CUSTOMER CHURN PREDICTION USING

MACHINE LEARNING

### A PROJECT REPORT

#### Submitted by

**SAMIKSHA UPADHYAY [Reg No: RA1911003010391]**

#### Under the Guidance of’

#### Mrs M. RAJALAKSHMI

Assistant Professor, Department of Computing Technologies

#### In partial fulfilment of the requirements for the degree of

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**



**DEPARTMENT OF COMPUTING TECHONOLOGIES**

**COLLEGE OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR – 603 203**

## MAY 2023



**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR – 603 203**

**BONAFIDE CERTIFICATE**

Certified that **18CSP111L** B.Tech project report titled “**CUSTOMER CHURN PREDICTION USING MACHINE LEARNING**” is the bonafide work of **Ms. Samiksha Upadhyay [Reg.No.RA1911003010391**] who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

**MS. M. RAJALAKSHMI**

Assistant Professor

Department of Computing Technologies

**Dr. B. Baranidharan**

### PANEL HEAD

### Associate Professor

### Department of Computing

### Technologies

**DR. M. PUSHPALATHA**

**HEAD OF THE DEPARTMENT**

Department of Computing Technologies

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

Department of Computing Technologies

**SRM Institute of Science and Technology**

**Own Work Declaration Form**

**Degree/ Course :** B.Tech in Computer Science and Engineering

**Student Names :** Samiksha Upadhyay

**Registration Number:** RA1911003010391

**Title of Work :** CUSTOMER CHURN PREDICTION USING MACHINE LEARNING

I hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

I confirm that all the work contained in this assessment is our own except where indicated, and that we have met the following conditions:

* Clearly references / listed all sources as appropriate
* Referenced and put in inverted commas all quoted text (from books, web, etc.)
* Not made any use of the report(s) or essay(s) of any other student(s) either past or present
* Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
* Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

|  |
| --- |
| **DECLARATION:** |
| I am aware of and understand the University’s policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.  **Student 1 Signature:**  **Date:** |
| If you are working in a group, please write your registration numbers and sign with the date  for every student in your group. |

**ACKNOWLEDGEMENT**

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T. V. Gopal**, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr. M. Pushpalatha,** Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators, **Dr. M. Kanchana**, **Dr. G. Usha, Dr. R. Yamini and Dr. K. Geetha**, Panel Head, **Dr. B. Baranidharan**, Associate Professor and Panel Members, **Mrs. M Rajalakshmi** Assistant Professor**, and Dr. P Saranya** Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for their inputs during the project reviews and support**.**

We register our immeasurable thanks to our Faculty Advisor, **Dr. G. Usha**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Ms. M. Rajalakshmi**, Associate Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff and students of Computing Technologies Department, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement.

# SAMIKSHA UPADHYAY [Reg. No: RA1911003010391]

# ABSTRACT

##### One significant problem that businesses face is customer attrition. It has become crucial for corporate operations and growth to prevent customer churn and work to keep clients. It is challenging to effectively estimate customer turnover because the majority of the existing projections use a single prediction model. Concentrating on the results of predictions of the models of machine learning, this study proposes a combination of estimating model for customer turnover and performs practical research on the model's efficacy. The combined prediction model outperforms the single customer churn prediction model in terms of accuracy and predictive impact, according to the findings of the predictions. It can also more naturally express the fundamental traits of the churn consumers.

**TABLE OF CONTENTS**

[**ABSTRACT**](#_2u6wntf) **vi**

[**TABLE OF CONTENTS iv**](#_3rdcrjn)

[**LIST OF FIGURES**](#_lnxbz9) **ix**

[**LIST OF TABLES**](#_35nkun2) **x**

[**ABBREVIATIONS**](#_lnxbz9) **xi**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | [**1 INTRODUCT**](#_1ksv4uv)**ION ……………………………………1** | | | | | | **1** | |
| 1.1 Overview ...…………………………………………….1 | | | | | |  | |
| 1.2 Approach ………………………………………………2 | | | | | |  | |
|  | | | | | |  | |
| **2 LITERATURE SURVEY…………………………….6** | | | | | | **3** | |
|  | | | | | |  | |
|  | | | | | |  | |
| **3 SYSTEM ARCHITECTURE AND DESIGN.………10** | | | | | | **8** | |
| 3.1 System Architecture ……………………….……….… 12  3.2 Data Flow Diagram ……………………….………….. 13  3.3 UML Diagrams ………………………….…………….14 | | | | | |  | |
| 3.3.1 Class Diagram………………………………………….14 | | | | | |  | |
| 3.3.2 Use case Diagram……………………………………...15  3.3.3 Activity Diagram………………………………………16. | | | | | |  | |
|  | | | | | |  | |
|  | **4 METHODOLOGY……………………………………17** | | | | | | **15** | |
|  | 4.1 Existing System ………………………..……………….18  4.2 Proposed System………………………….…………….19  4.3 Data Retrieval Process ……………………….………...20 | | | | | |  | |
|  |  | | | | | |  | |
|  |  | | | | | |  | |
|  |  | | | | | |  | |
| **5** | | **CODING AND TESTING………………………………………….... 37** | | | | | |
| **6** | | **RESULTS AND DISCUSSIONS…………………………..................77** | | | | | |
| **7** | | **CONCLUSION AND FUTURE ENHANCEMENT………………..80**  **REFERENCES………………………………………………………...81** | | | |
|  |  | | | | | |  | |
|  | | | |  |
|  | | | |  |
| **APPENDIX** | | |  | | | |  | |
| **A** | | | **CONFERENCE PUBLICATION…………………………..83** | | | | **50** | |
| **B** | | | **RESEARCH PAPER……………………………………..…84** | | | | **51** | |
| **C** | | | **PLAGIARISM REPORT………………………………..….82** | | | | **52** | |

