorithm. We should remove the outliers and we need to convert the variables. In order to cover these issues we use a chart function.

Train model on training data set:

Now that the model has been trained on the training data, predictions should be made for the test data. Our train dataset may be split into two tracts: testimony and train. On the basis of the training portion, we can train the model to generate predictions for the testimony portion. We can validate our prophecies in this way since we have the actual prophecies for the testimony portion (which we do not have for the test dataset).

```
    Loan_ID Gender Married Dependents     Education Self_Employed  \
0  LP001002   Male      No          0      Graduate            No
1  LP001003   Male     Yes          1      Graduate            No
2  LP001005   Male     Yes          0      Graduate           Yes
3  LP001006   Male     Yes          0  Not Graduate            No
4  LP001008   Male      No          0      Graduate            No

   ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
0             5849                0.0         NaN             360.0
1             4583             1508.0       128.0             360.0
2             3000                0.0        66.0             360.0
3             2583             2358.0       120.0             360.0
4             6000                0.0       141.0             360.0

   Credit_History Property_Area Loan_Status
0             1.0         Urban           Y
1             1.0         Rural           N
2             1.0         Urban           Y
3             1.0         Urban           Y
4             1.0         Urban           Y
```

Apply Model:

Before applying, we find out which classifier best predicts the loan status and after that we applied the model to the suitable one.Apply the built model on a test dataset.

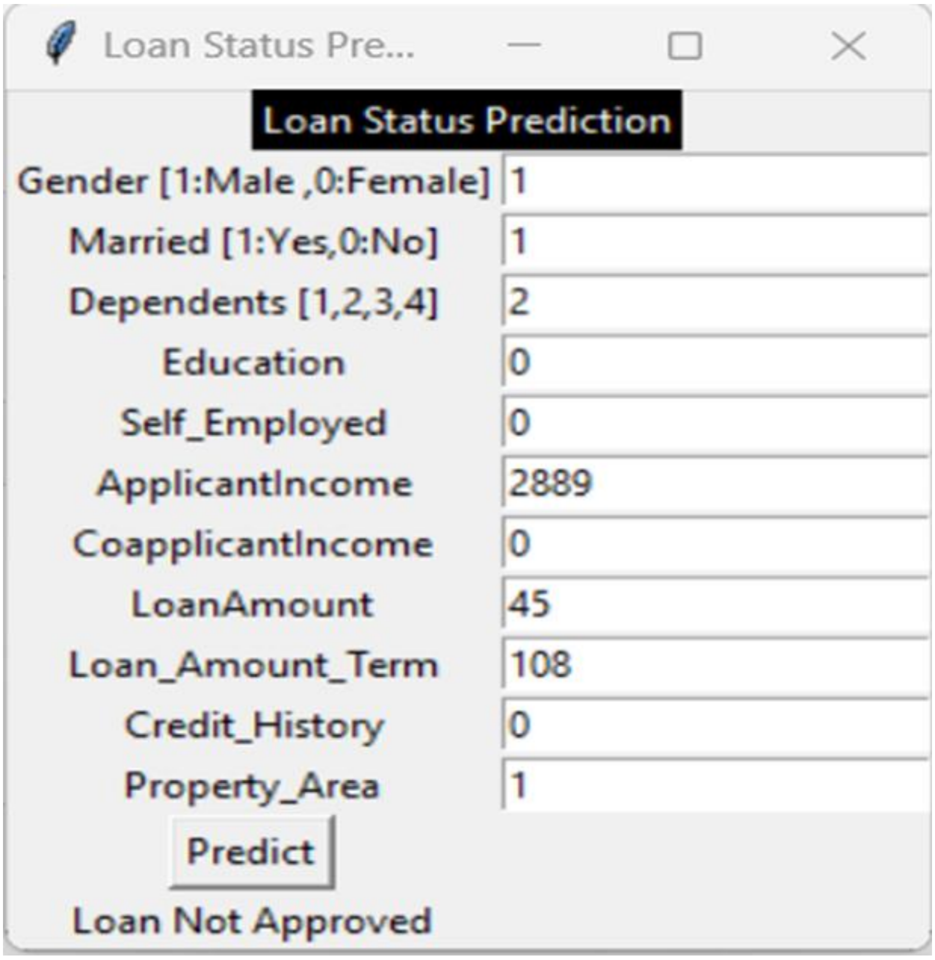
```
total_null = loan_train.isnull().sum().sort_values(ascending=False)
total_null.head(10)
```

```
Credit_History      50
Self_Employed       32
LoanAmount          22
Dependents          15
Loan_Amount_Term    14
Gender              13
Married              3
Loan_ID              0
Education            0
ApplicantIncome      0
dtype: int64
```

Generating the prediction:

Classify the applicants based on the applicant's information and bank's criteria using random forest classifier.

