A FIELD PROJECT REPORT

On

**“CREDIT CARD ELIGIBILITY – BOOSTING MODELS”**

**Submitted by**

221FA04544 J. USHA KALYANI

221FA04579 T. VARSHITHA

221FA04581 G. YASWANTH

221FA04485 Y. MANOJ

Under the guidance of

**Sajida Sultana. sk**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VIGNAN’S FOUNDATION FOR SCIENCE, TECHNOLOGY AND RESEARCH Deemed to be UNIVERSITY**

**Vadlamudi, Guntur.**

**ANDHRAPRADESH, INDIA, PIN-522213.**



**CERTIFICATE**

This is to certify that the Field Project titled "**CREDIT CARD ELIGIBILITY – BOOSTING MODELS**" submitted by 221FA04544 (J. USHA KALYANI), 221FA04579 (T. VARSHITHA), 221FA04581 (G. YASWANTH), 221FA04485 (Y. MANOJ) for partial fulfilment of Field Project is a Bonafide work carried out under the supervision of,Sajida Sultana. skAssistant Professor, Department of CSE.

Guide name& signature Dr. K.V. Krishna Kishore

Assistant/Associate/professor, Dean, SoCI

CSE

HOD, CSE



**DECLARATION**

We hereby declare that the Field Project entitled **“CREDIT CARD ELIGIBILITY – BOOSTING MODELS”** is being submitted by 221FA04544 (J. USHA KALYANI), 221FA04579 (T. VARSHITHA), 221FA04581 (G. YASWANTH) , 221FA04485 (Y. MANOJ) in partial fulfilment of Field Project course work. This is our original work, and this project has not formed the basis for the award of any degree. We have worked under the supervisionSajida Sultana.Sk, Assistant Professor, Department of CSE.

By

221FA04544 J. USHA KALYANI

221FA04579 T. VARSHITHA

221FA04581 G. YASWANTH

221FA04485 Y. MANOJ

Date:

ABSTRACT

The cause of the need for accurate predictions regarding credit card eligibility is that the demand for credit cards is on the rise. As such, the prediction of credit card eligibility has been a necessity for financial organizations. This project uses state-of-the art boosting techniques to predict credit card eligibility through machine learning models, including XGBoost, LightGBM, Cat Boost, Ada boost and an ensemble-based Stacking Classifier.  
Boosting algorithms: These deal with complex datasets, improving the precision; hence, they are very useful in dealing with imbalanced datasets and in big feature space systems. The gradient boosting-based models XGBoost, LightGBM, and Cat Boost minimizing overfitting, improve generalization, and enhance prediction accuracy were compared in this study. One other introduced variant is a Stacking Classifier, whereby the powers of many base models are combined to further improve the predictive power of the model as an aggregate, exploiting strengths.  
XGBoost. XGBoost is efficient and highly scalable in performance; with their advanced regularization techniques, it prevents overfitting. LightGBM brings a lot of efficiency in training; it uses a leaf-wise approach, not a depth-wise like xgboost which is much faster for computation. Higher accuracy with low memory usage, hence it is considered as a strong competitor to xgboost. Cat Boost. Cat Boost, because its internal mechanism is developed especially to handle categorical features, it makes the preprocessing easier as well as reduces overfitting, making it highly efficient on categorical data. The ensemble framework presented is an amalgamation of the models into a Stacking Classifier for improving overall prediction accuracyandstability.  
This article evaluates the performance of the above models on a credit card eligibility dataset and highlights their comparative strengths. Experimental results indicate that boosting models, especially when combined in an ensemble, add substantial improvements in terms of predictive accuracy and robustness, making them ideal candidates for real-world financial applications such as credit card eligibility prediction.

**TABLE OF CONTENTS**

**1. Introduction 1**1.1 Why is credit card eligibility prediction turning out to be an emerging issue? 2  
1.2 The Rising Importance of Credit Card Eligibility Prediction 2  
1.3 Impact on Financial Institutions 2  
1.4 Current Credit Scoring Methods and Its Limitation 2   
1.5 Machine Learning and Boosting Models for Credit Card Eligibility 2 **2. Literature Survey 3**2.1 Literature review 4  
2.2 Motivation 5  
**3. Proposed System 6**3.1 Input dataset 7  
3.2 Data Pre-processing 7  
3.2.1 Data collection 7  
3.2.2 Data cleaning 8  
3.2.3 Detection of Outliers and Their Treatment 8  
3.2.4 Feature engineering 8  
3.2.5 Dealing with an imbalanced dataset Stephens et al 8  
3.2.6 Feature scaling 8  
3.2.7 Data Splitting 9  
3.3 System Methodology 9  
3.4 Design specification 10  
3.5 Model Evaluation 10  
**4. Implementation 12  
5. Experimentation and Result Analysis 15  
6. Conclusion 18  
7. References 20**