**LIST OF FIGURES**

###### 3.1 System Architecture………………………………………...14

###### 3.2 DataFlow Diagram…….………………………………....…16

###### 3.3 UML Diagram…………………………………………...….17

**3.4 Class Diagram………………………………………………**18

###### 3.5 Use case diagram……………………………………………19

###### 3.6 Acvity diagram…………………………………………...…20

**4.1 Histogram Of Numerical Data………………………….....**22

**4.2 Distribution Of Label Encoded Tables………………...….**23

###### 6.1 Accuracy graph for different models………………………70

**LIST OF TABLES**

###### 6.1 Accuracy graph for different models………………………69

###### 

**LIST OF SYMBOLS AND ABBREVIATIONS**

**ANN** Artificial Neural Network

**CNN** Convolutional neural network

**MLP** Multilayer perception

**MFCC** Mel-Frequency Cepstral Coefficients

**KNN** K-nearest neighbor

**SVM** Support Vector Machine

## CHAPTER 1

**INTRODUCTION**

* 1. **Overview**

The percentage of customers who discontinue doing business with a company over time is referred to as customer churn. This may be the result of a number of factors, including unhappiness with the good or service, a better deal from a rival, or a change in the client's requirements.

Businesses should monitor and lower customer churn because it can significantly affect their revenue and profitability. A high rate of client turnover may be a sign that a business is underperforming and losing ground to rivals.

Companies can take a variety of actions to lower customer turnover, including enhancing the quality of their goods and services, offering better customer support, establishing loyalty reward schemes, and assessing customer input to pinpoint improvement areas. Finding customer turnover is crucial because it significantly affects a company's revenue and profitability. When customers leave doing business with a firm, it affects both the revenue of the business and the expense of attracting new clients to take their place.

Also, figuring out why customers leave a company might help that company develop better goods, services, or customer experiences. Companies may take corrective action to fix those issues, boost customer happiness, and lessen the possibility of future churn by identifying why customers are leaving[6]. Reducing customer turnover can also result in more loyal customers, which has a number of advantages like a higher lifetime value for customers, positive word-of-mouth marketing, and a market edge. Customer turnover may be predicted and reduced with the help of machine learning, which is a potent instrument. Machine learning models can help organizations identify customers who are at danger of leaving and assist them in taking preventative actions to keep them. Many machine learning algorithms, such as logistic regression, decision trees, random forests, and neural networks, can be used to forecast customer turnover[10]. These algorithms can be taught using past data to spot trends and indicate which clients are most likely to leave. A company can take a number of actions to keep its customers after determining which ones are likely to leave, like providing them with special offers or discounts. The elements that are causing client churn, such as quality of product, customer service, or pricing, can also be found using machine learning[4]. Businesses can learn important insights into the reasons why customers are departing and take corrective action to address those issues by evaluating customer feedback and behavior data. Overall, machine learning can be an effective tool for firms to boost client retention and decrease churn, which will increase sales and profitability. Figure 1 depicts the architecture model of the system described in this paper.

**1.2 Approach**

Customer churn, also known as customer attrition or customer turnover, is the rate at which customers stop doing business with a company over a certain period of time. It is a critical metric for businesses as it directly affects their revenue and profitability. Churn can occur for various reasons, including poor customer service, high prices, inferior product quality, or a lack of engagement with customers. It can also be due to factors outside the company's control, such as changes in the market, new competitors entering the market, or economic downturns.

To minimize churn, businesses must focus on improving customer experience, providing excellent customer service, offering competitive pricing, and regularly engaging with customers. Analyzing churn patterns can also help businesses identify areas of weakness and make strategic improvements to retain customers.

