**Mini Project Report on**



 **Home Loan Prediction Using Machine Learning**

**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name:**  **University Roll No.**

**Raj Singh Patel 2017530**

***Under the Mentorship of***

**Dr. Vikas Tripathi**

**Assistant Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January 2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Home Loan Prediction Using Machine Learning”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr. Vikas Tripathi, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name: - Raj Singh Patel University Roll no: -2017530 **\**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1** |
| Chapter 2 | Literature Survey | **2** |
| Chapter 3 | Methodology | **3-6** |
| Chapter 4 | Result and Conclusion | **7** |
| Chapter 5 | Conclusion and Future Work | **8** |
|  | References |  |

**Chapter 1**

**Introduction**

A loan is the core business part of banks. The main portion the bank’s profit is directly come from the profit earned from the loans. Though bank approves loan after a regress process of verification and testimonial but still there's no surety whether the chosen hopeful is the right hopeful or not. This process takes fresh time while doing it manually. We can prophesy whether that particular hopeful is safe or not and the whole process of testimonial is automated by machine literacy style. Loan Prognostic is really helpful for retainer of banks as well as for the hopeful also.

**Why do we need Loan Prediction Model?**

Generally, loan prediction involves the lender looking at various background information about the applicant and deciding whether the bank should grant the loan. Parameters like credit score, loan amount, lifestyle, career, and assets are the deciding factors in getting the loan approved. If, in the past, people with parameters similar to yours have paid their dues timely, it is more likely that your loan would be granted as well.

Machine learning algorithms can exploit this dependency on past experiences and comparisons with other applicants and formulate a data science problem to predict the loan status of a new applicant using similar rules.

Several collections of data from past loan applicants use different features to decide the loan status. A machine learning model can look at this data, which could be static or time-series, and give a probability estimate of whether this loan will be approved.

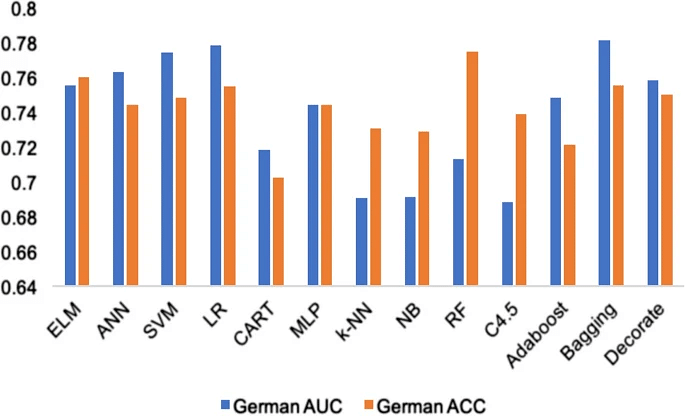
**Chapter 2**

**Literature Survey**

Historically, lending risk prediction has used statistical methods, including Linear Discriminant Analysis and Logistic Regression. However, with large credit datasets, ML-driven risk estimation algorithms like k-Nearest Neighbor, Random Forest, and Support Vector Machines are better at capturing complex relationships. Moreover, deep learning methods have gained a particular advantage in modeling non-linear relationships between risk and risk factors for large-scale lending risk and loan prediction datasets.

Novel frameworks like DEAL (Deep Ensemble Algorithm), or improvements over existing models of Recurrent Neural Networks (RNN) or Boosted Decision Tree or Autoencoders, give satisfactory accuracy over large datasets and generate features with domain expertise.

However, more work is available on machine learning models than deep learning architectures since the latter's performances are often specific to the dataset they were designed and tested on. In the figure shown below, a recent paper compares the performance of various machine learning algorithms on the German credit risk dataset. We can see that algorithms like SVM, Random forest, and the Logistic regression model perform better than ELM and ANN. However, decision trees and boosting also give a competitive performance on this dataset.



