Credit Card Users Churn Prediction Using Ensemble Techniques

Aishani Singh Rashmi Vashisth , Nidhi Sindhwani
Amity Institute of Information Technology
Amity University
Noida, India
aishani.singh@s.amity.edu

***Abstract*— Credit card companies are constantly striving to retain their customers due to intense competition in the financial industry. Churning refers to a customer who leaves one company to go to another company. Customer Churn is one of the most important and challenging problems for businesses such as Credit Card companies, cable service providers, SASS and telecommunication companies worldwide. Customer churn introduces not only some loss in income but also other negative effects on the operation of companies. While a certain level of churn is unavoidable, it is important to keep it under control, as high churn rate can potentially kill the business. Because of the significant importance of customer churn within a business, stakeholders are investing more time and effort in finding out the reasoning within their organizations, how they can accurately predict the type of existing customers that can stop doing business with them and what they can do to minimize the customer churn.**

In this paper, we propose an ensemble method for credit card user churn prediction, leveraging the strengths of multiple algorithms to enhance the predictive performance. We conduct experiments on a real-world credit card user dataset, and the experimental results demonstrate that our proposed ensemble method outperforms individual algorithms in terms of predictive accuracy, precision and recall. Our approach also provides insights into the key factors that contribute to credit card user churn, which can help credit card companies in formulating effective customer retention strategies. The findings of this study can be valuable for credit card companies to reduce customer churn and enhance customer satisfaction, thereby improving their business performance.

Keywords—credit card churn, ensemble techniques

I. INTRODUCTION

Ensemble methods are widely used in machine learning for their ability to combine multiple models and improve predictive accuracy. When it comes to credit card user churn prediction, which involves identifying customers who are likely to stop using their credit cards, ensemble methods can be particularly effective. It combines multiple machine learning algorithms to improve predictive accuracy, and has shown promising results in various domains. In order to proactively manage customer churn, it is imperative to develop a more robust and precise customer churn prediction model. [1]

Currently, the market is characterized by dynamism and high competitiveness, largely driven by the abundance of service providers, particularly banks, on a global scale. One of the primary obstacles faced by this sector pertains to the shifting customer behavior. Length, recency, and monetary variables were found to significantly impact churn, whereas frequency only emerged as a top predictor when the variability of the first three variables was constrained. [2] Customers serve as the nucleus of all industries, with customer-centric organizations, such as banks, playing a crucial role in accepting deposits, facilitating investments, and granting loans. The retention of long-term customers is intrinsically linked to the generation of profits. Therefore, it is imperative for banks to proactively prevent customer attrition. [3]

Credit card user churn prediction is the process of using data and machine learning algorithms to predict which credit card users are likely to churn, or stop using their credit card, in the near future. Churn prediction is an important task for credit card companies as it helps them identify at-risk customers who may be considering closing their credit card accounts, and allows them to take proactive measures to retain these customers and mitigate churn. According to research findings, it has been demonstrated that a bank has the potential to amplify its profits by a substantial 85% with a mere 5% escalation in its retention rate.[4]

This paper presents an overview of credit card users' churn prediction using ensemble techniques. We will discuss the importance of churn prediction in the credit card industry, the challenges associated with it, and how ensemble techniques can address these challenges. We will also highlight some commonly used ensemble techniques for credit card churn prediction, including bagging, and boosting, and discuss their advantages and limitations.

We will also discuss the considerations for selecting base models, ensemble methods, and hyperparameter tuning. Finally, we will provide an overview of evaluation metrics used in credit card churn prediction, such as accuracy, precision, recall, and highlight the importance of evaluating model performance using relevant business metrics. [6]

In conclusion, this paper aims to provide insights into credit card users' churn prediction using ensemble techniques, highlighting their advantages and limitations, and providing practical guidelines for building effective ensemble models for credit card churn prediction. By leveraging the power of ensemble techniques, credit card issuers and financial institutions can improve their churn prediction accuracy and take proactive measures to retain valuable customers, thus enhancing customer satisfaction and loyalty while mitigating revenue loss.

II. PROBLEM FORMULATION

Churn prediction is a common problem in the banking and financial industry, where identifying customers who are likely to churn or stop using a credit card is crucial for retaining and managing customer relationships. Churn prediction is a data analysis practice that involves scrutinizing historical data to forecast whether customers are likely to discontinue their business relationship in advance. [5] The prevention of customer churn is a pivotal factor in optimizing the revenue generation of an organization. Commonly known as customer attrition, it transpires when customers discontinue their usage of a company's products or services. [12] The goal is to predict which credit card users are most likely to churn so that proactive measures can be taken to retain them and minimize business losses.

