PREDICTIVE AND PROBABILISTIC APPROACH FOR LOAN PREDICTION

Cholleti Madhana chary
Email: madhan530@gmail.com
M. Tech, Department of Computer Science and Engineering.
Annamacharya Institute of Technology and Science, Hyderabad, Telangana, India.

Ramesh Babu Varugu
Email: rameshvarugu82@gmail.com
Assistant Professor & HOD, Department of CSE.
Annamacharya Institute of Technology and Science, Hyderabad, Telangana, India.

Abstract - In our banking system, main source of income of any banks is on its credit line. So they can earn from interest of thoseloans which they provide. A bank's profit or a loss depends to a large extent on loans i.e. whether the customers are paying back the loan or defaulting. By predicting the loan defaulters, the bank can reduce its Non- Performing Assets.This makes the study of this phenomenon very important. Previous research in this era has shown that there are so many methods to study the problem of controlling loan default. But as the right predictions are very important for the maximization of profits, it is essential to study the nature of the different methods and their comparison. The data is collected from the Kaggle for analysis and prediction. Various machine learning models have been performed and the different measures of performances are computed. The models are compared on the basis of the performance measures such as sensitivity and specificity. The results have shown that the different models produce different performance. Dataset includes variables (personal attributes of customer like age, purpose, credit history, credit amount, credit duration, etc.) other than checking account information (which shows wealth of a customer) that should be taken into account to calculate the probability of default on loan correctly. Therefore, by using machine learning models, the right customers to be targeted for granting loan can be easily detected by evaluating their likelihood of default on loan.

Keywords – Loan, Prediction, Logistic regression, Decision tree, Loan defaulters, Random Forest.

I. INTRODUCTION

The technical world is advancing toward complete automation. In order to attain automation various concepts are being developed and put to use, as can be observed from the numerous developments being made and symposiums being held. One of the most striking features that excites scientists and technologists, in regards to the development of automation, is Artificial Intelligence. Artificial Intelligence is the concept of simulate humanlike intelligence in a computer. To make a machine think exactly like a human is intriguing to the scientists and developers, and they striveto achieve this goal by putting Artificial Intelligence to use. The idea is not to overpower the human society but to work with the man so that the combined intelligence can lead to many more revelations in this technological era. Artificial Intelligence dates back to the time of advent of computers and since then it has diversified into numerous field. Over the years, technologists have acquired a great understanding in this field which has lead to development of defined models and further application of these models to real world problems. Various domains in Artificial Intelligence domain include machine learning, neural networks, fuzzy logic , natural language processing , expert systems. These concepts are deployed according to the specificity of the desired requirements. In regards with this paper , one of the concepts in the domain of machine learning is exploited and also applied to a real world application. Machine Learning is a tool which facilitates development of analytical models without explicit programming. Various machine learning algorithms are developed to tailor to the problem requirements. All the leading edge industries are now utilizing the capabilities of machine learning to gain higher sales growth and statistics have shown that they are getting positive results. With institutions generating more and more data, exploitation of data manually becomes difficult, hence machine learning, having the capability of analytical modelling is sought to, as a solution.

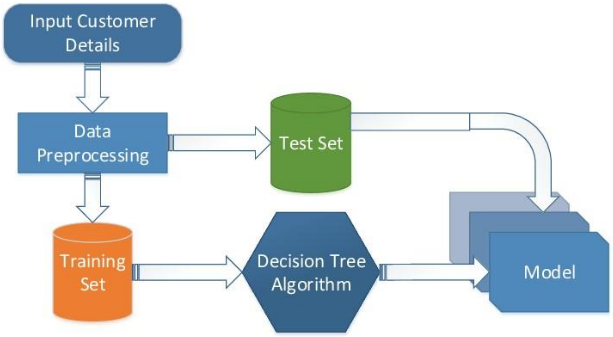
II. PROBLEM STATEMENT

Banks, Housing Finance Companies and some NBFC deal in various types of loans like housing loan, personal loan, business loan etc., in all over the part of countries. These companies have existence in Rural, Semi Urban and Urban areas. After applying loan by customer these companies validates the eligibility of customers to get the loan or not. This paper provides a solution to automate this process by employing machine learning algorithm. So the customer will fill an online loan application form. This form consists detailslike Sex, Marital Status, Qualification, Detailsof Dependents, Annual Income, Amount of Loan, Credit History of Applicant and others. Algorithm will identify those segments of the customers who are eligible to get loan amounts so bank can focus on these customers.