**Fig. 2.1**

**Chapter 3**

**Methodology**

**Introduction to Machine Learning:**

Machine learning is a field of artificial intelligence that focuses on the development of algorithms and models that can learn from data without being explicitly programmed. It has become an increasingly important part of many areas of technology, including image and speech recognition, natural language processing, and finance.

There are several types of machine learning, including:

1. Supervised learning: In supervised learning, a model is trained on a labeled dataset, where the correct output is provided for each example in the training set. The goal is to make predictions on new, unseen examples based on the patterns learned from the training data.
2. Unsupervised learning: In unsupervised learning, the model is not provided with labeled training examples. Instead, it must discover the underlying structure of the data through techniques like clustering.
3. Semi-supervised learning: Semi-supervised learning is a type of machine learning that combines both supervised and unsupervised learning. It is often used when a large amount of unlabeled data is available, but it is too expensive or time-consuming to label all of it.
4. Reinforcement learning: In reinforcement learning, an agent learns to interact with its environment in order to maximize a reward. The agent receives feedback in the form of rewards and punishments as it takes actions in the environment.

**Tools Used:**

**Google Colaboratory: -** The Colaboratory at Google Research, or "Colab" for short, is a product. Colab is a Python editor for the web that allows anyone to develop and run Python programmes. It's useful in fields like machine learning, data analysis, and teaching. Colab is a hosted Jupyter notebook service that doesn't require installation and gives you free access to computing resources like GPUs.

The availability of Colab resources is neither guaranteed or infinite, and usage limits are subject to change at any time. This is essential in order for Colab to provide free materials.

**Python: -** The language used to develop this project is Python. Python offers readable and concise codes. Since machine learning and artificial intelligence involve complex algorithms, the simplicity of Python adds value and enables the creation of reliable systems.

**Streamlit: -** Streamlit was used for building the web page. Streamlit is an open source app framework in Python language. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc**.**

**Machine Learning Algorithm used:**

**Random Forest:**

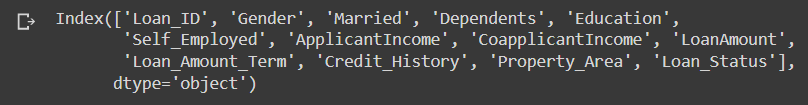
A Random Forest Algorithm is a supervised machine learning algorithm which is extremely popular and is used for Classification and Regression problems in Machine Learning. We know that a forest comprises numerous trees, and the more trees more it will be robust. Similarly, the greater the number of trees in a Random Forest Algorithm, the higher its accuracy and problem-solving ability.  Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. It is based on the concept of ensemble learning which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model.

**XGBoost:**

XGBoost is an implementation of Gradient Boosted decision trees. In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.

**Dataset Used:**

The loan eligible dataset was taken from Kaggle which contain following features:



**Fig. 3.1**

**Building The Model:**

**Data Preprocessing:**

Data preprocessing is a step in the data mining and data analysis process that takes raw data and transforms it into a format that can be understood and analyzed by computers and machine learning.

Raw, real-world data in the form of text, images, video, etc., is messy. Not only may it contain errors and inconsistencies, but it is often incomplete, and doesn’t have a regular, uniform design.

Machines like to process nice and tidy information – they read data as 1s and 0s. So calculating structured data, like whole numbers and percentages is easy. However, unstructured data, in the form of text and images must first be cleaned and formatted before analysis.

Timeline

Description automatically generated

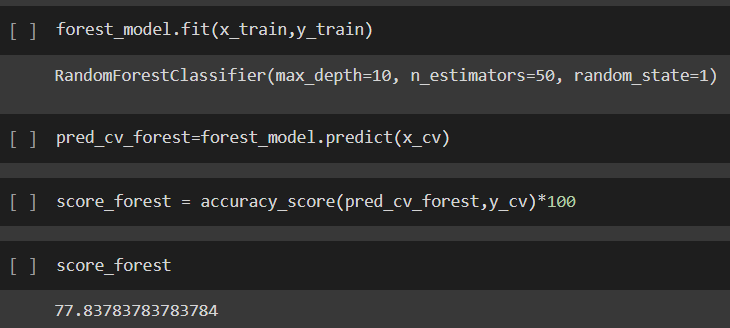
**Fig. 3.2**

**Training the Model:**

With the help of normalized data the model was first built using Random Forest Algorithm and later the performance was improved using XGBoost algorithm.

