# **Seminar Report on**

# Loan Prediction System using Ada-boost With Decission Tree

#### *Submitted by*

**Raghunatha Panda**

Regd. No: 1901229076

Seminar Report submitted in partial fulfillment of the requirements for the award of degree of B.Tech. in Computer Science & Engineering under

Biju Patnaik University of Technology (BPUT)



**2019- 2023**

#### *Under the Guidance of*

### Shekharesh Barik

#### Asso. Prof., Dept. of CSE



## Department of Computer Science and Engineering

### DRIEMS(Autonomous),Tangi,Cuttack-754022

Department of Computer Science & Engineering

**DRIEMS(Autonomous),Tangi,Cuttack - 745022**

**Certificate**

This is to certify that, this is a bonafide Seminar report, titled “**Loan Prediction System Using Ada-boosting With Decision Tree**”, done satisfactorily by Raghunatha Panda (1901229076) in partial fulfillment of requirements for the degree of B.Tech. in Computer Science & Engineering under Biju Patnaik University of Technology (BPUT).

This Seminar report on the above mentioned topic has not been submitted for any other examination earlier before in this institution and does not form part of any other course undergone by the candidate.

|  |  |
| --- | --- |
| **Shekharesh Barik** | **Surajit Mohanty** |
| Asso. Prof., Dept. of CSE | Asso. Professor & Head |
| Guide | Dept of CSE |

**ACKNOWLEDGEMENT**

I express my indebtedness to my guide **Shekharesh Barik**, Associate Professor of the Computer Science & Engineering department who spared his valuable time to go through manuscript and offer his scholar advice in the writing. His guidance, encouragement and all out help have been invaluable to me. There is short of words to express my gratitude and thankfulness to him.

I am grateful to all the teachers of Computer Science & Engineering department, DRIEMS, for their encouragement, advice and help.

At the outset, I would like to express my sincere gratitude to **Surajit Mohanty**, H.O.D of Computer Science & Engineering department for his moral support extended towards me throughout the duration of this seminar.

I am also thankful to my friends who have helped me directly or indirectly for the success of this seminar.

### Raghunatha Panda

Regd. No.: 1901229076

### Department of Computer Science & Engineering

DRIEMS Autonomous Engg. College

#### ABSTRACT

Banking system have large number of products to earn profit, but their vital source of income is from its credit system. Because Credit system can earn from interests of that loans which they credit. Banking system always need accurate modelling system for large number of issues. The prediction of credit defaulters is one of the difficult task for any bank. But by forecasting the loan defaulters, the banks definitely may reduce its loss by reducing its non-profit assets, so that recovery of approved loans can take place without any loss and it can play as the contributing parameter of the bank statement. This makes the study of this loan approval prediction important. Machine Learning techniques are very crucial and useful in prediction of these type of data. In this research paper four algorithms of classification based machine learning that is Ada-boosting with Decision tree, Ada-boosting with Random forest and XGboosting is applied and among them Ada-boosting with Decision tree algorithm is most accurate to predict the loan approval with large accuracy.

Keywords - Loan, Machine Learning, Prediction, Testing, Training.

**CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| LIST OF FIGURES | | | 1 |
| CHAPTER 1 | | | 2 |
| 1 | INTRODUCTION | | 2 |
| 1.1 | PROBLEM DEFINITION | | 3 |
| 1.2 | MOTIVITION FOR WORK | | 3 |
| 1.3 | OBJECTIVES OF THE PROJECT | | 3 |
| CHAPTER 2 | | | 4 |
| 2 | SYSTEM ANALYSIS AND DESIGN | | 4 |
| 2.1 | EXISTING SYSTEM | | 4 |
| 2.2 | PROPOSED SYSTEM | | 4 |
|  | 2.2.1 | COLLECTION OF DATASET | 5 |
|  | 2.2.2 | DATA PREPROCESSING | 5 |
|  | 2.2.3 | TRAIN MODEL ON TRAINING DATA SET | 6 |
|  | 2.2.4 | CORRELATING ATTRIBUTES | 7 |
| 2.3 | DESIGN REQUIREMENTS | | 7 |
|  | 2.3.1 | HARDWARE AND SOFTWARE REQUIREMENTS | 7 |
|  | 2.3.2 | PYTHONE LIBRARIES | 7 |
| CHAPTER 3 | | | 9 |
| 3 | METHODS AND ALGORITHMS USED | | 9 |
| 3.1 | BASIC OF ADA-BOOSTING | | 9 |
|  | 3.1.1 | MODEL-1 | 10 |
|  | 3.1.2 | MODEL-2 | 10 |
|  | 3.1.3 | MODEL-3 | 11 |
|  | 3.1.4 | FINAL MODEL | 11 |
| 3.2 | ADA-BOOSTING WITH DECISION TREE | | 12 |
|  | 3.2.1 | BOOSTING | 13 |
|  | 3.2.2 | DECISION TREES | 14 |
| 3.3 | ADA BOOSTING WITH RANDOM FOREST | | 17 |
| 3.4 | XGBOOST(Extreme Gradient Boosting) | | 17 |
| CHAPTER 4 | | | 19 |
| 4 | RESULT AND DISCUSSION | | 19 |
| 4.1 | MODEL ACCURACY | | 19 |
| CONCLUSION AND FUTURE SCOPE | | | 20 |
| REFERENCES | | | 21 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Fig 2.1 | Loan Prediction Dataset | 5 |
| Fig 2.2 | Data Processing | 6 |
| Fig 2.3 | Training And Testing | 6 |
| Fig 2.4 | Diagram Of Proposed Methodology | 7 |
| Fig 3.1 | Example Data | 9 |
| Fig 3.2 | Model-1 | 10 |
| Fig 3.3 | Model-2 | 10 |
| Fig 3.4 | Model-3 | 11 |
| Fig 3.5 | Final Model | 11 |
| Fig 3.6 | Boosting Ensemble | 12 |
| Fig 3.7 | Individual weak learner error and Ensemble training and Test error rates | 13 |
| Fig 3.8 | Boosting | 14 |
| Fig 3.9 | Decision Tree With Random Forest | 17 |
| Fig 3.10 | XGBoosting | 18 |
| Fig 4.1 | Accuracy Comparison | 19 |