There are different types of customer churn that businesses may experience, including:

* Voluntary churn: This occurs when customers decide to end their relationship with a business. It may be due to factors such as dissatisfaction with the product or service, high prices, or a better offer from a competitor.
* Involuntary churn: This occurs when customers are lost due to reasons beyond their control, such as relocation, death, or change in personal circumstances.
* Deliberate churn: This occurs when a customer intentionally terminates their relationship with a business, often to switch to a competitor or to a different product or service.
* Indirect churn: This occurs when customers are lost as a result of actions taken by third parties, such as resellers, distributors, or affiliates.
* Passive churn: This occurs when customers stop using a product or service but do not explicitly cancel their subscription or end their relationship with a business. It may be due to reasons such as forgetfulness, lack of interest, or being too busy.

Understanding the different types of churn can help businesses develop strategies to reduce customer churn and retain customers for the long term.

* 1. **General Steps Involved in Churn Prediction**

1. **Data collection**: Gather customer data such as demographics, transaction history, customer interactions, and other relevant data points. Here, we have used a dataset of a telecom service provider from Kaggle. Data collection is the process of gathering and measuring information on targeted variables or observations in a systematic manner. This process involves identifying a specific research question or hypothesis, selecting a relevant sample population, and determining the appropriate data collection methods and tools to use.
2. **Data preprocessing**: Clean the data by handling missing values, outliers, and inconsistencies. Then, transform and normalize the data to make it suitable for machine learning algorithms. Data preprocessing is the process of preparing and cleaning raw data before it is used in data analysis or machine learning models. It involves a series of steps that transform the data into a format that is more suitable for analysis. Some common techniques used in data preprocessing include:

* Data cleaning entails eliminating or rectifying any incomplete, inaccurate, or superfluous data.
* Data transformation: This entails changing the format of data, such as transforming categorical variables to numerical values or scaling numerical values to a particular range.
* Data reduction: This entails lowering the quantity of the data by sampling a subset of the data or choosing just pertinent attributes.
* Data integration: This entails fusing information from various datasets into one.
* Data normalization: This process entails scaling the data to provide a stable range of values.
* Data discretization, which entails dividing continuous data into discrete groups.

1. **Feature engineering**: Extract relevant features from the data that are likely to influence customer churn. This may include variables such as customer tenure, purchase frequency, complaints, and customer satisfaction ratings. Feature engineering is the process of transforming raw data into features that can be used in machine learning models. It involves selecting and creating features that are relevant to the problem being solved and that will help the machine learning model make accurate predictions. Some common techniques used in feature engineering include:

* Feature selection: This entails choosing the features that are most pertinent from a wider group of features. This can be accomplished by performing statistical analyses or by determining the significance of each attribute.
* Feature extraction: This technique entails generating fresh features from older ones. Principal component analysis techniques or the development of new features based on domain expertise can be used to accomplish this.
* Feature scaling: This entails adjusting feature values to give them a comparable range. This is significant because the scale of features can affect some machine learning models.
* Feature normalization: This process involves giving features' values a Gaussian distribution. Techniques like the Box-Cox transformation can be used for this.

1. **Model selection**: Choose a machine learning model that is appropriate for the problem. Common models used for churn prediction include logistic regression, decision trees, random forests, and support vector machines. Model selection is a crucial step in building predictive models, as it involves choosing the best model among a set of candidate models that have been trained on the same dataset. The goal of model selection is to find the model that can generalize well on unseen data and provide accurate predictions. The process involves evaluating the performance of each candidate model using appropriate evaluation metrics, such as accuracy, precision, recall, or F1 score, and selecting the one that performs the best. Model selection techniques can vary depending on the dataset and the type of model being used, but commonly used methods include cross-validation, holdout validation, and bootstrapping. Choosing the right model can have a significant impact on the performance of the predictive model, and thus, it is important to take the time to carefully select the best model for a given task.
2. **Model training**: Train the chosen model on the preprocessed data. Model training is a critical aspect of machine learning, where the objective is to teach a model how to make accurate predictions. In this process, a machine learning algorithm is used to iteratively adjust the model's parameters so that it can learn to recognize patterns in the input data and produce the desired output. The process involves feeding the algorithm with a training dataset consisting of input features and corresponding target labels. The algorithm uses this data to optimize the model's parameters so that it can make accurate predictions on unseen data. The success of model training is dependent on the quality of the training dataset, the choice of algorithm and optimization technique, and the hyperparameters of the model. The performance of the model is usually evaluated using validation techniques, such as cross-validation, to ensure that it can generalize well to new data. Model training can be a computationally intensive process, particularly for large datasets and complex models, and may require specialized hardware or cloud resources to achieve optimal results. Proper training of a model is crucial for achieving accurate and reliable predictions, which is essential for many applications, such as image recognition, speech recognition, natural language processing, and recommendation systems.
3. **Model evaluation:** Evaluate the model's performance using appropriate evaluation metrics such as accuracy, precision, recall, and F1 score. Model evaluation is a critical step in machine learning, where the objective is to assess the performance of a trained model on unseen data. The goal of model evaluation is to ensure that the model can generalize well to new data and produce accurate predictions. The evaluation process involves applying the trained model to a validation dataset that is separate from the training dataset, and measuring its performance using appropriate evaluation metrics, such as accuracy, precision, recall, or F1 score. The choice of evaluation metric depends on the specific application and the nature of the problem being solved. Model evaluation is an iterative process that may require adjusting the hyperparameters of the model and the data preprocessing steps to optimize the model's performance. Additionally, model evaluation can be used to compare the performance of different models and to select the best model for a specific application. It is essential to carefully evaluate a model's performance to ensure that it can be deployed in real-world applications with confidence.
4. **Model deployment**: Deploy the trained model to predict customer churn on new data. Model deployment is the process of making a trained machine learning model available for use in a production environment. It involves taking the trained model and integrating it into a system or application that can be used by end-users. Depending on the application, the deployment environment may need to be optimized for performance, reliability, and security. The deployment process typically involves converting the trained model into a format that can be used by the deployment environment and integrating it with the application's codebase. Once the model is deployed, it is essential to monitor its performance and continuously update it to ensure that it remains accurate and effective. Model deployment is a critical step in the machine learning pipeline, as it is where the model is put into action and can provide real-world value.