Ensemble techniques, which combine multiple machine learning models to make predictions, are often used in churn prediction due to their ability to improve prediction accuracy and robustness. [7] Here's a problem formulation for credit card user churn prediction using ensemble techniques:

- Problem Statement: The objective is to predict whether a credit card user will churn or not.

- Dataset: The dataset used for the prediction consists of historical data of credit card users, including various features such as customer demographics, transaction history, credit limit, payment history, etc. The dataset is divided into two parts: a training dataset used for training the ensemble models and a test dataset used for evaluating their performance.

- Target Variable: The target variable is a binary variable that indicates whether a credit card user has churned (1) or not churned (0).

- Feature Engineering: The dataset may require feature engineering, which involves selecting relevant features, handling missing values, encoding variables, and normalizing/standardizing numerical variables to prepare the data for model training.

- Ensemble Techniques: Various techniques can be employed, such as:

  Bagging: Using techniques such as Random Forest or Bootstrap Aggregating, where multiple base models are trained on different subsets of the training data to reduce overfitting and improve model performance.

  Boosting: Using techniques such as Gradient Boosting, AdaBoost, or XGBoost, where base models are sequentially trained to correct the errors made by previous models and improve prediction accuracy.

- Model Evaluation: The performance of the ensemble models is evaluated using appropriate evaluation metrics such as accuracy, precision, recall. Cross-validation techniques, such as k-fold cross-validation, can also be used to obtain a more reliable estimate of model performance.

- Hyperparameter Tuning: Hyperparameters of the ensemble models, such as the number of base models, learning rate, maximum depth, etc., may need to be tuned to optimize model performance. Grid search or randomized search can be used for hyperparameter tuning.

- Model Interpretability: Interpretability of the ensemble models may be challenging due to their complex nature. Techniques such as feature importance analysis, partial dependence plots, or model-specific interpretability methods can be used to gain insights into model predictions and explain them to stakeholders.

- Deployment: Once the best-performing ensemble model is selected, it can be deployed in a production environment for real-time prediction of credit card user churn. Regular model monitoring and maintenance should be conducted to ensure model performance remains optimal over time.

- Business Impact: The results of the credit card user churn prediction can be used to proactively identify at-risk customers and take appropriate measures, such as targeted marketing campaigns, personalized offers, or customer retention programs, to retain customers and reduce churn. This can ultimately lead to increased customer satisfaction, improved business revenue, and enhanced customer relationship management.

III. PROBLEM SOLUTION

A. *Exploring the dataset and Extracting insights using Explanatory Data Analysis*

This paper depends on the dataset of credit card customer churn for banks; the dataset is taken from https://www.kaggle.com/code/kaushikmajumder/credit-card-customer-churn-prediction. First, we explore the dataset and extract meaningful insights using Exploratory Data Analysis. To achieve our objective we have used a dataset with 10127 entries. The dataset originally had 23 columns but after removing the unwanted attributes there were 20 columns. Out of all the 20 attributes the dependent attribute was ‘Attrition_Flag’ and all the rest were independent variables. The unique elements stored in the Attrition_Flag were ‘Existing Customer’ and ‘Attrited Customer’. These data elements were later changed to 1(Existing Customer) and 0 (Attrited Customer) respectively.

B. Checking for missing values

Then, we create a correlation matrix. It helps in identifying multicollinearity in a dataset, which occurs when two or more variables are highly correlated with each other. Multicollinearity can be problematic in certain statistical analyses, as it can inflate the standard errors of the regression coefficients and make it difficult to interpret the results accurately. In such cases, it may be necessary to drop one of the highly correlated variables from the analysis to reduce multicollinearity. [8] This decision should be based on domain knowledge and the specific goals of the analysis. To drop features based on a correlation matrix, one common approach is to identify pairs of variables with high absolute correlation coefficients, and then choose one variable from each pair to be dropped from the analysis. This can be done manually or using automated techniques such as stepwise regression, which iteratively adds and removes variables from a regression model based on their significance and impact on the model's performance. Therefore, we find drop all features with correlation greater than 0.95. Now, dataset is divided in the ratio of 70:30 for training and testing respectively.