**CHAPTER 1:**

**INTRODUCTION**

One of the most important factors which affects our country’s economy and the financial condition is the credit system governed by the banks. “As we know credit risk evaluation is very crucial, there are variety of techniques are used for risk level calculation. In addition, credit risk is one of the main functions of the banking community” [4]. “In this paper we have taken the information of past clients of different banks to whom on a bunch of boundaries advance were endorsed. The primary segment of bank's beneficial resource straightforwardly comes from the benefit acquired from the advances being circulated by the bank [4]. Loan Approval Prediction is extremely handy for employee of banks as well as for the applicant also. “It is the responsibility of bank to ensure their assets is in the right hand. By implementation of this system we will be able to predict and ensure that applicant for the loan is safe or not by this automatic loan approval prediction system” [3]. “One of disadvantage of this model is that it emphasizes different weights to each factor but in real life sometime loan can be approved on the basis of single strong factor only, which is not possible through this system” [4]. There may be various benefits that the bank can obtain, such as setting a time limit for applicants to check and ensure whether their loan will be sanctioned or not. This prediction system may be helpful in the sense that, it gives the right to bankers to more focus on valuable assets for the bank not focus on the poor applicants. It will reduce the time for the loan application process of the applicant. “Result against particular Loan Id can be send to various department of banks so that they can take appropriate action on application. This helps all others department to carried out other formalities” [4]. In the next section we have defined the problem statement. After that, there is a brief description of our dataset. In the next section there is a literature survey. In the next, the algorithms used to make the model. then result and analysis and then conclusion.

**1.1 PROBLEM DEFINITION**

Account firms and banks need to automatize the credit qualification activity (continuously) essentially dependent on data given by customers when rounding out an online structure. Gender, Marital Status, Education, Number of Dependents, Salary, Loan Amount, Credit History, and different subtleties are incorporated. To digitize this interaction, they made an issue to group the client sections that can apply for a credit sum, permitting them to focus on these clients explicitly. They have introduced a fractional informational collection for this situation.

Approval of Loan is a very common real-life problem that every company faces in their lending operations. If the loan approval process is automated, it can save a lot of man hours and improve the speed of service to the customers. The increase in customer satisfaction and savings in operational costs are significant” [9]. “However, the rewards can only be realized if the bank has a sturdy model in place to accurately forecast which client's loans it should accept and which it should reject, in order to reduce potential risk” [2]

**1.2 MOTIVATION FOR THE WORK**

Machine learning techniques have been around us and used for analysis for many kinds of data science applications. The major motivation behind this research-based project was to explore the feature selection methods, data preparation and processing behind the training models in the machine learning. With first hand models and libraries, the challenge we face today is data where beside their abundance, and our cooked models, the accuracy we see during training, testing and actual validation has a higher variance. Hence this project is carried out with the motivation to explore behind the models, and further implement ,Ada-boosting with decision Tree and Ada-boosting with Random Forest and Gradient Boosting model to train the obtained data.

**1.3 OBJECTIVES OF THE PROJECT**

The main objectives of developing this project are:

* To develop machine learning model to predict future possibility of loan prediction by implementing Ada-boosting with decision tree and Ada-boosting with Random Forest and Gradient Boosting model.
* To design a machine learning model that can be forecast loan default.
* To train and test the machine learning model to predict loan default.

**CHAPTER 2:**

**SYSTEM ANALYSIS AND DESIGN**

**2.1 EXISTING SYSTEM**

Bank employees manually check the details of the applicant and grant the loan to the eligible applicant. It takes a lot of time to check the details of all the applicants. Artificial neural network model for predicting credit risk of a bank. Feed-forward back propagation neural networks are used to predict credit defaults. A method of combining two or more classifiers together to produce an ensemble model for better prediction. They used bagging and boosting techniques and then random forest techniques. The process of classifier is to improve the performance of the data and it provides better efficiency. In this work, the authors describe various ensemble techniques for binary classification and for multiclass classification. The new technique described by the authors for ensemble is COB which gives effective performance of classification but it also compromises with noise and external data of classification. In the end they concluded that the ensemble based algorithm improves the results of the training data set.