## CHAPTER 2

**LITERATURE SURVEY**

At the moment, classical statistics-based forecasts and predictions based on integrated classifiers are both used in domiciliary and global users turnover prediction algorithms. In order to estimate customer attrition using machine learning techniques and statistical theory. employed consumer visual insights to find relations between indicators[1]. MGUIIS and CO. created a predictive model using logistic regression depending on how long retail customers spend on average per transaction. The augmented decision tree model was used by Du Gang and Huang Zhenyu to anticipate the buying habits of consumers[2]. To confirm the usefulness of the new technique in predicting consumer buying behavior, they compared the effects of their analysis before and after optimization.

The customer retention analysis was conducted by the authors using a logistic regression model. Age, gender, the kind of registration and length of service use, the type of phone, and the monthly price are used to train and evaluate the model. This initial step's test results show 74% correctness. The accuracy level of this model increased to 79% after researchers supplemented the prior data set with subscriber internet usage data[8]. Lastly, at the end of this investigation, researchers demonstrated that combining the two distinct data sets indicated above can significantly raise the level of accuracy[9]. My opinion is that when looking at the churn prediction model, they neglected to consider several crucial elements that could affect subscriber decision-making processes, such as the most recent packages utilized by customers, their happiness with customer support, etc[11]. So, this model cannot be used in the future to identify client turnover causes. In any case, this research is beneficial.

In the Improved Churn Prediction Method, SNA ideas are used to forecast the churn of telecom consumers[3]. The method entails three steps: quantifying tie strength, applying machine learning techniques to combine traditional and social variables, and influence propagation model[13]. Before clients cancel the service, a pattern analysis framework is suggested to give strategic planners advice. The chat graph approach of churn prediction concentrates on forecasting the churn in the conversation activity[5]. This methodology does not take into account the social aspects derived from graph theory. According to their online actions, users are divided into groups for churn prediction using the clustering method, which then applies retention rules to keep them from leaving[15]. The Churn Prediction by Local Community Detection approach uses a greedy scheme to divide the network into communities. This approach disregards the network's structure and content. Identification of potential users inside an operator's network is the main emphasis of Churn Prediction by Utilizing the Diffusion Process. The network's diffusion process can be directed using graph theory.

1. Logistic Regression Model and Its Application [1]

Authors: C. Zhenhai and Liu Wei

The paper "Logistic Regression Model and Its Application" by C. Zhenhai and Liu Wei provides an overview of the logistic regression model and its applications in various fields, including medicine, economics, and social sciences. The authors explain the mathematical foundation of logistic regression and its relationship with linear regression. They also discuss the assumptions and limitations of the logistic regression model. The authors provide examples of logistic regression models used in different applications, such as predicting the likelihood of a patient having a disease based on their medical history and predicting customer churn in a telecommunications company. The paper highlights the advantages of logistic regression, such as its ability to handle categorical and continuous predictors and to provide interpretable results. The authors also discuss some of the challenges associated with logistic regression, such as overfitting and multicollinearity. The paper provides a valuable introduction to the logistic regression model and its applications, making it useful for researchers and practitioners in various fields.