C. Detecting and Removing Outliers

The next step is to detect outliers using the box-plot and remove them using the Z-score method. A box plot is a graphical representation of a dataset that provides a summary of its distribution, including the presence of outliers. Outliers are data points that deviate significantly from the majority of the data points and can have a significant impact on statistical analysis. [9]

Box plots are particularly useful for detecting outliers because they display the data's quartiles and the interquartile range (IQR), which is the range between the 25th percentile (Q1) and the 75th percentile (Q3). Outliers are typically identified as data points that fall outside the "whiskers" of the box plot, which represent the range within Q1 - 1.5 * IQR to Q3 + 1.5 * IQR. [10]

The Z-score method is a commonly used statistical technique for detecting outliers in a dataset. It involves calculating the z-score for each data point, which measures the number of standard deviations that a data point is away from the mean. The formula for calculating the z-score of a data point x is:

$$z = (x - \mu) / \sigma$$

where:
x is the data point
μ is the mean of the dataset
σ is the standard deviation of the dataset

To use the z-score method for detecting outliers, you can follow these steps:

1. Calculate the mean and standard deviation of the dataset.
2. Calculate the z-score for each data point using the formula above.
3. Define a threshold z-score value beyond which data points are considered outliers. We used a threshold value of 3, meaning that any data point with a z-score greater than 3 or less than -3 is considered an outlier.
4. Identify the outliers in the dataset based on the threshold z-score value. [11]

There were a total 10127 entries in the dataset however after removing the outliers present in our dataset the number of entries were reduced to 8842. The outliers were detected using the box-plot and removed using the Z-score method.

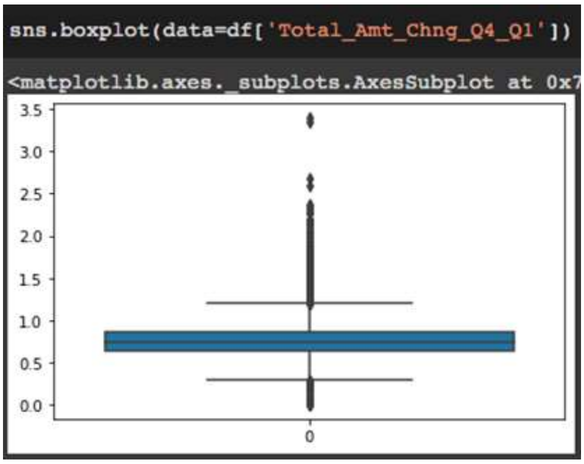

Fig 1- The above figure of the box plots shows the outliers present in the respective attribute.

D. Machine Learning Algorithms

We have developed 7 different classification algorithms for Credit Card Users Churn prediction to compare all the algorithms to find the classification model which will give optimum results. The 7 machine learning algorithms are :

1.Logistic Regression
2.Random Forest Classification
3.Decision Tree
4.Bagging
5.Gradient Boosting Classification
6.AdaBoost Classification
7.XGBoost Classification

1) Logistic Regression

Logistic regression is a statistical method used in machine learning for binary classification, which involves predicting the probability of an input data point belonging to one of two classes (e.g., yes/no, spam/ham, fraud/non-fraud). It is a simple but effective algorithm that is widely used for classification tasks in various fields such as finance, healthcare, marketing, and more. [12]

2) Random Forest Classification

Random Forest Classification is a popular supervised machine learning technique used for solving classification problems. It is an ensemble learning method that combines multiple decision tree models to make predictions. [13]

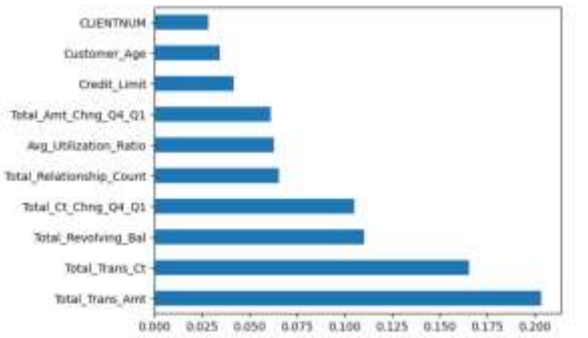

Fig-2 is showing the importance of every feature in Random Forest algorithm

3) *Decision Tree*

A decision tree is a popular supervised machine learning algorithm used for both classification and regression tasks. It is a tree-like structure where an internal node represents a decision based on a particular feature, and the branches represent the possible outcomes or decisions based on the values of that feature. The leaves of the tree represent the final decision or predicted output. [14]