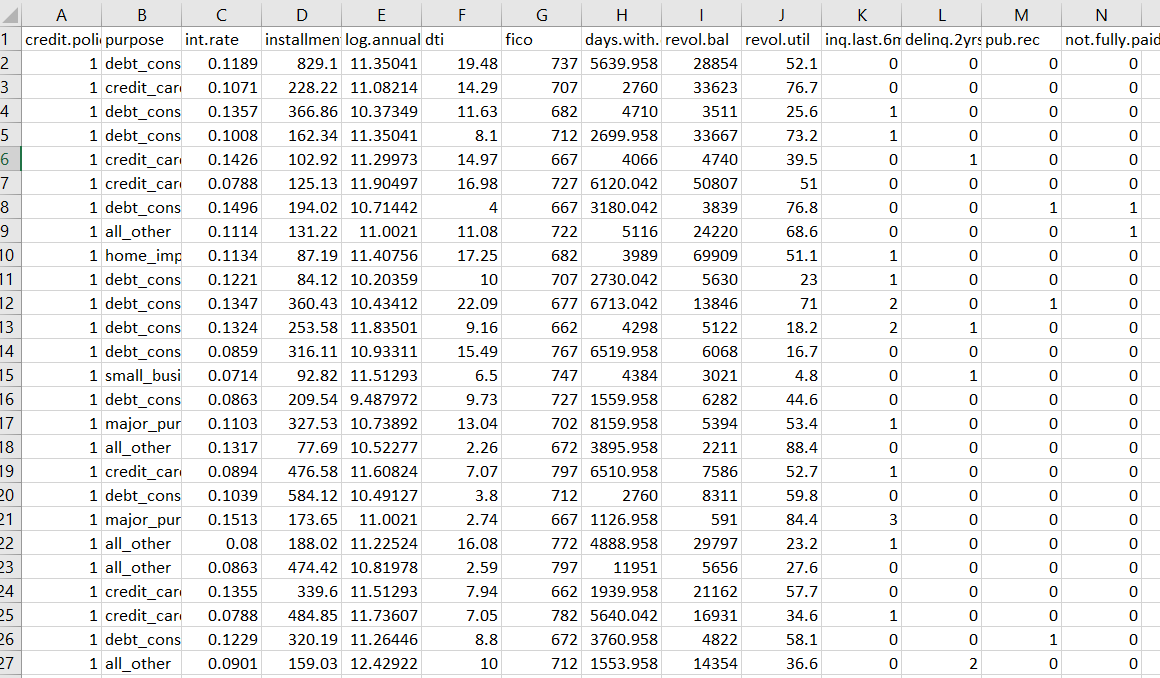
**2.2 PROPOSED SYSTEM**

To deal with the problem, we developed automatic loan prediction using machine learning techniques. We will train the machine with previous dataset. so machine can analyze and understand the process . Then machine will check for eligible applicant and give us result. Advantages Time period for loan sanctioning will be reduced. Whole process will be automated so, human error will be avoided Eligible applicant will be sanctioned loan without any delay.

**2.2.1 COLLECTION OF DATASET**

The dataset collected to predict loan failure customers is described in the training set and tested set. The 80:20 ratio is usually applied to separate the training set and the test set. Data model which was made using the Decision Tree, is mounted on the training set and hung on the Test Tech Finesse,

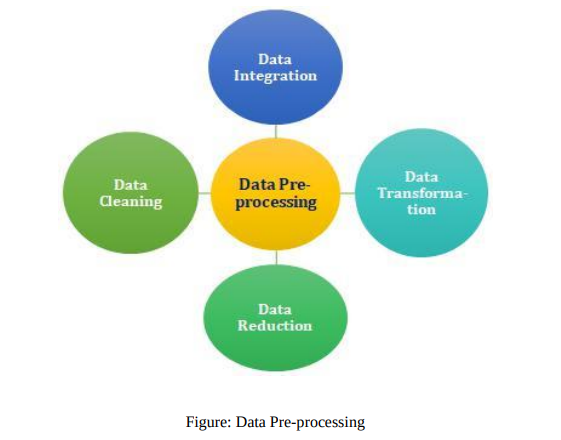
The test set is used to forecast. Has the following properties:



**Fig 2.1: Loan Prediction Dataset**

**2.2.2 DATA PREPROCESSING**

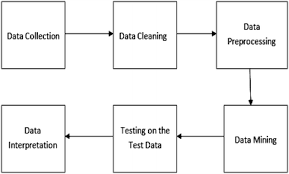
The collected data may contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed and so it'll better the effectiveness of the algorithm. We should remove the outliers and we need to convert the variables. In order to flooring these issues we use chart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