1. L\_(1/2) Regularized Logistic Regression [2]

Authors: Qian, Meng. Deyu, Xu. Zongben

The paper "L\_(1/2) Regularized Logistic Regression" by Z. Qian, Meng. Deyu, and Xu. Zongben introduces a novel regularization method for logistic regression, called L\_(1/2) regularization. The authors propose this method to address the limitations of L1 and L2 regularization, such as their inability to handle high-dimensional data with a large number of irrelevant features. The L\_(1/2) regularization method combines the advantages of L1 and L2 regularization by penalizing the absolute value of the weight vector and the square of the weight vector. The authors demonstrate the effectiveness of L\_(1/2) regularization on synthetic and real-world datasets and compare it with other regularization methods, such as L1, L2, and elastic net regularization. The results show that L\_(1/2) regularization outperforms the other methods in terms of classification accuracy and feature selection. The paper provides a valuable contribution to the field of logistic regression regularization and has practical applications in various domains, including bioinformatics, finance, and image analysis.

1. Customer-Churn Research Based on Customer Segmentation [3]

Authors: X. Zhang, G. Feng and H. Hui

The paper "Customer-Churn Research Based on Customer Segmentation" by X. Zhang, G. Feng and H. Hui proposes a method to reduce customer churn by segmenting customers based on their behavior and characteristics. The authors use a dataset of customer interactions and demographic information to cluster customers using the K-means clustering algorithm. The authors then analyze the characteristics and behavior of each cluster to identify potential reasons for churn. The authors propose a method of using decision trees to identify the most important factors contributing to churn in each cluster. The authors evaluate the performance of their approach using a case study, and the results show that their method effectively reduces customer churn. The paper provides a valuable contribution to the field of customer churn prediction and customer segmentation. The proposed method has practical applications in customer retention, marketing, and customer relationship management. The paper highlights the importance of understanding customer behavior and characteristics to reduce customer churn and improve customer satisfaction.

1. Telecom customer churn prediction method based on cluster stratified sampling logistic regression [4]

Authors: Peng Li, Siben Li, Tingting Bi and Yang Liu

The paper "Telecom Customer Churn Prediction Method Based on Cluster Stratified Sampling Logistic Regression" by Peng Li, Siben Li, Tingting Bi, and Yang Liu proposes a method to predict customer churn in the telecommunications industry using a combination of cluster stratified sampling and logistic regression. The authors use a dataset of customer interactions and demographic information to create a stratified sample of customers based on their behavior and characteristics. They then use logistic regression to build a predictive model of customer churn. The authors propose a novel method of combining the results of multiple logistic regression models trained on different strata of customers to improve the accuracy of the churn prediction model. The authors evaluate the performance of their approach using a case study and compare it with other methods, such as decision trees and neural networks. The results show that their method outperforms the other methods in terms of classification accuracy and feature selection. The proposed method has practical applications in customer retention, marketing, and customer relationship management in the telecommunications industry. The paper provides a valuable contribution to the field of customer churn prediction and highlights the importance of stratified sampling and model combination in improving the accuracy of churn prediction models.

1. Customer Churn Prediction Using Data Mining Techniques for an Iranian Payment Application [5]

Authors: O. Rezaeian, S. S. Haghighi and J. Shahrabi

The paper "Customer Churn Prediction Using Data Mining Techniques for an Iranian Payment Application" by O. Rezaeian, S. S. Haghighi, and J. Shahrabi presents a case study of customer churn prediction using data mining techniques for an Iranian payment application. The authors use a dataset of customer transactions and demographic information to build a predictive model of customer churn using several data mining techniques, including decision trees, random forests, and support vector machines. The authors compare the performance of these techniques in terms of accuracy, precision, recall, and F1 score. The results show that random forests outperform the other techniques in terms of accuracy and F1 score. The authors also use feature importance analysis to identify the most important features affecting customer churn. The paper provides a practical example of how data mining techniques can be applied to customer churn prediction in the context of a real-world application. The authors highlight the importance of selecting appropriate data mining techniques and feature selection methods to improve the accuracy of churn prediction models. The paper has practical implications for businesses in various domains, including finance, e-commerce, and telecommunications, that rely on customer retention for their success.

1. Churn Prediction by Finding Most Influential nodes in Social Network [14]

Authors: Reena Pagare and Dr. Akhil Khare

The paper "Churn Prediction by Finding Most Influential nodes in Social Network" by Reena Pagare and Dr. Akhil Khare proposes a method to predict customer churn in social networks by identifying the most influential nodes. The authors use a dataset consisting of customer interactions in a social network to create a graph-based representation of the network. They then use centrality measures, such as degree centrality and betweenness centrality, to identify the most influential nodes in the network. The authors propose a novel method of combining multiple centrality measures to improve the accuracy of the churn prediction model. The authors evaluate the performance of their approach using a classification algorithm, and the results show that their method outperforms traditional methods of churn prediction. The proposed method has practical applications in marketing, customer retention, and social network analysis. The paper provides a valuable contribution to the field of social network analysis and churn prediction.