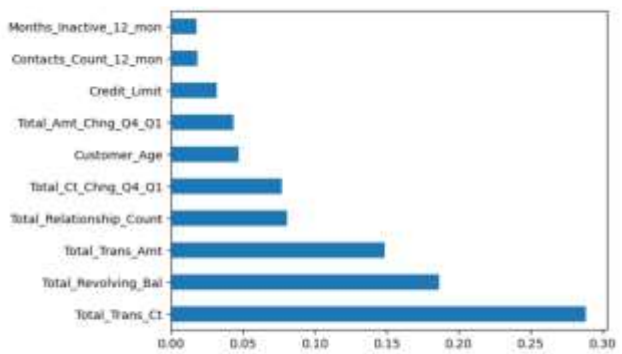

Fig-3 is showing the importance of every feature in Decision Tree algorithm

4) *Bagging*

Bagging, short for Bootstrap Aggregating, is a machine learning technique used for improving the accuracy and robustness of predictive models. It involves training multiple instances of the same model on different subsets of the training data and then combining their predictions to obtain a more accurate and stable prediction. [15]

5) *Gradient Boosting Classification*

Gradient Boosting Classification is a popular machine learning technique used for solving binary or multiclass classification problems. It belongs to the family of ensemble methods, which combine the predictions of multiple base models to produce a more accurate and robust model. [16]

6) *AdaBoost Classification*

AdaBoost, short for Adaptive Boosting, is a popular ensemble learning algorithm used for classification in machine learning. Ensemble learning refers to combining the predictions of multiple base models to create a more accurate and robust model. AdaBoost is one such technique that combines weak learners (base models that perform slightly better than random chance) to create a strong learner. [17]

7) *XGBoost Classification*

XGBoost (Extreme Gradient Boosting) is a popular machine learning algorithm used for classification tasks. It is an advanced version of the gradient boosting algorithm that is optimized for both speed and performance. XGBoost is particularly well-suited for handling large datasets and achieving high accuracy in various classification tasks, such as spam detection, fraud detection, image classification, and medical diagnosis. [18]

TABLE I. COMPARISON OF ALGORITHMS

Parameters	Logistic Regression	Random Forest	Decision Tree
Algorithm Type	Linear Regression	Ensemble Method	Ensemble Method
Model Interpretability	Interpretable	Less interpretable as they combine multiple trees to make predictions.	Interpretable
Handling of categorical variables	Can handle both categorical and continuous variables, but requires encoding categorical variables into numerical values	Can handle categorical variables directly	Require encoding categorical variables into numerical values.
Handling of missing values	Can handle missing values in the data	Can handle missing values in the data	Can handle missing values in the data
Model performance	May be simpler and computationally efficient options, but may not perform as well in certain scenarios.	Tend to perform well in terms of accuracy, especially when dealing with complex datasets with non-linear relationships	May be simpler and computationally efficient options, but may not perform as well in certain scenarios.
Model training time	Typically faster to train compared to ensemble methods	Takes more time, as it require sbuilding multiple trees	Typically faster to train compared to other ensemble methods
Model robustness	More sensitive to noisy or imbalanced datasets, and may require additional data preprocessing or tuning	Are known for their ability to handle noisy or imbalanced datasets, as they can reduce overfitting by averaging or combining multiple trees	More sensitive to noisy or imbalanced datasets, and may require additional data preprocessing or tuning
Hyperparameter tuning	Has relatively fewer hyperparameters to tune compared to ensemble methods	Have multiple hyperparameters that need to be tuned for optimal performance	Have multiple hyperparameters that need to be tuned for optimal performance