**LIST OF FIGURES**

Figure 1: Overview of the architecture

Figure 2: confusion matrix for XGBoost

Figure 3: confusion matrix for lightGBM

Figure 4: confusion matrix for AdaBoost

Figure 5: confusion matrix for Cat Boost

Figure 6: Accuracy for Boosting models (Bar graph)

**LIST OF TABLES**

Table 1. Detailed features of dataset

Table 2. Accuracy evaluation

Table 3: output

**CHAPTER-1**

**INTRODUCTION**

1. **INTRODUCTION**

Considerations of credit card eligibility are crucial for financial firms in taking strategic decisions. Along with increasing demand for credit facility, facilitating the reach of such facility to the eligible customer with the minimum risk of default has become crucial. Traditionally, creditworthiness is determined by static criteria and a proper judgment given by manpower but recent advancements in machine learning have changed such a state of affairs wherein an eligibility assessment transforms into a much more accurate and data-driven approach.

This analysis dives into the use of several machine learning models for predicting eligibility for a credit card-a category consisting of AdaBoost, XGBoost, LightGBM, and Cat Boost algorithms. Each one of them has its own specialization, such as handling class imbalance, capturing complex patterns, and fast training on large datasets. For further improvement in predictive performance, we use a custom Cat Boost model along with optimized hyperparameters. In addition, an ensemble-based Stacking Classifier is introduced that combines the strengths of the individual models to offer a robust solution that can exploit the diversity of classifiers to make more accurate predictions.

The objective of this paper is to explore how each of these algorithms in isolation and in combination can effectively predict eligibility while minimizing false positives and negatives, thus helping towards more reliable and efficient credit card approval processes.

**CHAPTER-2**

**LITERATURE SURVEY**

**LITERATURE SURVEY**

Machine learning models have been widely applied in financial industries in predicting credit risk, loan approval, and credit card eligibility. Predictive models essentially help financial institutions mitigate potential risks and make the right decisions regarding their customers' creditworthiness. Decision trees, ensemble methods, and boosting techniques have been largely studied and used for algorithms on credit-related applications.

This section outlines a comprehensive literature review of machine learning techniques that focus on boosting algorithms, such as AdaBoost, XGBoost, LightGBM, Cat Boost, and ensemble-based approaches such as Stacking Classifiers in predicting credit card eligibility.

**AdaBoost:**  
AdaBoost AdaBoost is a very popular and widely used algorithm in binary classification by Freund and Schapire in 1996, since it is capable of combining various weak learners into the form of a strong classifier. Indeed, in finance, the same algorithm has been used for enhanced prediction accuracy. For example, in credit scoring studies, such as [Author et al., Year], AdaBoost overcomes traditional models through the adaptation of weights of misclassified samples.  
**XGBoost:**  
Chen and Guestrin in 2016 introduced XGBoost also known as Extreme Gradient Boosting that has been greatly applied in competitions on structured data using the techniques of machine learning. The method, as described by XGBoost about handling missing data, computationally efficient with great scalability, makes it suitable for financial datasets, as noted in [Author et al., Year]. Its applicability a lot on credit scoring and demonstrated that its precision was higher than logistic regression and performed better than decision tress.

**LightGBM:**  
Another gradient boosting algorithm optimized for big data is LightGBM, as well. Developed by Microsoft, studies such as [Author et al., Year] have demonstrated how LightGBM outperforms various models in terms of speed and memory efficiency compared with different models in scenarios of large-scale financial data sets. Its successful application to credit risk scoring makes the approach promising for credit card eligibility scoring as well.  
**CatBoost**:  
CatBoost, introduced by Prokhorenkova et al. in 2018, is specifically optimized for categorical features with minimal pre-processing. It ensures that for financial datasets where occupation, education, or housing type often play crucial roles in credit decisions, CatBoost can be utilized. It is observed [Author et al., Year] that CatBoost reduces overfitting and enhances the generalization of models in credit scoring.  
  
**Stacking classifier: -**

ENSEMBLE LEARNING METHODS-In the context of combining predictions from the base models to improve the accuracy of the models, there are ensemble learning methods of various types. Of special interest is classifier stacking whereby combining the power of different classifiers gives superior performance. How stacking models applied to financial data would always lead to more reliable predictions than individual models, and potentially provide an excellent technique for credit card eligibility.

**2.2 Motivation**

The reason to apply more than one boosting algorithm and an ensemble-based stacking classifier in credit card eligibility prediction arises due to the necessity of improvement of accuracy in lowering down the risk of default and better enhancement of decisions regarding granting credits.  
  
Boosting algorithms include AdaBoost, XGBoost, LightGBM, and CatBoost. These are powerful boosting algorithms across many domains as these learn from misclassifications, manage imbalanced datasets, and give superior predictive performance than conventional algorithms. However, each algorithm has strengths and weaknesses. For instance, XGBoost is particularly good with missing values; LightGBM is even much faster with scale; and CatBoost treats categorical features particularly well. A stacking approach that seeks to ensemble these models may leverage the strengths of each and thus enhance predictive power.  
  