## CHAPTER 3

**SYSTEM ARCHITECTURE AND DESIGN**

**3.1 System architecture**

Diagram

Description automatically generated

**Fig.3.1 Architecture Of The Proposed System**

The system architecture for Customer Churn Prediction using Machine Learning typically involves several components that work together to build, train, and deploy a churn prediction model.

The first component is data acquisition, which involves collecting and preprocessing customer data from various sources, such as transaction records, customer feedback, and demographic information. The data is then cleaned, transformed, and prepared for modeling. The next element is feature engineering, which entails choosing and modifying pertinent features from the data to raise the predictive model's accuracy. Techniques like feature selection, feature scaling, and dimensionality reduction may be used for this.

The system architecture of customer churn prediction encompasses various components and stages to effectively identify and predict customer churn within a business.

At the core of the architecture is the data collection phase, where relevant data is gathered from various sources such as customer interactions, transaction history, demographic information, and customer feedback. This data is then stored in a central data repository, such as a data warehouse or a big data platform, for further processing and analysis.The next stage involves data preprocessing, where the collected data undergoes cleaning, transformation, and feature engineering. This step ensures that the data is in a suitable format for analysis and modeling. It may include tasks such as handling missing values, normalizing data, and creating new features based on domain knowledge.

Following data preprocessing, the architecture involves the model development phase. In this stage, predictive models are created using machine learning or statistical techniques. These models are trained on historical customer data, including both churned and non-churned customers, to learn patterns and identify key factors that contribute to churn. Various algorithms, such as logistic regression, decision trees, or neural networks, can be employed depending on the complexity of the problem and the available data.

Once the models are developed, they are evaluated using appropriate performance metrics such as accuracy, precision, recall, or area under the curve (AUC). This evaluation helps assess the model's effectiveness in predicting customer churn. If the model's performance is satisfactory, it can be deployed into the production environment for real-time prediction. The deployment phase involves integrating the predictive model into the existing business systems or customer relationship management (CRM) platforms. The model takes input data, such as customer attributes and behavior, and generates churn predictions for individual customers. These predictions can be used to prioritize retention efforts, allocate resources effectively, and personalize customer interactions to mitigate churn risks.

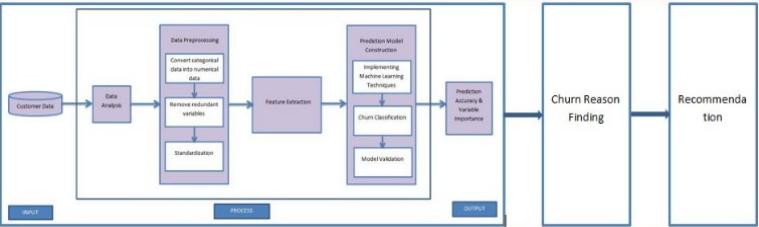
Finally, the system architecture incorporates a feedback loop to continuously improve the churn prediction model. By collecting feedback on the accuracy of the predictions and monitoring the actual churn outcomes, the model can be retrained and updated periodically to adapt to changing customer behavior and market dynamics.

In summary, the system architecture of customer churn prediction involves data collection, preprocessing, model development, evaluation, deployment, and continuous improvement. It enables businesses to proactively identify customers at risk of churn, take preventive measures, and enhance customer retention strategies.

Model selection, the third element, entails choosing the best machine learning algorithm for the task at hand. In order to do this, it may be necessary to compare various algorithms, such support vector machines, decision trees, random forests, and logistic regression, and choose the one that offers the highest performance in terms of accuracy, precision, recall, and F1 score.

The fourth element is model training, which entails applying cross-validation and hyperparameter tweaking techniques to train the chosen algorithm on the prepared data in order to maximize the model's performance. The trained model's performance is assessed on a holdout set of data as the fifth component, and this ensures that it generalizes effectively to fresh data. The last step is model deployment, which is putting the trained model into use and utilizing it to make predictions based on fresh data. In order to do this, the model might be made available as a microservice or REST API that can be included into already-existing applications. Overall, the system architecture for Customer Churn Prediction using Machine Learning involves several steps that require expertise in data preprocessing, feature engineering, machine learning algorithms, and deployment infrastructure. The system must also be scalable, reliable, and maintainable to ensure that it can handle large volumes of data and continue to provide accurate predictions over time.

**3.2 Data Flow Diagram**



**Fig.3.1 Data Flow Diagram**

A data flow diagram (DFD) for Customer Churn Prediction using Machine Learning typically includes several key components that illustrate how data is processed and flows through the system.

=A data flow diagram (DFD) is a graphical representation that illustrates the flow of data within a system. It is a powerful tool used in system analysis and design to depict the movement of data between various processes, entities, and data stores.

At the heart of a DFD are four main components: processes, data flows, data stores, and external entities. Processes represent activities or functions performed within the system. They can range from simple tasks like data input to complex operations such as data manipulation or generation of reports.