TABLE II. COMPARISON OF ALGORITHMS

Parameters	Bagging	Gradient Boosting
Algorithm Type	Ensemble Method	Ensemble Method
Model Interpretability	Less interpretable as they combine multiple trees to make predictions.	Less interpretable as they combine multiple trees to make predictions.
Handling of categorical variables	Can handle categorical variables directly	Usually require encoding categorical variables into numerical values.
Handling of missing values	Can handle missing values in the data	Generally require imputation of missing values before training the model.
Model performance	Tend to perform well in terms of accuracy, especially when dealing with complex datasets with non-linear relationships	Tend to perform well in terms of accuracy, especially when dealing with complex datasets with non-linear relationships
Model training time	Among the ensemble methods, Bagging is usually faster	Slower than bagging
Model robustness	Are known for their ability to handle noisy or imbalanced datasets, as they can reduce overfitting by averaging or combining multiple trees	Are known for their ability to handle noisy or imbalanced datasets, as they can reduce overfitting by averaging or combining multiple trees
Hyperparameter tuning	Have multiple hyperparameters that need to be tuned for optimal performance	Have multiple hyperparameters that need to be tuned

TABLE III. COMPARISON OF ALGORITHMS

Parameters	AdaBoost	XgBoost
Algorithm Type	Ensemble Method	Ensemble Method
Model Interpretability	Less interpretable as they combine multiple trees to make predictions.	Less interpretable as they combine multiple trees to make predictions.
Handling of categorical variables	Usually require encoding categorical variables into numerical values.	Usually require encoding categorical variables into numerical values.
Handling of missing values	Generally require imputation of missing values before training the model.	Generally require imputation of missing values before training the model.
Model performance	Tend to perform well in terms of accuracy, especially when dealing with complex datasets with non-linear relationships	Tend to perform well in terms of accuracy, especially when dealing with complex datasets with non-linear relationships
Model training time	Slower than bagging	Known for its fast training speed
Model robustness	More sensitive to noisy or imbalanced datasets, and may require additional data preprocessing or tuning	More sensitive to noisy or imbalanced datasets, and may require additional data preprocessing or tuning
Hyperparameter tuning	Have multiple hyperparameters that need to be tuned	Have multiple hyperparameters that need to be tuned

In summary, Logistic Regression is a simple linear model, while Decision Tree, Bagging, Gradient Boosting, AdaBoost, and XGBoost are ensemble methods that combine multiple models for improved accuracy. Random Forest is a specific ensemble method that uses decision trees. Bagging and AdaBoost adjust the weights of samples during training, while Gradient Boosting and XGBoost use sequential training with error corrections. Each algorithm has its strengths and weaknesses, and the choice depends on the specific requirements of the task at hand, the size and complexity of the dataset, and the need for interpretability, accuracy, and computational efficiency.

E. Applying Upsampling and Downsampling

Upsampling and **Downsampling** are techniques used in machine learning, including logistic regression, to address class imbalance in datasets. Class imbalance occurs when the distribution of samples among different classes is uneven, and it can lead to biased model performance, where the minority class is often poorly predicted.

- Upsampling involves randomly duplicating instances from the minority class to increase their representation in the dataset. This can be done by randomly selecting instances from the minority class and adding them to the original dataset, effectively increasing the number of samples in the minority class. This technique helps the model to learn from more instances of the minority class, which can improve its ability to predict the minority class accurately.

- Downsampling involves randomly removing instances from the majority class to reduce its representation in the dataset. This can be done by randomly selecting instances from the majority class and removing them from the original dataset, effectively reducing the number of samples in the majority class. This technique helps to balance the class distribution in the dataset, which can prevent the model from being biased towards the majority class and give equal importance to the minority class during model training. [20]

F. Hyperparameter Tuning

All the models were also tuned using **Grid Search** and **Randomized Search**. Tuning hyperparameters is an essential step in machine learning model development to optimize their performance. Grid search and randomized search are two popular techniques for hyperparameter tuning. [21]

- Grid search involves specifying a fixed set of hyperparameter values and exhaustively searching all possible combinations of those values. It creates a grid of all possible hyperparameter values and evaluates the model's performance for each combination. Grid search is straightforward and guarantees that all possible combinations will be evaluated.

- Randomized search, on the other hand, randomly selects a fixed number of combinations from the hyperparameter space and evaluates the model's performance for those combinations. Randomized search is more efficient in terms of computational resources compared to grid search as it explores only a subset of hyperparameter combinations.

Hyperparameter Tuning was performed using Pipelines. After applying Grid Search and Randomised Search the best model was predicted along with the best estimator in both the cases respectively.