**Fig. 2.2: Data Processing**

**2.2.3 TRAIN MODEL ON TRAINING DATA SET**

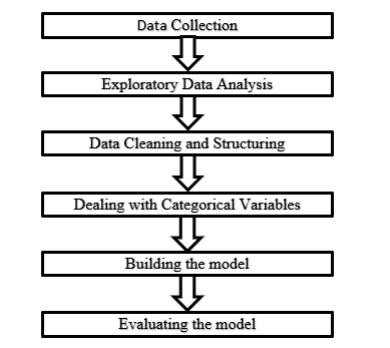
Now we should train the model on the training dataset and make soothsaying for the test dataset. We can divide our train dataset into two tract train and testimony. We can train the model on this training part and using that make soothsaying for the testimony part. In this way, we can validate our soothsaying as we've the true soothsaying for the testimony part (which we don't have for the test dataset)



**Fig 2.3: Training And Testing**

**2.2.4 CORRELATING ATTRIBUTES**

Grounded on the correlation among attributes it was observed more likely to pay back their loans. The attributes that are individual and significant can include Property area, education, loan measure, and originally credit History, which is since by insight it's considered as important. The correlation among attributes can be associated using corplot and boxplot in Python platform.



**Fig 2.4: Diagram Of Proposed Methodology**

**2.3 DESIGN REQUIREMENTS**

For make a loan prediction system we mainly required following requirements**,**

**2.3.1 HARDWARE AND SOFTEARE REQUIREMENTS**

The hardware requirements for the project include a laptop with at least 4GB ram running windows or Linux operating system. The software requirements include a code editor. This project uses Microsoft Visual Studio which is a code editor redefined and optimized for building and debugging modern web and cloud applications.

**2.3.2** **PYTHON LIBRARIES**

The machine learning models are implemented using python version 3.7 on a Jupyter notebook with the listed libraries: numpy, pandas , matplotlib, seaborn , and sklearn.

* **Jupyter notebooks** are a web-based interface in which you can write, visualize, and execute python code in cells. It is good for exploratory analysis and enable to run individual code cells.
* **Numpy** is a Python library that may be used to work with multi-dimensional arrays, linear algebra, the Fourier transform, and matrices.
* **Pandas** is a data manipulation and analysis package written in Python.
* **Matplotlib** is a Python package that allows you to create static, animated, and interactive visualizations.
* **Seaborn** is a matplotlib-based python data visualization package. It has a high-level interface for creating visually appealing and instructive statistics visuals.
* **Sklearn** is a Python toolkit that allows you to create machine learning and statistical models including clustering, classification, and regression.

**CHAPTER 3:**

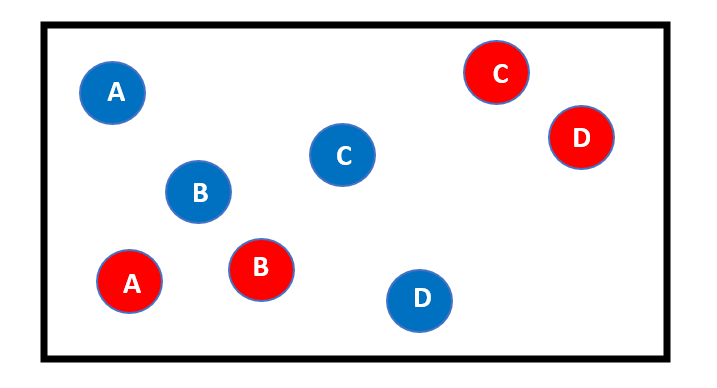
**METHODS AND ALGORITHMS USED**

The main purpose of designing this system is to automate this process by employing machine learning algorithm.

We have used Ada-boosting with Decision Tree and Ada-boosting with Random Forest and XG boosting as a machine-learning algorithm to train our system.

**3.1 BASIC OF ADA-BOOSTING**

Ensemble methods is a machine learning technique that combines several base models in order to produce one final model.

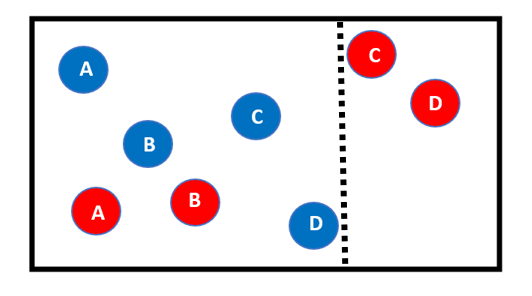


**Fig 3.1: Example Data**

This is my complete data. Here, I have two class blue and red. Now the first step is to build a model to classify this data.

**3.1.1 MODEL-1**

Suppose the 1st model gives the following result, where it is able to classify two red points on the right side(C,D). But the model also miss-classify the two red points here(A,B).

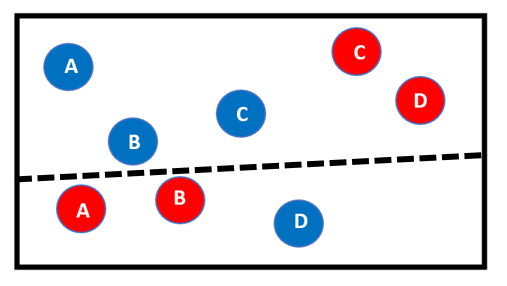


**Fig 3.2: Model-1**

These two red points (A,B) will be given higher weights in the next iteration.

In this model we giving higher weights to previous model miss-classify points means, my model is going to focus more on these points(A,B).