Data flows represent the movement of data between processes, data stores, and external entities. They depict the path that data follows as it is processed and transformed within the system. Arrows indicate the direction of data flow, and labels describe the nature of the data being transferred. Data stores are repositories where data is stored for future use. They can be physical entities like databases or files, or they can be virtual representations such as cloud storage or memory buffers. Data stores act as sources or destinations of data during the system's operation.

External entities are external systems, organizations, or individuals that interact with the system. They can include users, customers, suppliers, or other systems that provide input or receive output from the system. External entities are shown as squares in a DFD. By using a DFD, analysts can visualize the flow of data and identify potential bottlenecks, redundancies, or inefficiencies within a system. It helps in understanding how information is processed, stored, and communicated within the system's boundaries. Additionally, a DFD serves as a communication tool between stakeholders, enabling them to discuss and refine the system's requirements and functionality.

In summary, a data flow diagram is a valuable tool for system analysis and design. It provides a clear and concise representation of how data moves through a system, allowing for better understanding, analysis, and improvement of the system's operations.Here's an example of a DFD for such a system:

*Level 0:*

- Customer data: This input data source includes customer data such as transaction records, demographic information, and feedback.

- Preprocessing module: This module processes the raw customer data, performs data cleaning and transformations, and outputs cleaned data.

- Feature engineering module: This module processes the cleaned data, selects relevant features, and engineers new features to improve the accuracy of the prediction model.

- Machine learning module: This module takes the engineered features as input, selects and trains an appropriate machine learning algorithm, and outputs a trained model.

- Evaluation module: This module evaluates the trained model's performance on a holdout set of data and outputs performance metrics.

- Prediction module: This module takes new customer data as input, applies the trained model to predict customer churn, and outputs the churn prediction results.

- Churn analysis module: This module analyzes the churn prediction results and provides insights and recommendations to the business.

*Level 1:*

- Preprocessing module: This module includes sub-modules for data cleaning, data transformation, and data integration.

- Feature engineering module: This module includes sub-modules for feature selection, feature engineering, and feature scaling.

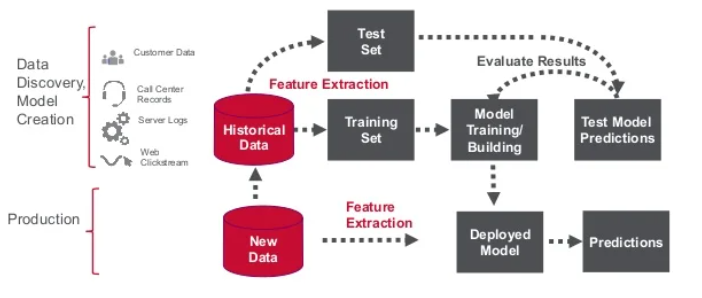
- Machine learning module: This module includes sub-modules for algorithm selection, hyperparameter tuning, and model training.

- Evaluation module: This module includes sub-modules for performance metric calculation, model comparison, and model selection.

- Prediction module: This module includes sub-modules for data preprocessing, feature engineering, and model prediction.

- Churn analysis module: This module includes sub-modules for churn segmentation, churn rate calculation, and churn prediction interpretation.

**3.3 UML Diagrams**



**Fig.3.3 UML Diagram**

Model Transformation Interpretation, or UMI In object-oriented software design, the model dialect UML is a broad sense, guidelines tool. The Object Current Management oversees and develops the specifications. The objective is for UML to establish itself as a standard modelling tool for entity software programs. There are presently two primary parts of UML: a notation and a meta-model. In the future, a process or method may also be coupled or incorporated in UML. A official usage for describing is the Universal Design Methodology. Generating, generating, and describing existing software artefacts in addition to those for business models and many other non-software platforms.

The UML incorporates the highest performing techniques that have proved effective in simulating big, complicated systems. The creation of entity software and the software development life cycle both rely heavily on the UML. The UML primarily use visual footnotes to communicate web application architecture.

*GOALS:*

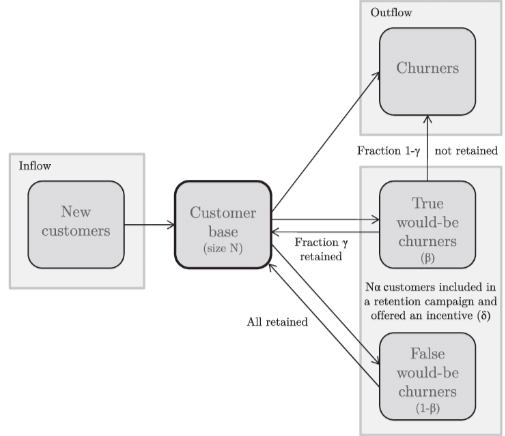
The key design goals of the UML are as follows:

1.Offer consumers a fully prepared visual programming simulation environment to enable them to build and exchange effective method.