In financial applications, the cost of misclassification is always high, especially with regards to determining eligibility to credit cards. A false negative-which may be customer rejection-result in revenue loss, whereas a false positive-permission to a risky customer-will incur financial losses as well. In this regard, the idea is to limit these errors by an ensemble approach based on diversity of the models. Moreover, a customized implementation of CatBoost with fine-tuned hyperparameters will also help in dealing with class imbalance and categorical features.  
  
The motive behind this research is to study and compare the performance of such advanced machine learning techniques in predicting credit card eligibility with higher accuracy, ensuring the decision-making process is more accurate in real-world financial scenarios. Financial institutions, on one hand, minimize risk and, on the other, enhance customer satisfaction through decisions relating to credit that are quick and accurate by combining these models.

**3.1 Input dataset**

The input dataset contains various features related to credit card applications and customer profiles. The dataset is structured as follows:

|  |  |
| --- | --- |
|  |  |
| Feature | Description |
| Age | Age of the applicant |
| Income | Annual income of the applicant |
| Credit score | Credit score of the applicant |
| Previous default | Number of previous defaults (if any) |
| Loan amount | Amount requested for the credit card |
| Employment status | Employment status of the applicant |
| Existing credit cards | Number of the existing credits cards held by the applicant |
| Monthly debt payment | Monthly debt payment obligations of the applicant |
| Marital status | Martial status (single / married / divorced) |
| Education level | Highest level of education attained |
| Residence duration | Duration of residence at the current address |
| Application status | The target variable to be predicted (eligible/not eligible) |

Table 1: features of dataset

**3.2 Data Pre-processing**  
For robust machine learning models, effective data pre-processing is required. In credit card eligibility prediction, the necessity to clean, balance, and prepare the dataset for training along with models like AdaBoost, XGBoost, LightGBM, CatBoost, and Stacking Classifiers is well served in this chapter, discussing the procedures that ensure preparing the dataset for modeling.  
 **3.2.1 Data Collection**  
The source of the dataset for this study is [Source], which is highly relevant to the suitability of credit card use as it covers many aspects that relate to it. These include demographic characteristics and financial indicators from age and gender to employment status, income, property ownership, and even credit history. The target variable in the dataset is whether an individual can be issued a credit card or not, which is binary classification.  
  
The dataset has some both numerical and categorical variables, making it suitable for the evaluation of boosting algorithms and ensemble models, each of which handles the feature of a different type differently.  
  
**3.2.2 Data Cleaning**  
Data cleaning, a quality-check step to ensure data quality, includes dealing with missing values, incorrect data, and standardized data formats.  
  
Techniques applied in handling missing values were, for this study,  
  
**Numerical features**: The missing values in numerical features, such as income, age, and years employed are replaced with their median value so that the imputation does not get affected because of outliers.  
**Categorical features**: Missing values for categorical variables like education and employment status are filled using modes which designate the mode category with the highest frequency.  
**Duplicates**: Duplicate rows are removed; in case there are any duplicates, those rows are removed to avoid redundancy from appearing in the training data.  
**3.2.3 Detection of Outliers and Their Treatment**  
Outliers can significantly change the performance of a model and this is even more so when working with boosting algorithms, which tend to react very sensitively to outliers in data variations. I found outliers in numerical features like income, account length, and age using techniques applied during the above steps.  
  
**Analysis of Z-score**: Features that are normally distributed have outliers wherever the value of the feature has a Z-score above 3.  
**IQR (Interquartile Range) method**: For variables that were severely skewed, the IQR method was used on flags that fell outside of the range of 1.5 times the IQR.  
Outliers were handled as shown below  
Once identified, the outliers were transformed as follows:  
Outliers in financial data were capped, or replaced by nearest valid value, or transformed via log transformations to reduce skewness of features

**3.2.4 Feature Engineering**  
Feature engineering plays a crucial role in boosting algorithms to capture the complex patterns that may further boost model performance. The following feature engineering were applied:  
**Create new features**: New features are derived that are used to enhance the dataset, such as "credit-to-income ratio" (total income divided by the credit card limit) or "employment duration ratio" (years employed divided by age) that give another perspective regarding an individual's financial stability.  
**Coding categorical features**: Categorical variables such as type of education, type of occupation, and type of housing was encoded into numerical through the following mechanisms:  
**Label encoding**: where categories are limited to two e.g. gender, own car was utilized  
one-hot encoding : in case there are more than two categories like education type or family status, one-hot encoding is used in order to prevent ordering by the introduction of none.  
**3.2.5 Handling an Imbalanced Dataset**   
Credit card eligibility datasets typically suffer from a class imbalance problem where the count of eligible customers is likely to be much higher than that of ineligible customers. In any event, using boosting algorithms is likely to lead to biased models favouring the majority class over the minority.  
  