Using Random Forest:

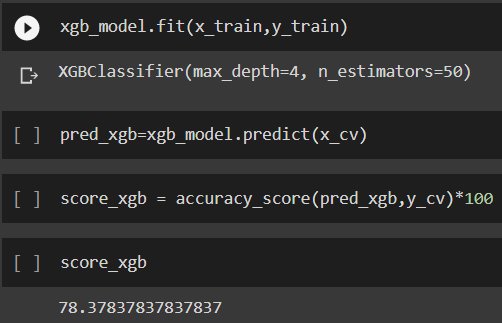
An accuracy of 77.83% was achieved using Random Forest Algorithm.



**Fig. 3.3**

Using XGBoost Classifier:

An increase in performance was observed when the model was trained using XGBoost Algorithm. An accuracy of 78.38% was achieved.



**Fig. 3.4**

**Chapter 4**

**Result and Discussion**

Although both the algorithms were close to each other in terms of performance but the XGBoost Classifier (Accuracy: 78.38%) performed slightly well than the Random Forest Classifier (77.83%).

**Why XGBoost is preferred over Random Forest?**

* Random Forest and XGBoost are decision tree algorithms where the training data is taken in a different manner. XGBoost trains specifically the gradient boost data and gradient boost decision trees. The training methods used by both algorithms is different. We can use XGBoost to train the Random Forest algorithm if it has high gradient data or we can use Random Forest algorithm to train XGBoost for its specific decision trees. Also, we can take samples of data if the training data is huge and if the data is very less, we can use the entire training data to know the gradient of the same.
* XGBoost helps in numerical optimization where the loss function of the data is minimized with the help of weak learners so that iteration happens in the local function in a differentiable manner. Sample is not modified here but different levels of importance are given to each feature in the data. Random Forest is mostly a bagging technique where various subsets are considered and an average of each subset is calculated. Either random subset of features or bootstrap samples of data is taken for each experiment in the data.
* Several hyperparameters are involved while calculating the result using XGBoost. Some include regularization rate, subsample, minimum weights, maximum depths, and learning rates. Though XGBoost is noted for better performance and high speed, these hyperparameters always stop developers from looking into this algorithm. Hyperparameters are not needed in Random Forest and developers can easily understand and visualize Random Forest algorithm with few parameters present in the data.

At last the model Predicted in terms of 0 and 1 whether the customer is eligible for loan or not. Sreamlit was used to build the web wage for user interface.

**Chapter 5**

**Conclusion and Future Work**

From a proper analysis of positive points and constraints on the member, it can be safely concluded that the product is a considerably productive member. This use is working duly and meeting to all Banker requisites. This member can be freely plugged in numerous other systems. There have been mathematics cases of computer glitches, violations in content and most important weight of features is fixed in automated prophecy system, so in the near future the so – called software could be made more secure, trustworthy and dynamic weight conformation. In near future this module of prophecy can be integrated with the module of automated processing system

Further improvement may be done in this using Neural Networks. The system is trained on old training dataset in future software can be made resembling that new testing date should also take part in training data after some fix time.

**References:**

[1] Loan Prediction Dataset. https://www.kaggle.com/datasets/altruistdelhite04/loan-prediction-problem-dataset

[2] XGBoost Tutotial from towardsdatascience. https://towardsdatascience.com/a-beginners-guide-to-xgboost-87f5d4c30ed7

[3] Geeks For Geeks and Youtube support for stremlit. https://www.geeksforgeeks.org/a-beginners-guide-to-streamlit/