III. PROPOSED SYSTEM

In regards with this paper, one of the concepts in the domain of machine learning is exploited and also applied to a real world application. Machine Learning is a tool which facilitates development of analytical models without explicit programming. Various machine learning algorithms are developed to tailor to the problem requirements. All the leading edge industries are now utilizing the capabilities of machine learning to gain higher sales growth and statistics have shown that they are gettingpositive results. With institutions generating more and more data, exploitation of data manually becomes difficult, hence machine learning, having the capability of analytical modelling is sought to, as a solution.

IV. SYSTEM DESIGN:



Architecture Diagram of Proposed Framework.

DATA FLOW DIAGRAM:

- The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

- The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows inthe system.
- DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that areapplied as data moves from input tooutput.
- DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

V. SYSTEM ANALYSIS

Requirement Specification

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed.

In software engineering, such requirements are often called functional specifications. Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

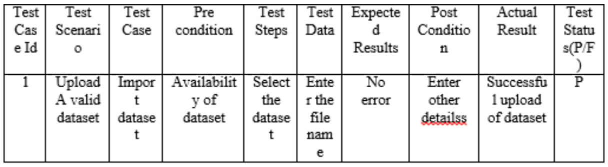
Data Set

This paper involved dataset taken from Kaggle named loan train. The dataset was entirely collected from the previous bank loan data. The features that are included in the dataset are Gender, Marital status, Dependents, Education, Employment status, Income, Credit history, Property area. The dataset includes some missing values and also outliers in data.

VI. IMPLEMENTATION:

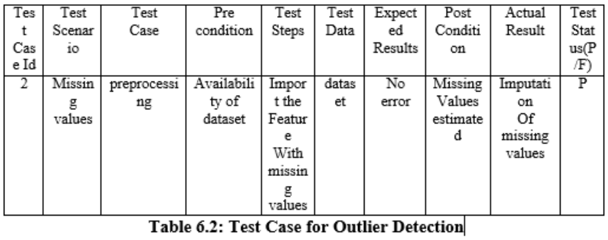
- Loading require python classes
- Preprocessing dataset
- Apply label encoder
- Shuffling & normalizing dataset values
- Splitting data into train & test
- Calculate weights using random forest
- Train decision tree with weights
- Train decision tree without weights

VII. TEST CASES



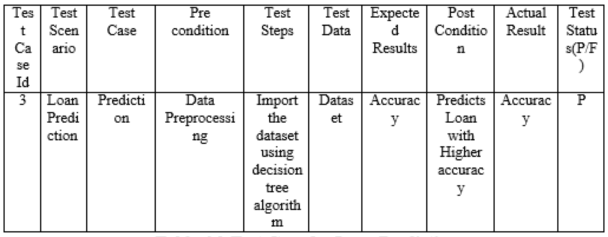
Test Case Id	Test Scenario	Test Case	Pre condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Test Status(P/F)
1	Upload A valid dataset	Import dataset	Availability of dataset	Select the dataset	Enter the file name	No error	Enter other detailss	Successful upload of dataset	P

Table 6.1: Test case to Import Dataset



Test Case Id	Test Scenario	Test Case	Pre condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Test Status(P/F)
2	Missing values	preprocessing	Availability of dataset	Import the Feature With missing values	dataset	No error	Missing Values estimated	Imputation Of missing values	P

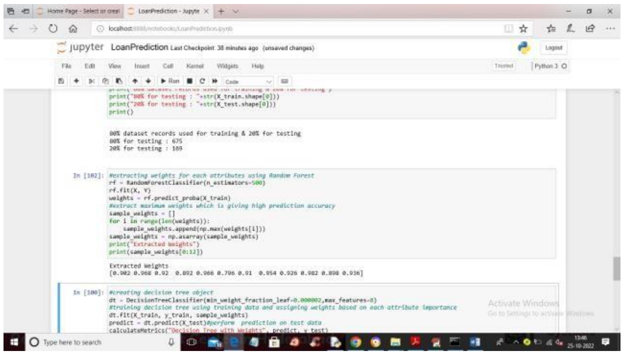
Table 6.2: Test Case for Outlier Detection



Test Case Id	Test Scenario	Test Case	Pre condition	Test Steps	Test Data	Expected Results	Post Condition	Actual Result	Test Status(P/F)
3	Loan Prediction	Prediction	Data Preprocessing	Import the dataset using decision tree algorithm	Dataset	Accuracy	Predicts Loan with Higher accuracy	Accuracy	P