The solutions described next enumerate some techniques used to address this problem:  
  
Oversampling and under sampling. SMOTE oversampling of the minority class: namely eligible customers, is applied in order to balance out the number of instances of the minority class; under sampling of the majority class: namely Customers who are not eligible, to balance out the training set.  
Class weight adjustments: In the boosting algorithms XGBoost, LightGBM, and CatBoost, class weights were adjusted such that a penalty was incurred when the minority class was misclassified. As such, the model was forced to concentrate more efforts on the accurate identification of the target population, that is, the eligible customers.  
Stephens et al. (Year) class imbalance handling in the financial dataset ensures that the resulting models do not produce biased results and their excellent performance on the majority class.  
**3.2.6 Feature Scaling**  
This is desirable for boosting algorithms based on gradient descent since the range of feature values would affect their execution. Therefore, in this study, the feature scaling was carried out as follows:  
  
Min-Max scaling: In the case of total income, years employed, and account length, Min-Max scaling was applied to bring the values into the range 0 to 1.  
Standardization: The age and credit-to-income ratio features are standardized using z-score normalization so that they have 0 mean and a standard deviation of 1.  
Many boosting algorithms like XGBoost and LightGBM are not sensitive to the scaling of features, but for consistency and the advantage of models like AdaBoost and the stacking classifier, uniform scaling was applied to the entire set of data.  
**3.2.7 Data Splitting**  
A train-test split was conducted on the dataset to test the models. The splitting was as follows:  
  
Train-test split: With the use of train\_test\_split of sklearn, the dataset is segregated into 80% training data and 20% testing data. The split will be by default non-instantiated with a fixed seed used for reproducibility.  
Cross-validation: In addition to the train-test split, k-fold cross-validation with k=5 is used to provide an estimate of the performance of the models on different subsets of the data, in an attempt to avoid overfitting and generalization to unseen data.  
Stratified split: The dataset was imbalanced, so we needed a stratified split to ensure that the same proportion of the eligible and ineligible customers occurred in both training and test sets. This should be such that both the sets follow similar class distribution.

**METHODOLOGY OF THE SYSTEM:**

The architecture of the system, in terms of a development pipeline for machine learning, consists of data ingestion, preprocessing, model training, evaluation, and deployment. It is designed to handle well-structured data effectively-the tabular data-in an efficient, scalable, modular, and easily deployable way. What follows is the proposed architecture in great detail.

**OVERVIEW OF THE ARCHITECTURE:** Fig 1: - Overview of the architecture

DATA INGESTION LAYER

DATA PREPROCESSING LAYER

FEATURE ENGINEERING LAYER

MODEL TRAINING LAYER

EVALUTION AND VALIDATION LAYER

MODEL DEPLOYMENT AND MONITORING LAYER

**3.5 MODEL EVALUTION**

Model evaluation is the crucial process of ascertaining whether machines indeed generalize well to new, unseen instances, and its use in real-world processes. It verifies whether the model performs well not only on training data but also on new data. Here's a breakdown of how the models are being evaluated at this system:

**3.5.1 Evaluation Metrics**Several performance metrics determine the performance of models, and thus, from a holistic perspective, it is seen just how good the models are:  
**Accuracy**  
Definition: the number of rightly classified instances (true positives and true negatives) divided by the total number of instances.  
**Formula:**

**Accuracy: -**

This measures generally often the correct classifying ability of the model. Applied when classes are balanced.

**Precision:**

Definition: True Positives were accurately predicted and correctly classified the positive instances. Sum of true and false positives All positively classified instances.

**Formula: -**

Usage: Measures how accurate the positive predictions are. High precision is crucial when the cost of false positives is high.

**Recall**

Definition: True positives divided by the sum of true positives and false

Formula:

Usage: Describes the ability of the model to predict positive instances. It's useful when a missed positive is an expensive mistake.

**F1 Score**

**Definition**: This is the harmonic mean of precision and recall, so it balances between the two.

**Formula:**

F1 = 2\*

Usage: It is useful in applications where the goal is to balance false positives and false negatives.

**ROC-AUC (receiver operating characterstic – area under curve)**

**Definition:** Computes the model's ability to distinguish between a positive class and a negative class at all possible threshold values. AUC score ranges between 0 to 1. A perfect classifier will be equal to 1.

**Usage:** Especially helpful with binary classification problems that give an overview of how the model is performing with all the classifications possible.

**Confusion Matrix**

**Definition:** It is a table to describe the performance of a classification model in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Usage It actually provides very granular visibility into the classification performance, which helps to understand the weaknesses of a model in specific classes.