**3.1.2 MODEL-2**



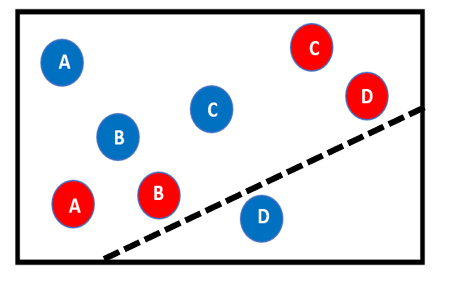
**Fig 3.3: Model-2**

In the 2nd model gives the above result, where it is able to classify two red points (A,B) which are miss-classify in previous model. But the model also miss-classify one blue point(D).

So, this one blue point(D) will be given higher weights in the next iteration.

**3.1.3 MODEL-3**

 In this model we giving higher weights to previous model miss-classify points means, my model is going to focus more on these points(D).

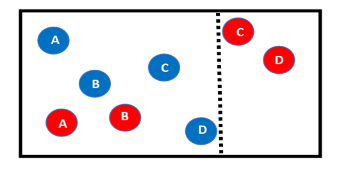
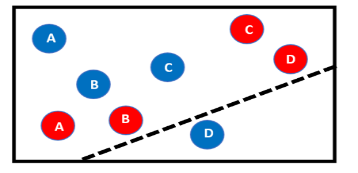
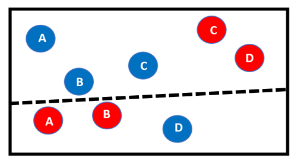
****

**Fig 3.4: Model-3**

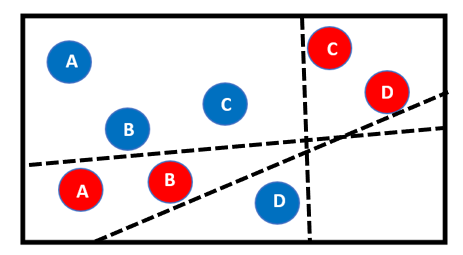
In the 3rd model gives the above result, where it is able to classify one blue points (D) which are miss-classify in previous model.

we can say all these individual models are not strong enough to classify the points correctly and are often called weak learners.

**3.1.4 FINAL MODEL**



Model-1 model-2 model-3



**Fig 3.5: Final Model**

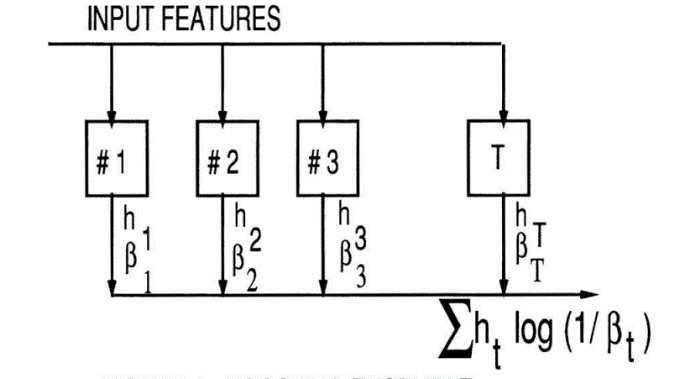
After multiple iterations, we will be able to create the right decision boundary with the help of all the previous weak learners.

As you can see the final model is able to classify all the points correctly.

This final model is known as a strong learner.

**3.2 ADA-BOOSTING WITH DECISION TREE**

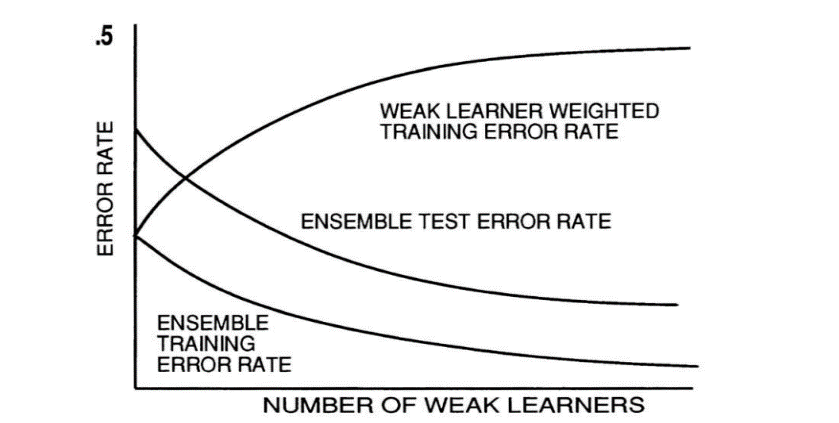
Ada-Boosting is a technique in Machine Learning used as an Ensemble Method. The most common algorithm used with AdaBoost is decision trees with one level that means with Decision trees with only 1 split. In the case of AdaBoost, higher points are assigned to the data points which are incorrectly predicted by the previous model. A new boosting algorithm termed AdaBoost by their inventors (Freund and Schapire,1995) has advantages over the original boosting algorithm (Schapire, 1990) and a second version (Freund, 1990). The implications of a boosting algorithm is that one can take a series of learning machines (termed weak learners) each having a poor error rate (but no worse than .5-y, where y is some small positive number) and combine them to give an ensemble that has very good performance (termed a strong learner). The first practical implementation of boosting was in OCR (Drucker, 1993, 1994) using neural networks as the weak learners. In a series of comparisons (Bottou, 1994) boosting was shown to be superior to other techniques on a large OCR problem. The general configuration of AdaBoost is shown in Figure 3.6. Each box is a decision tree built using Quinlan’s C4.5 algorithm (Quinlan, 1993) The key idea is that each weak learner is trained sequentially. The first weak learner is trained on a set of patterns picked randomly (with replacement) from a training set. After training and pruning, the training patterns are passed through this first decision tree. In the two class case the hypothesis hi is either class 0 or class 1. Some of the patterns will be in error.