Output



Next, we can try a single decision tree with the max depth ranging from 4 to 25 and minimum samples for leaf and split between 10 and 100. 4 is the best max depth, while the ideal criterion is the default 'Gini' index.

Appendices

```
import pandas as pd
data = pd.read_csv('loan_prediction.csv')
data.head()
data.tail()
Data.shape
data.info()
data.isnull().sum()
data.isnull().sum()*100 / len(data)data = data.drop('Loan_ID',axis=1)
data.head(1)
columns = ['Gender','Dependents','LoanAmount','Loan_Amount_Term']
data = data.dropna(subset=columns)
data.isnull().sum()*100 / len(data)
data['Self_Employed'].mode()
[0]data['Gender'].unique()
data['Self_Employed'].unique()
data['Credit_History'].mode()[0]
data['Credit_History']=data['Credit_History'].fillna(data['Credit_History'].mode()[0])
data.isnull().sum()*100 / len(data)
data['Dependents']=data['Dependents'].replace(to_replace="3+",value='4')
data['Dependents'].unique()
data['Loan_Status'].unique()
data.head()
X = data.drop('Loan_Status',axis=1)
y = data['Loan_Status']
y
data.head()
cols = ['ApplicantIncome','CoapplicantIncome','LoanAmount','Loan_Amount_Term']
from sklearn.preprocessing import StandardScaler
st = StandardScaler()
X[cols]=st.fit_transform(X[cols])
X
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score
import numpy as np
model_df
```

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model_val(model,X,y)
from sklearn import svm
model = svm.SVC()
model_val(model,X,y)
from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model_val(model,X,y)
from sklearn.ensemble import RandomForestClassifier
model =RandomForestClassifier()
model_val(model,X,y)
from sklearn.ensemble import GradientBoostingClassifier
model =GradientBoostingClassifier()
model_val(model,X,y)
from sklearn.model_selection import RandomizedSearchCV
log_reg_grid={"C":np.logspace(-4,4,20),
            "solver":['liblinear']}
rs_log_reg=RandomizedSearchCV(LogisticRegression(),
                  param_distributions=log_reg_grid,
                 n_iter=20,cv=5,verbose=True)
rs_log_reg.fit(X,y)
rs_log_reg.best_score_
rs_log_reg.best_params_
svc_grid = {'C':[0.25,0.50,0.75,1],"kernel":["linear"]}
rs_svc=RandomizedSearchCV(svm.SVC(),
                 param_distributions=svc_grid,
                  cv=5,
                  n_iter=20,
                 verbose=True)
rs_svc.fit(X,y)
rs_svc.best_score_
rs_svc.best_params_
RandomForestClassifier()
rf_grid={'n_estimators':np.arange(10,1000,10),
  'max_features':['auto','sqrt'],
 'max_depth':[None,3,5,10,20,30],
 'min_samples_split':[2,5,20,50,100],
 'min_samples_leaf':[1,2,5,10]
 }
rs_rf=RandomizedSearchCV(RandomForestClassifier(),
                 param_distributions=rf_grid,
```

```
                  cv=5,
                  n_iter=20,
                 verbose=True)
rs_rf.fit(X,y)
rs_rf.best_score_
rs_rf.best_params_
X = data.drop('Loan_Status',axis=1)
y = data['Loan_Status']
rf = RandomForestClassifier(n_estimators=270,
 min_samples_split=5,
 min_samples_leaf=5,
 max_features='sqrt',
 max_depth=5)
rf.fit(X,y)
import joblib
joblib.dump(rf,'loan_status_predict')
model = joblib.load('loan_status_predict')
import pandas as pd
df = pd.DataFrame({
    'Gender':1,
    'Married':1,
    'Dependents':2,
    'Education':0,
    'Self_Employed':0,
    'ApplicantIncome':2889,
    'CoapplicantIncome':0.0,
    'LoanAmount':45,
    'Loan_Amount_Term':180,
    'Credit_History':0,
    'Property_Area':1
},index=[0])
df
result = model.predict(df)
if result==1:
    print("Loan Approved")
else:
    print("Loan Not Approved")
from tkinter import *
import joblib
import pandas as pd
def show_entry():
```