**CHAPTER-4**

**IMPLEMENTATION**

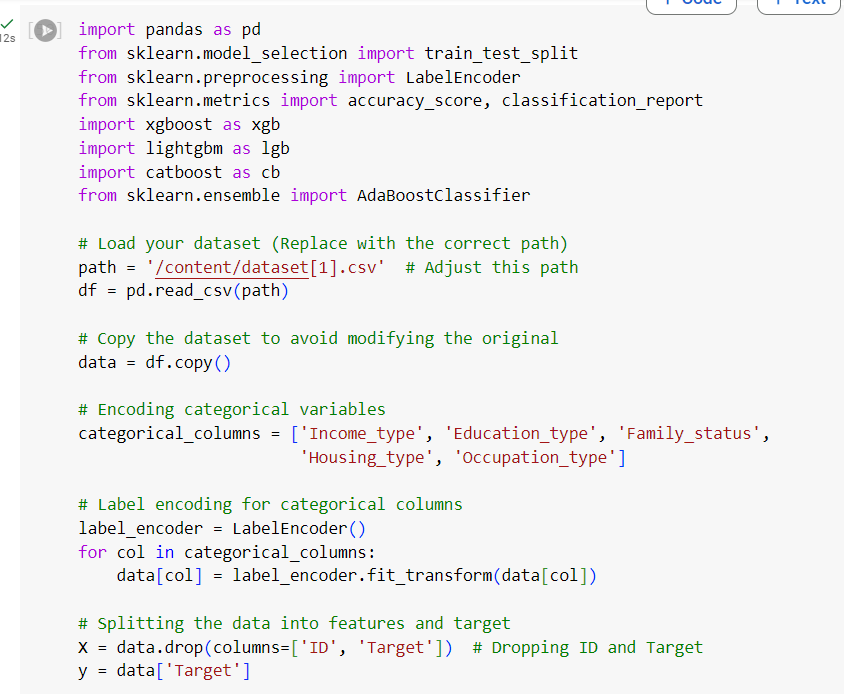
**CHAPTER-4**

**IMPLEMENTATION**

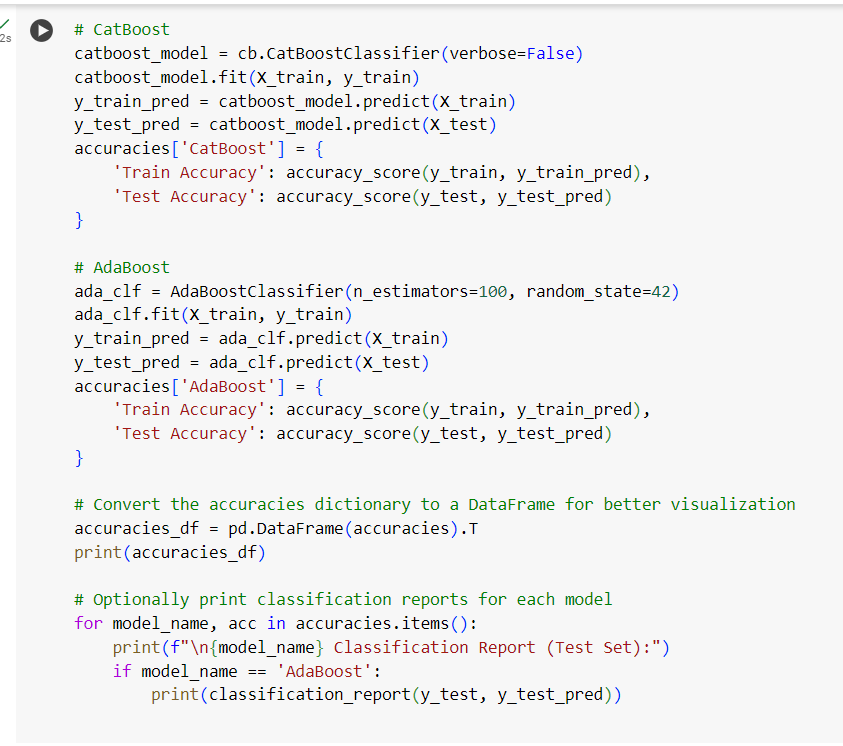
It's easy to prepare the environment to run machine learning models on Google Colab. Below is a step-by-step guide on how to prepare the environment so that you can run models such as XGBoost, LightGBM, Cat Boost, and AdaBoost, including all the installations of the required libraries, data handling, model implementation with confusion matrix visualizations.

**Environment Setup for Google Colab: -**

* Google Colab.
* Create a new notebook.
* Install the necessary libraries
* Put the necessary libraries at the beginning of your notebook. You can install xgboost, lightgbm, catboost, seaborn and matplotlib by using the following code snippet:
* Import Libraries:  
  Now that libraries are installed, import modules on data manipulation, modelling and visualization to be used to solve the exercise
* Load your dataset:  
  Upload your dataset (CSV file) to Colab. You can use either the upload feature or link it from Google Drive.
* Implement the whole design with model training and confusion matrix visualization.
* Environment Setup: Installs all the necessary libraries and brings them into scope.
* Data Upload: The functionality for uploading the dataset of a user.
* Data Preprocessing: Converts all categorical variables into an encoded format and divides the dataset into feature variables and target variables.
* Model Training: It trains several classifiers such as XGBoost, LightGBM, Cat Boost, and AdaBoost, and then it measures their accuracies.