**Fig 3.6: Boosting Ensemble**







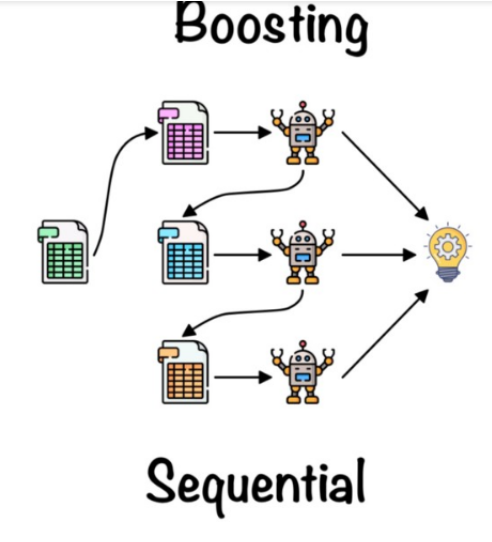
**Fig 3.7: INDIVIDUAL WEAK LEARNER ERROR RATE AND**

**ENSEMBLE TRAINING AND TEST ERROR RATES**

The training set for the second weak learner will consist of patterns picked from the training set with higher probability assigned to those patterns the first weak learner classifies incorrectly. Since patterns are picked with replacement, difficult patterns are more likely to occur multiple times in the training set. Thus as we proceed to build each member of the ensemble, patterns which are more difficult to classify correctly appear more and more likely. The training error rate of an individual weak learner tends to grow as we increase the number of weak learners because each weak learner is asked to classify progressively more difficult patterns. However the boosting algorithm shows us that the ensemble training and test error rate decrease as we increase the number of weak learners. The ensemble output is determined by weighting the hypotheses with the log of (l!βi) where β is proportional to the weak learner error rate. If the weak learner has good error rate performance, it will contribute significantly to the output, because then 1 / β will be large. Figure 3.7 shows the general shape of the curves we would expect. Say we have constructed N weak learners where N is a large number (right hand side of the graph). The N'th weak learner (top curve) will have a training error rate that approaches .5 because it is trained on difficult patterns and can do only sightly better than guessing. The bottom two curves show the test and training error rates of the ensemble using all N weak learners. which decrease as weak learners are added to the ensemble

**3.2.1 BOOSTING**

Boosting arises from the PAC (probably approximately correct) learning model which has as one of its primary interests the efficiency of learning. Schapire was the first one to show that a series of weak learners could be converted to a strong learner. Let us call the set of N 1 distinct examples the original training set. We distinguish the original training set from what we will call the filtered training set which consists of N 1 examples picked with replacement from the original training set. Basically each of N 1 original examples is assigned a weight which is proportional to the probability that the example will appear in the filtered training set (these weights have nothing to do with the weights usually associated with neural networks). Initially all examples are assigned a weight of unity so that all the examples are equally likely to show up in the initial set of training examples. However, the weights are altered at each state of boosting and if the weights are high we may have multiple copies of some of the original examples appearing in the filtered training set. In step three of this algorithm, we calculate what is called the weighted training error and this is the error rate over all the original N 1 training examples weighted by their current respective probabilities. The algorithms terminates if this error rate is .5 (no better than guessing) or zero . Although not called for in the original C4.5 algorithm, we also have an original set of pruning examples which also are assigned weights to form a filtered pruning set and used to prune the classification trees constructed using the filtered training set. It is known (Mingers, 1989a) that reducing the size of the tree (pruning) improves generalization.

****

**Fig 3.8 Boosting**

**3.2.2 DECISION TREES**

For our implementation of decision trees, we have a set of features (attributes) that specifies an example along with their classification (we discuss the two-class problem primarily). We pick a feature that based on some criterion, best splits the examples into two subsets. Each of these two subsets will usually not contain examples of just one class, so we recursively divide the subsets until the final subsets each contain examples of just one class. Thus, each internal node specifies a feature and a value for that feature that determines whether one should take the left or right branch emanating from that node. At terminal nodes, we make the final decision, class 0 or 1. Thus, in decision trees one starts at a root node and progressively traverses the tree from the root node to one of the

Inputs: N I training patterns N 2 pruning patterns. N 3 test patterns

initialize the weight vector of the N I training pattens:

initialize the weight vector of the N 2 pruning patterns:

initialize the number of trees in the ensemble to t = 1

Do until weighted training error rate is 0 or .5 or ensemble test error rate asymptotes

1.For the training set and pruning sets

Pick N 1 samples from original training set with probability P(i) to form filtered training set Pick N 2 samples from original pruning set with probability r(i) to form filtered pruning set

2. Train tree t using filtered training set and prune using filtered pruning set

3. Pass the N 1 original training examples through the pruned tree whose output ht (i) is either 0 or 1 and classification c(i) is either 0 or 1.