```
    p1 = float(e1.get())
    p2 = float(e2.get())
    p3 = float(e3.get())
    p4 = float(e4.get())
    p5 = float(e5.get())
    p6 = float(e6.get())
    p7 = float(e7.get())
    p8 = float(e8.get())
    p9 = float(e9.get())
    p10 = float(e10.get())
    p11 = float(e11.get())

    model = joblib.load('loan_status_predict')
    df = pd.DataFrame({
    'Gender':p1,
    'Married':p2,
    'Dependents':p3,
    'Education':p4,
    'Self_Employed':p5,
    'ApplicantIncome':p6,
    'CoapplicantIncome':p7,
    'LoanAmount':p8,
    'Loan_Amount_Term':p9,
    'Credit_History':p10,
    'Property_Area':p11
},index=[0])
    result = model.predict(df)

    if result == 1:
        Label(master, text="Loan approved").grid(row=31)
    else:
        Label(master, text="Loan Not Approved").grid(row=31)

master =Tk()
master.title("Loan Status Prediction Using Machine Learning")
label = Label(master,text = "Loan Status Prediction",bg = "black",
               fg = "white").grid(row=0,columnspan=2)

Label(master,text = "Gender [1:Male ,0:Female]").grid(row=1)
Label(master,text = "Married [1:Yes,0:No]").grid(row=2)
Label(master,text = "Dependents [1,2,3,4]").grid(row=3)
Label(master,text = "Education").grid(row=4)
```

```
Label(master,text = "Self_Employed").grid(row=5)
Label(master,text = "ApplicantIncome").grid(row=6)
Label(master,text = "CoapplicantIncome").grid(row=7)
Label(master,text = "LoanAmount").grid(row=8)
Label(master,text = "Loan_Amount_Term").grid(row=9)
Label(master,text = "Credit_History").grid(row=10)
Label(master,text = "Property_Area").grid(row=11)


e1 = Entry(master)
e2 = Entry(master)
e3 = Entry(master)
e4 = Entry(master)
e5 = Entry(master)
e6 = Entry(master)
e7 = Entry(master)
e8 = Entry(master)
e9 = Entry(master)
e10 = Entry(master)
e11 = Entry(master)

e1.grid(row=1,column=1)
e2.grid(row=2,column=1)
e3.grid(row=3,column=1)
e4.grid(row=4,column=1)
e5.grid(row=5,column=1)
e6.grid(row=6,column=1)
e7.grid(row=7,column=1)
e8.grid(row=8,column=1)
e9.grid(row=9,column=1)
e10.grid(row=10,column=1)
e11.grid(row=11,column=1)

Button(master,text="Predict",command=show_entry).grid()
mainloop()
```

```
svm_param_grid = {
    'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
    'C':range(1,11)
    }
svm = SVC()

svm_random = RandomizedSearchCV(param_distributions=svm_param_grid,
                                estimator = svm, scoring = "accuracy",
                                verbose = 0, n_iter = 100, cv = 4)
svm_random.fit(X_train,y_train)
best_params = svm_random.best_params_
print(f'Best parameters: {best_params}')
# Best parameters: {'kernel': 'linear', 'C': 1}