G. Confusion Matrix

To assess performance and errors, we used **Confusion Matrix**. A confusion matrix is a performance evaluation tool used in machine learning to assess the accuracy of a classification model. It is a square matrix that displays the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by the model on a set of data. [19]

Where:

- True Positive (TP): The number of samples that are correctly predicted as positive by the model.
- True Negative (TN): The number of samples that are correctly predicted as negative by the model.
- False Positive (FP): The number of samples that are incorrectly predicted as positive by the model.
- False Negative (FN): The number of samples that are incorrectly predicted as negative by the model.

The confusion matrix provides a visual representation of the model's performance, allowing for an assessment of how well the model is correctly predicting positive and negative samples. It is a table used to evaluate the performance of a classification model, "recall," "F1 score," and "precision" are performance metrics that provide insights into different aspects of the model's performance:

- Recall (also known as true positive rate or sensitivity): It measures the proportion of actual positive cases that were correctly predicted by the model. It is calculated as the ratio of true positives (TP) to the sum of true positives and false negatives (FN) in the confusion matrix. Mathematically, recall is defined as:
  Recall = TP / (TP + FN)

- F1 score: It is the harmonic mean of precision and recall, and is a single score that balances both precision and recall. It is calculated as the harmonic mean of precision (P) and recall (R), and is often used when both precision and recall are important. Mathematically, F1 score is defined as:
  F1 score = 2 * (Precision * Recall) / (Precision + Recall)

- Precision (also known as positive predictive value): It measures the proportion of predicted positive cases that were correctly predicted by the model. It is calculated as the ratio of true positives (TP) to the sum of true positives and false positives (FP) in the confusion matrix. Mathematically, precision is defined as:
  Precision = TP / (TP + FP)

These metrics are commonly used in machine learning and classification tasks to evaluate the performance of a model and assess its accuracy, sensitivity, and specificity in predicting positive and negative cases. A higher value for recall, F1 score, and precision indicates better performance of the model, while a lower value indicates poorer performance. The appropriate metric to use depends on the specific requirements and goals of the classification task.

IV. RESULTS AND DISCUSSION

The following are the test accuracies of all the algorithms:

TABLE IV. TEST ACCURACIES

S.NO.	TEST ACCURACIES	
	Algorithms	***Accuracy***
1	Logistic Regression Test Accuracy	0.8964269561284487
2	Decision Tree Test Accuracy	0.9271822704658526
3	Random Forest Test Accuracy	0.95838986883763
4	Gradient Boosting Test Accuracy	**0.9651741293532339**
5	AdaBoost Test Accuracy	0.9543193125282677
6	XGBoost Test Accuracy	0.9642695612844867
7	Bagging Test Accuracy	0.9123527621249832

Fig-4 Test Accuracies Comparison Graph of Algorithms

Undersampling - Logistic Regression

Undersampling was applied to the train set.
The train set has 5924 entries: ‘0’ - 963 , ‘1’ - 4961.
After applying undersampling to the dataset the entries in the train set reduced as follows : ‘0’ - 963 , ‘1’ - 1203.
Accuracy on the test set : **0.7655604361347561**

Oversampling - Logistic Regression

Oversampling was applied to the train set.
The train set has 5924 entries: ‘0’ - 963 , ‘1’ - 4961.
After applying Oversampling to the dataset the entries in the train set increased as follows : ‘0’ - 3720 , ‘1’ - 4961.
Accuracy on the test set : **0.9356137226051304**

Grid-Search

After applying, the Grid-Search using the pipeline ‘Gradient Boosting Classifier’ was predicted to be the best model with an accuracy of 0.9701492537313433.
Computational time of grid search was **173.36 seconds**.

Randomized-Search

After applying the Grid-Search using the pipeline ‘Gradient Boosting Classifier’ was predicted to be the best model with an accuracy of 0.9651741293532339.

Computational time of randomized search was **16.09 seconds.**

Based on the predictive models we created and the results obtained it is observed that **Gradient Boosting Classifier** performs better than the other models.

Gradient Boosting Classifier gave the highest accuracy before any hypertuning parameters were applied to any of the models. After applying grid search to the models the best model was predicted to be the gradient boosting classifier and after randomized search also the best model was predicted to be the gradient boosting classifier. The computational time taken by grid search was way more than randomized search. Therefore, **Randomized search** is a better method for hyperparameter tuning than the Grid search.