Calculate the weighted training error rate,=

4.set

5. Set the new training weight vector to be

Pass the N 2 original pruning pattern through the pruned tree and calculate new pruning weight vector:

6.For each tree t in the ensemble (total trees T), pass the j’th test pattern through and obtain ht(j) for each t. The final hypothesis hf (j) for this pattern:

1,





hf(j)=

0,

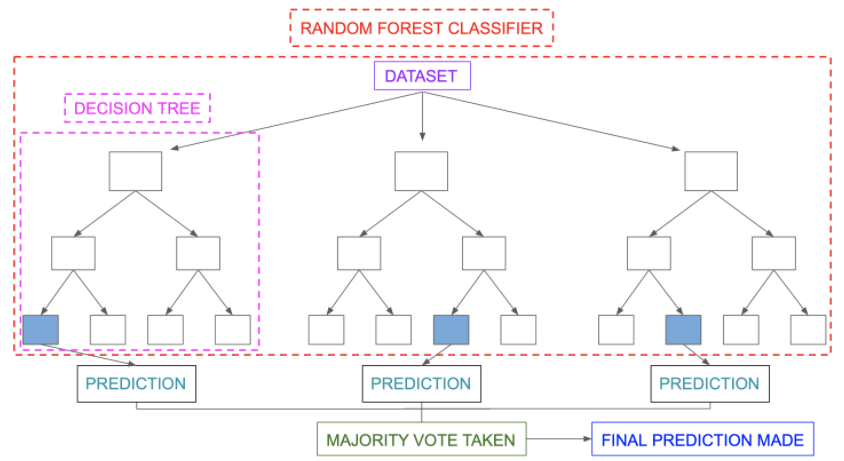
Do for each test pattern and calculate the ensemble test error rate:

7.t=t+1

End until terminal nodes where a final decision is made. CART (Brei man, 1984) and C4.5 (Quinlan 1993) are perhaps the two most popular tree building algorithms. Here, C4.5 is used. The attraction of trees is that the simplest decision tree can be respecified as a series of rules and for certain potential users this is more appealing than a nonlinear "black box" such as a neural network. That is not to say that one can not design trees where the decision at each node depends on some nonlinear combination of features, but this will not be our implementation. Other attractions of decision trees are speed of learning and evaluation. Whether trees are more accurate than other techniques depends on the application domain and the effectiveness of the particular implementation. In OCR, our neural networks are more accurate than trees but the penalty is in training and evaluation times. In other applications which we will discuss later a boosting network of trees is more accurate. As an initial example of the power of boosting, we will use trees for OCR of hand written digits. The main rationale for using OCR applications to evaluate AdaBoost is that we have experience in the use of a competing technology (neural networks) and we have from the National Institute of Standards and Technology (NIST) a large database of 120,000 digits, large enough so we can run multiple experiments. However, we will not claim that trees for OCR have the best error performance. Once the tree is constructed, it is pruned to give hopefully better generalization performance than if the original tree was used. C4.5 uses the original training set for what is called "pessimistic pruning" justified by the fact that there may not be enough extra examples to form a set of pruning examples. However, we prefer to use an independent set of examples to prune this tree. In our case, we have (for each tree in the ensemble) an independent filtered pruning set of examples whose statistical distribution is similar to that of the filtered training set. Since the filtering imposed by the previous members of the ensemble can severely distort the original training distribution, we trust this technique more than pessimistic pruning. In pruning (Mingers, 1989), we pass the pruning set though the tree recording at each node (including non-terminal nodes) how many errors there would be if the tree was terminated there. Then, for each node (except for terminal nodes), we examine the subtree of that node. We then calculate the number of errors that would be obtained if that node would be made a terminal node and compare it to the number of errors at the terminal nodes of that subtree. If the number of errors at the root node of this subtree is less than or equal to that of the subtree, we replace the subtree with that node and make it a terminal node. Pruning tends to substantially reduce the size of the tree, even if the error rates are not substantially decreased.

**3.3 ADA-BOOSTING WITH RANDOM FOREST**

Random forest is a machine learning algorithm that is supervised. The bagging approach is used to train a forest, which is a collection of decision trees. Random forest generates a large number of decision trees and then combines them to get a consistent and accurate classification. The Random Forest algorithm has the advantage of being able to be used for both classification and regression analysis. In machine learning, boosting is an ensemble meta-algorithm for primarily reducing bias, and also variance in supervised learning, and a family of machine learning algorithms that convert weak learners to strong ones. AdaBoost also called Adaptive Boosting is a technique in Machine Learning used as an Ensemble Method. The most common algorithm used with AdaBoost is decision trees with one level that means with Decision trees with only 1 split. These trees are also called Decision Stumps. As we know Random forest is created using a bunch of decision trees which make use of different variables or features and makes use of bagging techniques for data sample. In AdaBoost, the forest is created using a bunch of what is called as decision stump. By using this algorithm we can predict the model with accuracy value.