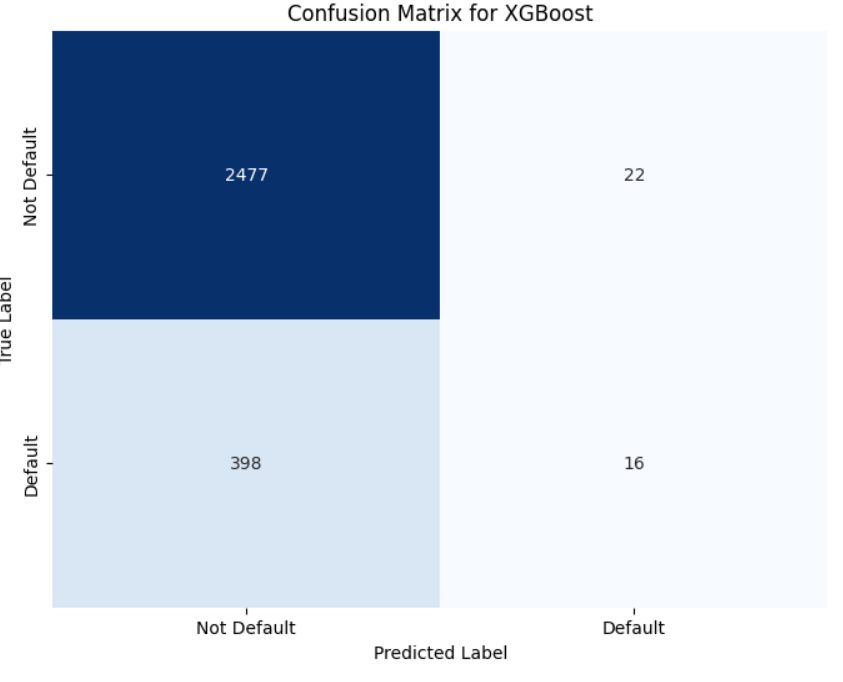


****

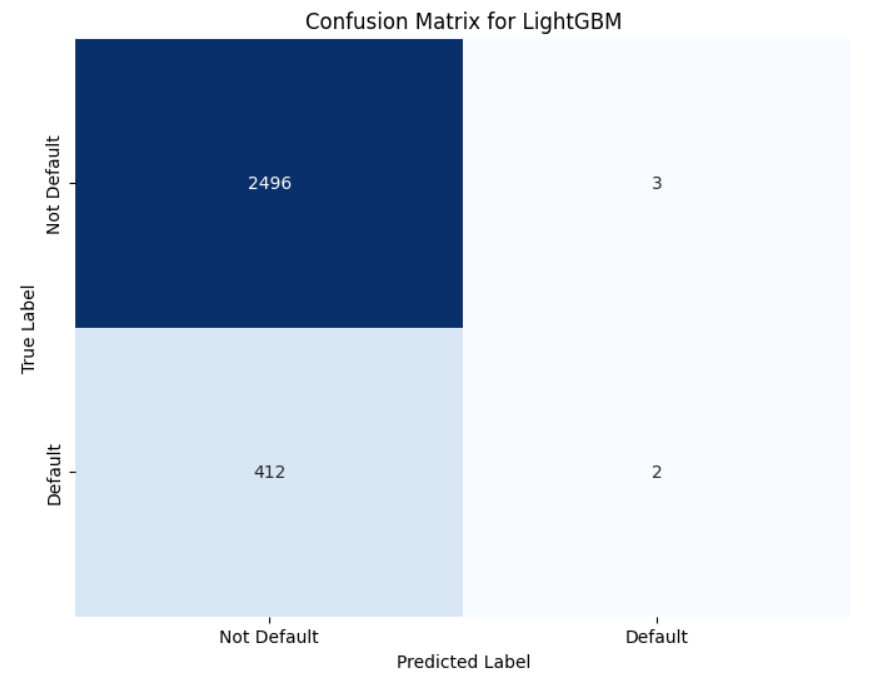
**CHAPTER – 5**

**EXPERIMENTATION AND RESULT ANALYSIS**

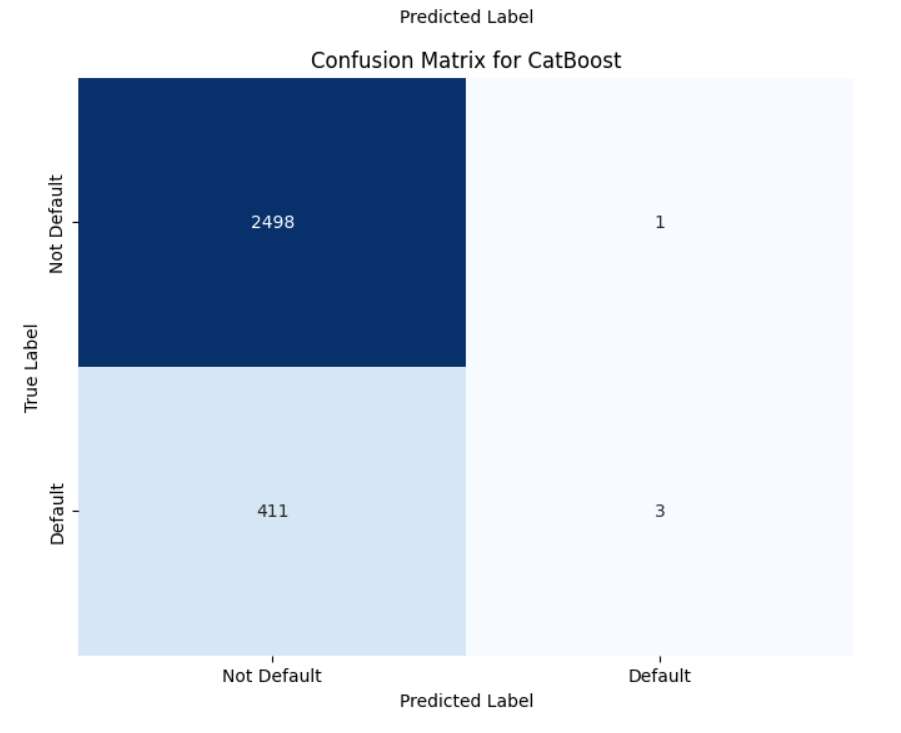
Our experimentation on different machine learning models was tested for both training and testing accuracy. XGBoost has attained a high train accuracy rate of 93.95% and a corresponding test accuracy of 85.58%, indicating very sound capacity for learning and generalizing unseen data. LightGBM appears very close to this figure with a 89.17% train accuracy and an 85.75% test accuracy, indicating the efficiency of these model performances as well. CatBoost has established a training accuracy of 88.70% and its corresponding test accuracy