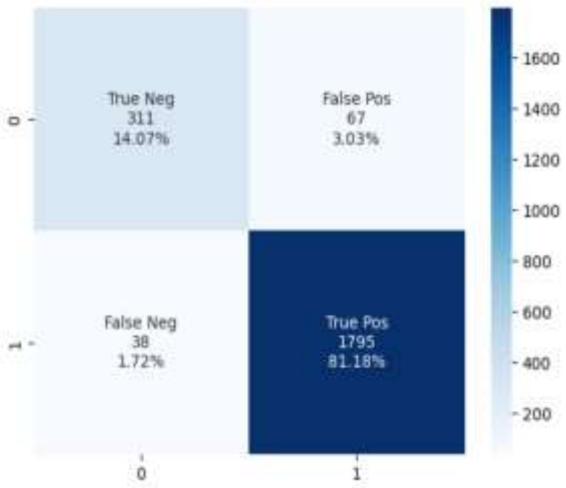

Fig-4 Confusion Matrix of the Best Table

TABLE V. CONFUSION MATRIX TABLE

	CONFUSION MATRIX TABLE						
ALGORITHM	TP	TN	FP	FN	RECALL	PRECISION	F1 SCORE
GRADIENT BOOSTING	2165	375	75	38	0.982	0.966	**0.973**
LOGISTIC REGRESSION	2203	0	450	0	1	0.830	0.907
DECISION TREE	2116	360	90	87	0.960	0.959	0.958
RANDOM FOREST	2170	365	85	31	0.985	0.962	0.972
BAGGING	2198	147	303	5	0.997	0.878	0.933
ADABOOST	2165	75	375	38	0.982	0.852	0.909
XGBOOST	2163	380	72	42	0.980	0.967	0.972

V. CONCLUSION

Ensemble methods are effective techniques for credit card user churn prediction. By combining multiple models, ensemble methods can overcome the limitations of individual models and provide more accurate and robust predictions. Some commonly used ensemble methods for credit card user churn prediction include bagging and boosting.

Bagging, which involves training multiple instances of the same model with different subsets of the training data and averaging their predictions, can reduce overfitting and improve model performance. Boosting, on the other hand, combines weak models into a strong model by sequentially giving more importance to misclassified instances, leading to improved accuracy.

Ensemble methods have several advantages in credit card user churn prediction, such as increased accuracy, robustness, and generalization ability. They can also handle imbalanced data, noisy data, and missing values effectively, making them suitable for real-world credit card churn prediction scenarios.

However, it's important to note that ensemble methods also have some limitations. They can be computationally expensive and require careful tuning of hyperparameters. Additionally, the interpretability of ensemble models may be lower compared to individual models, which can make it challenging to explain the reasoning behind the predictions to stakeholders.

In conclusion, ensemble techniques are powerful tools , and their effectiveness can be enhanced by carefully selecting appropriate base models, tuning hyperparameters, and conducting proper model evaluation. They can improve the accuracy and robustness of churn prediction models, helpring credit card companies identify potential churners and take proactive measures to retain customers and improve business performance.

VI. FUTURE SCOPE

Credit card user churn prediction is an important area of research and analysis for credit card companies to identify customers who are likely to close their credit card accounts and stop using their credit cards. By accurately predicting churn, credit card companies can take proactive measures to retain customers and improve customer satisfaction, which ultimately leads to increased revenue and profitability.

Here are some potential future scopes of credit card user churn prediction:

- Enhanced customer retention: By predicting customer churn, financial institutions can take proactive steps to retain their customers, such as offering targeted promotions, loyalty programs, and personalized recommendations. This could help enhance customer loyalty and reduce churn rates.

- Improved customer experience: Credit card user churn prediction can also help financial institutions identify the factors that contribute to customer dissatisfaction and improve their overall experience.

This could include improving customer service, offering better rewards programs, and simplifying the credit card application process.

- Fraud detection: Credit card user churn prediction could also be used to detect fraudulent activities. By analyzing customer behavior, financial institutions can identify unusual or suspicious transactions and take appropriate action to prevent fraud.

- Machine learning algorithms: With the advent of advanced machine learning algorithms, credit card user churn prediction models can be developed and trained more accurately and efficiently. This could lead to better prediction accuracy and enable financial institutions to identify customers who are likely to churn in real-time.

- Integration with other systems: Credit card user churn prediction models could be integrated with other systems such as customer relationship management (CRM) software, billing systems, and fraud detection systems. This could provide a more holistic view of the customer and help financial institutions to make better decisions.