****

**Fig 3.9: Decision Tree With Random Forest**

**3.4 XGBOOST(Extreme Gradient Boosting)**

Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm.

XGBoost-Extreme Gradient Boosting (Chen and Guestrin 2016) is a scalable and highly efficient boosting system. It has been shown to achieve state-of-the-art results on many machine learning tasks. In XGBoost algorithm unlike the traditional gradient boosting, the process of adding weak learners does not happen sequentially; it approaches this phase in parallel using a multithreaded pattern, thereby resulting in proper utilization of hardware resources leading to greater speed and efficiency. Some important features that make XGBoost more efficient than traditional boosting algorithms are:

1. Sparse aware implementation.

2. Weighted quantile sketch for approximate tree learning.

3. Cache-aware access.

4. Blocks for out-of-core computation.



**Fig 3.10: XGBoosting**

**CHAPTER 4:**

**Result and Discussion**

**4.1 MODEL ACCURACY**

After performing the machine learning approach for testing and training we find that accuracy of the Ada-boosting with decision tree is much efficient as compare to other algorithms. It is concluded that Ada-boosting with decision tree is best among them with 84.8% accuracy when it comes to classification with comparison to the other algorithm’s accuracy obtained and the comparison is shown in (fig 4.1).Other algorithms have accuracy of 84.7% and 84.4% from Ada-boosting with Random Forest and Gradient Boosting respectively.

|  |  |
| --- | --- |
| **Algorithm** | **Accuracy** |
| Ada-boosting with Decision tree | 84.8% |
| Ada-boosting with Random forest | 84.7% |
| Gradient Boosting | 84.4% |

**Fig 4.1 Accuracy Comparison**

**CONCLUSION AND FUTURE SCOPE**

In this report machine learning was used to predict loan acceptance. The prediction method begins with data pre-processing, filling the missing values, experimental data analysis. After evaluating model on test dataset, each of these algorithms obtained a precision rate between 80% and 90%. Although here it can be concluded with certainty that the Ada-boosting with decision tree model is very efficient and produces superior results than other models.

This research explores using machine learning algorithms to improve the accuracy of predicting loan default. The best performing model in the research which is Ada-boosting with decision trees achieves an accuracy of about 84.8%. This is a fair performance and can further be improved through different methods of parameter tuning and feature selection which may possibly yield improvements in the model performance. It may also be beneficial to do a cross validation with other sources of open dataset as they become more accessible to compare the performance of the model. Since the research is also limited to the probability of default in a default state , further exploration may be made in determining the expected return of the loan based on borrower’s characteristics , loan characteristics the recollection of loans processes.

**REFERENCES**

1. Ashwini S. Kadam, Shraddha R Nikam, Ankita A. Aher, Gayatri V. Shelke, Amar S. Chandgude. “Prediction for Loan Approval using Machine Learning Algorithm”, International Research Journal of Engineering and Technology (IRJET), Apr 2021
2. Mohammad Ahmad Sheikh, Amit Kumar Goel,Tapas Kumar. "An Approach for Prediction of Loan Approval using Machine Learning Algorithm", International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020
3. X.FrencisJensy, V.P.Sumathi,Janani Shiva Shri, “An exploratory Data Analysis for Loan Predict ion based on nature of clients”, International Journal of Recent Technology and Engineering (IJRTE),Volume-7 Issue-4S, November 2018
4. J. Tejaswini1, T. Mohana Kavya, R. Devi Naga Ramya, P. Sai Triveni Venkata Rao Maddumala. “ACCURATE LOAN APPROVAL PREDICTION BASED ON MACHINE LEARNING APPROACH” Vol 11, www. jespublication.com, page 523, ISSN NO: 0377-9254, Issue 4 April/ 2020
5. “Prediction for Loan Approval using Machine Learning Algorithm” International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 08 www.irjet.net p-ISSN: 2395-0072 , Issue: 04 | Apr 2021
6. Vaidya, "Predictive and probabilistic approach using logistic regression: Application to prediction of loan approval," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, 2017, pp. 1-6.doi: 10.1109/ICCCNT.2017.8203946
7. M. Bayraktar, M. S. Aktaş, O. Kalıpsız, O. Susuz and S. Bayracı, "Credit risk analysis with classification Restricted Boltzmann Machine," 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, 2018, pp. 1-4.doi: 10.1109/SIU.2018.840 4397
8. Y. Shi and P. Song, "Improvement Research on the Project Loan Evaluation of Commercial Bank Based on the Risk Analysis," 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, 2017, pp. 3-6.doi: 10.1109/ISCID.2017.60
9. V. C. T. Chan et al., "Designing a Credit Approval System Using Web Services, BPEL, and AJAX," 2009 IEEE International Conference on eBusiness Engineering, Macau, 2009, pp. 287- 294.doi: 10.1109/ICEBE.2009.46