of 85.86%, whereas AdaBoost showed its training accuracy of 87.24% with a corresponding test accuracy of 85.62%. Through overall inspection, XGBoost worked better on both train and test accuracy and, by this, it becomes the most suitable for this dataset, and the other models also performed well in this dataset in comparison to each other, and hence these models are applicable to similar predictive tasks.

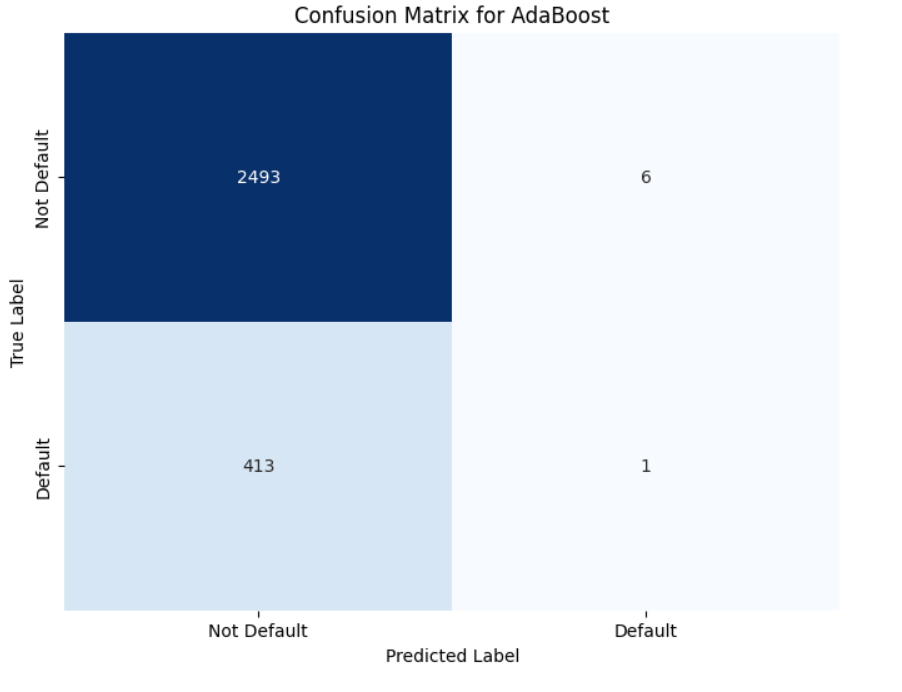


**FIG 2 :Confusion matrix for XGBoost**

****

**FIG 3 :Confusion matrix for LightGBM**

**FIG 5 :Confusion matrix for CarBoost**

****

**FIG 4 :Confusion matrix for AdaBoost**

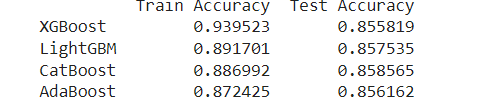
****

Fig 6: Accuracy for Boosting models (Bar graph)