Overall, credit card user churn prediction is an essential area of research and development for financial institutions, and it has significant future scope. With the increasing adoption of advanced machine learning algorithms, integration with other systems, and the focus on enhancing customer experience, credit card user churn prediction is poised to become a critical tool for financial institutions in retaining their customers.

REFERENCES

[1] Sharma A, Panigrahi PK (2011) A neural network based approach for predicting customer churn in cellular network services. Int J Comput Appl27(11):26–31

[2] Chen K, Hu Y-H, Hsieh Y-C (2014) Predicting customer churn from valuable B2B customers in the logistics industry: a case study. IseB 13:475–494.

[3] Machine Learning to Develop Credit Card Customer Churn Prediction Dana AL-Najjar 1 , Nadia Al-Rousan 2 and Hazem AL-Najjar 3,

[4] Customer Churn Prediction on Credit Card Services using Random Forest Method Xinyu Miao1, Haoran Wang2

[5] Churn Prediction Using Ensemble Learning, Authors -Xing Yang, Khang Nguyen. Binh P. Nguyen

[6] Machine Learning Case Study : Telco Customer Churn Prediction, By-Felix Frohbose

[7] Ensemble Methods in Customer Churn Prediction: A Comparative Analysis of the State-of-the-Art by Matthias Bogaert 1,2,*ORCID andLex Delaere 1

[8] Multicollinearity in Regression Analysis: Problems, Detection, and Solutions- By Jim Frost

[9] How To Find Outliers in Data Using Python (and How To Handle Them) Eric Kleppen, Contributor to the CareerFoundry blog -BY ERIC KLEPPEN, UPDATED ON APRIL 5, 2023

[10] A Complete Guide to Box Plots- Posted by Mike Yi

[11] 5 Ways to Find Outliers in Your Data- By Jim Frost

[12] Logistic Regression for Machine Learning by Jason Brownlee on April 1, 2016 in Machine Learning Algorithms

[13] Random Forests, January 2011Machine Learning 45(1):157-176, In book: Ensemble Machine Learning: Methods and Applications (pp.157-176)Authors: Adele Cutler, David Richard Cutler, John R Stevens

[14] Decision Tree in Machine Learning By- Prince Yadav ,Nov 14, 2018

[15] An Empirical Study of Ensemble Techniques (Bagging, Boosting and Stacking), March 2019, Authors: Rising Odegua, Nossa Data

[16] Gradient Boosting Machine(GBM), By- Shreyas Sukhadeve

[17] Master the AdaBoost Algorithm: Guide to Implementing & Understanding AdaBoost, Author- Anshul Saini — Published On September 15, 2021 and Last Modified On March 3rd, 2023

[18] Extreme Gradient Boosting (XGBoost) Ensemble in Python by Jason Brownlee on November 23, 2020 in Ensemble Learnin

[19] Understanding & Interpreting Confusion Matrices for Machine Learning (Updated 2023), Author- Aniruddha Bhandari — Published On April 17, 2020 and Last Modified On April 11th, 2023

[20] Random Oversampling and Undersampling for Imbalanced Classification- by Jason Brownlee on January 15, 2020 in Imbalanced Classification

[21] Random search for hyper-parameter optimization- Authors: James Bergstra,Yoshua Bengio

[22] R. Vashisth, S. Tripathi, H. Goel and P. Srivastava, "Visualization of Covid-19 Pandemic Data: An Analysis," 2022 3rd International Conference on Computation, Automation and Knowledge Management (ICCAKM), Dubai, United Arab Emirates, 2022, pp. 1-6, doi: 10.1109/ICCAKM54721.2022.9990151.

[23] M. Manhas, D. Sanduja, R. Vashisth and N. Aggarwal, "Automated Gate Control System using Face Mask Detection," 2022 International Mobile and Embedded Technology Conference (MECON), Noida, India, 2022, pp. 184-188, doi: 10.1109/MECON53876.2022.9752417.

[24] M. Manhas, D. Sanduja, N. Aggarwal and R. Vashisth, "Design and Implementation of Artificial Intelligence (AI) Based Home Automation," 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, India, 2021, pp. 122-126, doi: 10.1109/ISPCC53510.2021.9609482.

[25] Anushka, V. Mishra, S. Bhardwaj and R. Vashisth, "Design and Development of Smart Cart System using Artificial Intelligence," 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, India, 2021, pp. 127-130, doi: 10.1109/ISPCC53510.2021.9609507.