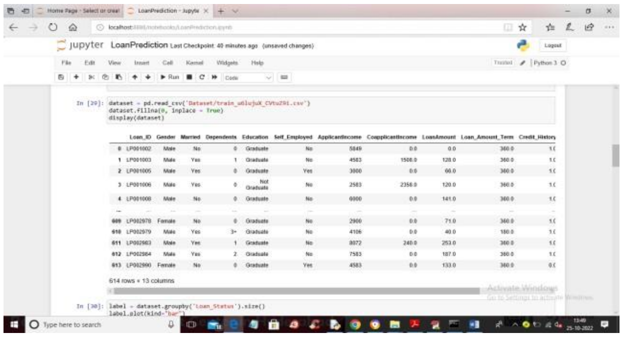
Table 6.3:Test Case for Loan Prediction

VIII. RESULTS



```
    print("80% for testing : "+str(X_train.shape[0]))
    print("20% for testing : "+str(X_test.shape[0]))
    print()

80% dataset records used for training & 20% for testing
80% for testing : 675
20% for testing : 169

In [102]: #extracting weights for each attributes using Random Forest
          rf = RandomForestClassifier(n_estimators=500)
          rf.fit(X, Y)
          weights = rf.predict_proba(X_train)
          #extract maximum weights which is giving high prediction accuracy
          sample_weights = []
          for i in range(len(weights)):
              sample_weights.append(np.max(weights[i]))
          sample_weights = np.asarray(sample_weights)
          print("Extracted Weights")
          print(sample_weights[0:12])

Extracted Weights
[0.902 0.968 0.92  0.892 0.966 0.796 0.91  0.954 0.926 0.982 0.898 0.936]

In [100]: #creating decision tree object
          dt = DecisionTreeClassifier(min_weight_fraction_leaf=0.000002,max_features=8)
          #training decision tree using training data and assigning weights based on each attribute importance
          dt.fit(X_train, y_train, sample_weights)
          predict = dt.predict(X_test)#perform  prediction on test data
          calculateMetrics("Decision Tree with Weights", predict, y_test)
```

In above screen using random forest we got weights for each attribute and this weight will be assigned to decision tree.



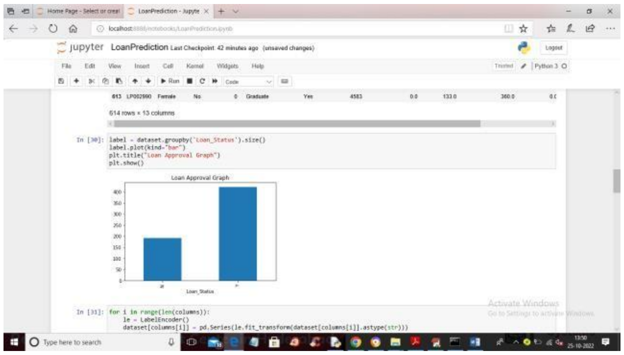
```
In [29]: dataset = pd.read_csv('Dataset/train_u6lujuX_CVtuZ9i.csv')
         dataset.fillna(0, inplace = True)
         display(dataset)
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
0	LP001002	Male	No	0	Graduate	No	5849	0.0	0.0	360.0	1.(
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.(
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.(
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.(
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.(
...	...	...	...	...	...	...	...	...	...	...	...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	360.0	1.(
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	180.0	1.(
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	360.0	1.(
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	360.0	1.(
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	360.0	0.(

614 rows × 13 columns

```
In [30]: label = dataset.groupby('Loan_Status').size()
         label.plot(kind="bar")
```

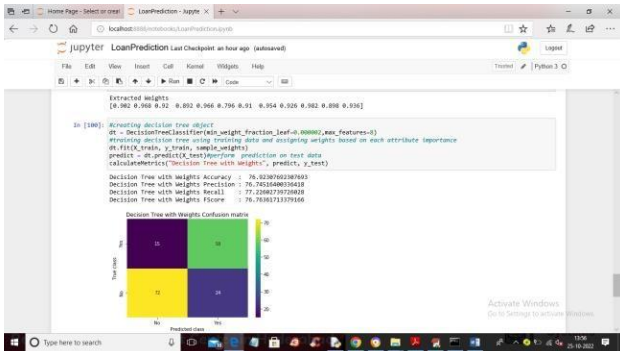
In above screen we are reading and displaying dataset values and we can see maximum columns contains non-numeric data but machine learning algorithms only accept numeric data so by using label encoder we will convert all non-numeric data to numeric



```
613  LP002990  Female  No  0  Graduate  Yes  4583  0.0  133.0  360.0  0.(

614 rows × 13 columns

In [30]: label = dataset.groupby('Loan_Status').size()
         label.plot(kind="bar")
         plt.title("Loan Approval Graph")
         plt.show()
```