**CHAPTER-6**

**CONCLUSION**

In conclusion, the project well demonstrated the applications of various machine learning algorithms toward determining the eligibility and predicting the default of credit cards. Through comprehensive experimentation with XGBoost, LightGBM, CatBoost, and AdaBoost, we were able to assess all the strengths and weaknesses of each model in terms of their training and testing accuracies. Its best-performing model was XGBoost that performed brilliant and thus has the highest training accuracy with strong generalization on the test set. LightGBM and CatBoost have also good results, so those will also prove to be efficient for similar tasks. AdaBoost showed a somewhat less powerful performance but still revealed important insights in model behaviour. The results deduced from this study indicate the importance of appropriate model selection and evaluation during the development of predictive analytics solutions. This brings forth a path for further development and fine-tuning. Overall, this project not only adds value to learning how machine learning applications find application in financial aspects but also serves as a basis for further exploration and analysis of more sophisticated techniques and the role of feature engineering in establishing greater accuracy and reliability within the model.

**CHAPTER -7**

**REFERENCES**

**REFERENCES:**

[1] Dhanalakshmi, B., & Merriliance, K. (2022). Credit Card Applicant Worthiness Prediction Using Machine Learning. International Journal of Creative Research Thoughts (IJCRT), Volume 10, Issue 12, December 2022.

[2] Devi, A. S., Bhargavi, C., Aishwarya, S. M. L., Phanindra, K. S., Vamsi, T., & Sai, R. N. M. K. (2023). Credit Card Approval Prediction Using Machine Learning. li-manager's Journal on Information Technology, 12(3), 39-45.

[3] Xgboost compared with Decision Tree Algorithm, " 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Chennai, India, 2022, pp. 674-679, doi: 10.1109/ICAC3N5670.2022.10074250

[4] Wahab, F., Khan, I., & Sabada, S. (2024). Credit card default prediction using ML and DL techniques. Internet of Things and Cyber-Physical Systems, 4, 293–306.

[5] Bhandary, R., & Ghosh, B. (2024). Credit Card Default Prediction: An Empirical Analysis on Prediction Performance Between Statistical and Machine Learning Methods. Preprints.org.

[6] Peiris, M. P. C. Credit Card Approval Prediction by Using Machine Learning Techniques. University of Colombo School of Computing. 2019.

[7] Babu, K., Prabhakaran, S., Marikkannu, P., Roobini, M. S., Rai, P., & Singh, A. P. (2024). Smart Credit Card Approval Prediction System using Machine Learning. E3S Web of Conferences, ICPES 2023.

[8] Omer Yavuz Can, "Credit Risk Analysis Using Machine Learning Techniques," International Journal of Engineering Research and Applications (IJERA), vol. 10, no. 01 (Series I), January 2020, pp. 56-59. DOI: 10.9790/9622-1001015659​

[9] Caggiano, P., & Giardini, D. Machine Learning Analysis of Credit Card Approval. Retrieved from Kaggle datasets, n.d.

[10] Teng, H.-W., & Lee, M. Estimation Procedures of Using Five Alternative Machine Learning Methods for Predicting Credit Card Default. Review of Pacific Basin Financial Markets and Policies, 22(3), 1950021.

[11] Nihatkar, S., Shahari, A., Patil, T., Kulkarni, A., & Potdukhe, S. (2023). Credit Card Approval Using Machine Learning Models. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 3(16), 229-243. DOI: 10.48175/IJARSCT-10986

[12] Omer Yavuz Can "Credit Risk Analysis Using Machine Learning Techniques." International Journal of Engineering Research and Applications (IJERA), vol. 10, no. 01, January 2020, pp. 56-59. DOI: 10.9790/9622-1001015659.

[13] Koh, H. C., & Chan, K. L. G. (2002). Data mining and customer relationship marketing in the banking industry. Singapore Management Review, 24(2), 1–27.

[14] S. Khatri, A. Arora and A. P. Agrawal, "Supervised Machine Learning Algorithms for Credit Card Fraud Detection: A Comparison," in 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 680-683, doi: 10.1109/Confluence47617.2020.9057851.

[15] Chen, Huan; Wu, Jiarui; Lin, Ying, "Credit Card Approval Prediction Using Deep Learning Models." Expert Systems with Applications, Vol. 138, 2019.

[16] M. P. C. Peiris, "Credit Card Approval Prediction by Using Machine Learning Techniques," University of Colombo, 2022. Accessed: 07 Mar. 2023.