y_pred_best=svm_random.predict(X_test)
acc = metrics.accuracy_score(y_test,y_pred_best)
print(acc)
# 0.788638980231
```

Conclusion :

From a proper point of view of analysis this system can be Perfectly used to find customers who qualify Loan approval. The software is perfect and can work To be used for all banking needs. This may be the system Can be easily uploaded to any operating system. Since then As technology moves online, there is more to this system Space for the coming days. This system is more secure and reliable. Since we used random forest Algorithm system gives very accurate results. There There is no problem if many customers are applying Loan This system accepts data for N no. of customers. In In the future we may add more algorithms to this system Get more accurate results.

References

[1]. Yadav, O. P., Soni, C., Kandakatla, S. K., Sswanth, S. (2019). Loan prediction using decision tree. International Journal of Information and Computer Science, 6(5).

[2].Kumar Arun, Garg Ishan, Kaur Sanmeet, May-Jun. 2016. Loan Approval Prediction based on Machine Learning Approach, IOSR Journal of Computer Engineering (IOSR-JCE)

[3] “Prediction for Loan Approval using Machine Learning Algorithm” International Research Journal of Engineering and Technology (IRJET).

[4]X.FrencisJensy, V.P.Sumathi,Janani Shiva Shri, “An exploratory Data Analysis for Loan Prediction based on nature of clients”, International Journal of Recent Technology and Engineering (IJRTE),Volume-7 Issue-4S, November 2018

[5]https://www.researchgate.net/publication/357449126_THE_LOAN_PREDICTION_USING_MACHINE_LEARNING

[6]Nitesh Pandey1 , Ramanand Gupta2 , Sagar Uniyal3 , Vishal Kumar4 1,2,3,4 Department of Computer Science and Engineering, Raj Kumar Goel Institute of Technology, AKTU

[7]WITTEN, H., and FRANK, E. Data mining practical machine learning tools and techniques with Java implementations. Elsevier, 2017.

[8]ZAMANI, S., and MOGADDAM, A. Natural Customer Ranking of Banks in Terms of Credit Risk by Using Data Mining: A Case Study: Branches of Mellat Bank of Iran. Journal of UMP Social Science Technology Management, 2016, 3(2).

[9]JIN, Y., and ZHU, Y. A data-driven approach to predicting loan default risk for online peer-to-peer (P2P) lending. 2015 Fifth International Conference on Communication Systems and Network Technologies, 2017, 609–613.

[10]HAMID, A.J., and AHMED, T.M. Developing prediction model of loan risk in banks using data mining. Machine Learning Appliances an International Journal, 2016, 3(1).

[11]TURKSON, R.E., BAAGYERE, E.Y., and WENYA, G.E. A machine learning approach for predicting bank creditworthiness. 2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR), 2016, 1–7.

[12]LAWI, A., AZIZ, F., and SYARIF, S. Ensemble GradientBoost for increasing classification accuracy of credit scoring. 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), 2017, 1–4.

[13]BAGHERPOUR, A. Predicting mortgage loan default with machine learning methods. University of California/Riverside, 2017.

[14]BAE, J.K., and KIM, J. A personal credit rating prediction model using data mining in ubiquitous smart environments. International Journal of Distributed Sensor Networks, 2018, 11(9): 179060.

[15]KUMAR, B., BAWANE, I., SHIRSATHE, A., and PARDESHI, P. An Expert System Based On Fuzzy Logic for Automated Decision Making For Loan Approval. 2016.

[16]YADAV, O., SONI, C., KANDAKATLA, S., and SAWANT, S. Loan Prediction System Using Decision Tree.

[17]COŞER, A., MAER-MATEI, M.M., and ALBU, C. Predictive Models for Loan Default Risk Assessment. Economical Computer: Economic Cybernetics Study Resources, 2019, 53(2).

[18]VAIDYA, A. Predictive and probabilistic approach using logistic regression: Application to prediction of loan approval. 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017, 1–6.

[19]PRIYA, K.U., PUSHPA, S., KALAIVANI, K., and SARTIHA, A. Exploratory Analysis on Prediction of Loan Privilege for Customers using Random Forest. International Journal of Engineering Technology, 2018, 7(2.21): 339–341.

[20]JIANG, C., WANG, Z., WANG, R., and DING, Y. Loan default prediction by combining soft information extracted from descriptive text in online peer-to-peer lending. Annual Operation Resources, 2018, 266(1–2): 511–529.

[21]ARUN, K., ISHAN, G., and SANMEET, K. Loan Approval Prediction based on Machine Learning Approach. IOSR Journal of Computing Engineering, 2016, 18(3): 18–21.

[22]GAHLAUT, A., and SINGH, P.K. Prediction analysis of risky credit using Data mining classification models. 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017, 1–7.

[23]TANEJA, S., SURI, B., GUPTA, S., NARWAL, H., JAIN, A., and KATHURIA, A. A fuzzy logic-based approach for data classification. Data engineering and intelligent computing. Springer, 2018.