```
In [31]: for i in range(len(columns)):
             le = LabelEncoder()
             dataset[columns[i]] = pd.Series(le.fit_transform(dataset[columns[i]].astype(str)))
```

In above graph we are plotting number of YES and NO records available in dataset



```
Extracted Weights
[0.902 0.968 0.92  0.892 0.966 0.796 0.91  0.954 0.926 0.982 0.898 0.936]

In [100]: #creating decision tree object
          dt = DecisionTreeClassifier(min_weight_fraction_leaf=0.000002,max_features=8)
          #training decision tree using training data and assigning weights based on each attribute importance
          dt.fit(X_train, y_train, sample_weights)
          predict = dt.predict(X_test)#perform  prediction on test data
          calculateMetrics("Decision Tree with Weights", predict, y_test)

Decision Tree with Weights Accuracy  :  76.92307692307693
Decision Tree with Weights Precision : 76.74516400336418
Decision Tree with Weights Recall    : 77.22602739726028
Decision Tree with Weights FScore    : 76.76361713379166
```

In above screen decision tree with weights we got 76% accuracy and this accuracy may vary at each run as algorithm get trained on random train data and test on random test data and in confusion matrix graph x-axis represents Predicted classes and y-axis represents true classes and blue colour boxes represents INCORRECT prediction count and different colour boxes represents CORRECT prediction count

Machine Learning Models	Accuracy	Precision	Recall	F1 Score
Decision Tree without weights	71.00	70.47	70.54	70.50
Decision Tree with weights	76.92	76.74	77.22	76.76
Bagging without weights	73.37	73.51	73.47	73.39
Bagging with weights	75.14	75.47	75.30	75.12
XGBoost without weights	75.73	75.89	75.84	75.73
XGBoost with weights	77.51	77.54	77.56	77.51

Comparison of models

IX. CONCLUSION

The process of prediction starts from cleaning and processing of data, imputation of missingvalues, experimental analysis of data set and then model building to evaluation of model and testing on test data. On Data set, the best case accuracy obtained on the original data set is 0.811. The following conclusions are reachedafter analysis that those applicants whose credit score was worst will fail to get loan approval, due to a higher probability of not paying back the loan amount. Most of the time, thoseapplicants who have high income and demands for lower amount of loan are more likely to get approved which makes sense, more likely to pay back their loans. Some other characteristic like gender and marital status seems not to be taken into consideration by the company.

X. FUTURE SCOPE

Various machine learning models exist for predictive analysis like logistic regression, decision trees, and artificial neural networks (ANN) and Bayesian Networks. While decision tree is a statistical model the other three are graphical models. ANNs have a very complex structure with multiple layers of nodes. Decision tree and ANNs are most widely used because they are easy to develop and provide most accurate predictive analysis. Decision tree canhandle non linear effect and power terms. The independent variable based on which prediction takes place need not be normally distributed. And also we showed bagging classifier and XGBoost algorithms with & without weights in this paper.

XI. REFERENCES

[1] Chao-Ying Joanne Peng, Kuk Lida Lee, Gary M. Ingersoll, "An introduction to logistic Regression Analysis and Reporting", The journal of Educational Research Indiana University Bloomington, vol. 96, PP 1-13, September/October 2002.

[2] Kevin P. Murphy, Machine Learning a Probabilistic Approach, Massachusetts Institute of Technology 2012, pp. 1-21.

[3] Vijayalakshmi Sampath, Andrew Flagel, Carolina Figueroa, "A Logistic Regression Model to Predict Freshmen Enrolments", Northern Virginia Community College VA, pp. 1-12, May 2016

[4] Yhat- Machine Learning, Data Science and Engineering blog, "Logistic Regression in Python", March 2013.

[5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman," The Elements of Statistical Learning: Data Mining, Inference, and Prediction," Springer, Kindle Toby Segaran, "Programming Collective Intelligence: Building Smart Web 2.0 Applications." O'Reilly Media.

[6] Pidikiti Supriya, Myneedi Pavani, Nagarapu Saisushma, Namburi Vimala Kumari, k Vikash,"Loan Prediction by using Machine Learning Models", International Journal of Engineering and Techniques. Volume 5 Issue 2, Mar-Apr 2019.

[7] Nikhil Madane, Siddharth Nanda," Loan Prediction using Decision tree", Journal of the Gujrat Research History, Volume 21 Issue 14s, December 2019.

[8] Aakanksha Saha, Tamara Denning, Vivek Srikumar, Sneha Kumar Kasera. "Secrets inSource 20 Code: Reducing False Positives using Machine Learning", 2020 International Conference on Communication Systems &Networks (COMSNETS), 2020.