2. Provide strategies for specialising and expanding the basic concepts.

3. neither depend on specific computer languages or methodologies

**3.3.1 Class Diagram**



**Fig.3.4 Class Diagram**

A class diagram for customer churn prediction provides a visual representation of the key classes and their relationships within the system. It helps to conceptualize the structure and organization of the components involved in predicting customer churn. At the center of the class diagram is the "Customer" class, which represents individual customers. It encapsulates attributes such as customer ID, demographic information, transaction history, and customer behavior metrics. This class serves as the main entity for churn prediction.

The "ChurnPredictionModel" class represents the predictive model used for churn prediction. It contains methods and algorithms to analyze customer data and generate churn predictions. This class may have associations with other classes, such as "DataPreprocessing" and "FeatureEngineering," to handle data preparation tasks required for the model's input. The "DataPreprocessing" class encapsulates methods and functions for cleaning, transforming, and normalizing the input data. It prepares the customer data to be used by the churn prediction model. This class may also have associations with other classes, such as "DataCollection" or "DataStorage," to retrieve and store relevant customer data. "FeatureEngineering" class focuses on creating new features from the raw customer data. It applies domain-specific knowledge and techniques to extract meaningful information that can improve the accuracy of churn predictions. This class may have associations with the "Customer" class to access and manipulate customer attributes.

The "ModelEvaluation" class assesses the performance of the churn prediction model. It includes methods to compare the model's predictions with actual churn outcomes and calculates evaluation metrics such as accuracy, precision, recall, or AUC. This class provides insights into the model's effectiveness and aids in further model refinement. Additional classes such as "ModelDeployment," "FeedbackLoop," or "Visualization" can be included in the class diagram to depict other aspects of the churn prediction system. These classes handle tasks related to deploying the model into production, gathering feedback on predictions, and visualizing churn-related insights for stakeholders.

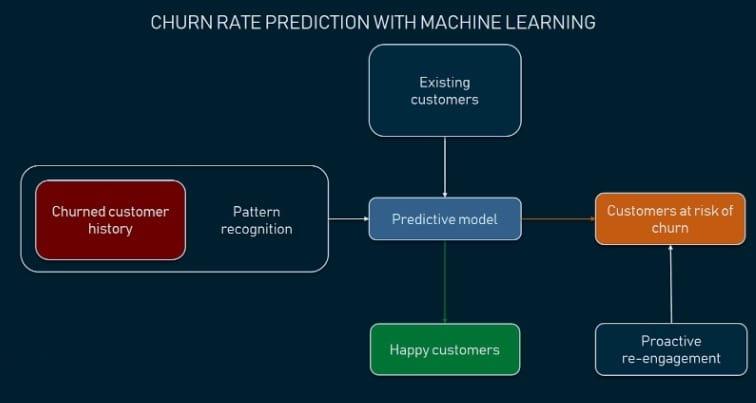
Overall, the class diagram for customer churn prediction illustrates the key classes and their relationships within the system. It captures the data flow and interactions between different components involved in predicting customer churn, enabling a better understanding of the system's structure and facilitating effective implementation and maintenance of the churn prediction system.

The class diagram, which also provides a complete system architecture, is used to polish the use case diagram. The class diagram divides the players in the use case diagram into a number of related classes. There can be a connection between the classes. Each class in the class diagram might offer a different set of functionalities. These functions that the class provides are referred to as its "methods." Each class may also have unique "attributes" that set them apart from other classes. Figure 3.3.1 shows the class diagram in its entirety.

The Customer class represents the customer data and contains attributes such as id, age, gender, balance, and service. The Churn class represents the churned data and contains attributes such as whether the customer has churned, contract, payment, and prediction.

The Preprocessor class is responsible for cleaning and scaling the customer data before it is fed into the machine learning model. The FeatureEngineer class is responsible for transforming and selecting features from the preprocessed data. The Model class is responsible for training the machine learning model and making predictions on the data. Finally, the Evaluation class is responsible for evaluating the performance of the model using a confusion matrix and classification metrics.

**3.3.2 Use Case Diagram**

****

**Fig.3.5 Use case diagram**

A use case diagram in UML is used to show the myriad ways that a client might interact with such a technology. The use case diagram is shown in Fig.3.3.2.

A use case diagram is a type of UML diagram that illustrates the interactions between the actors and the system in different scenarios or use cases. In the context of Customer Churn Prediction using Machine Learning, the use case diagram represents the main functionalities that the system offers to the users or actors, including business users and data administrators. The diagram depicts two primary use cases, which are Train Model and Predict Churn.

The Train Model use case involves the process of building a machine learning model using a training dataset to predict customer churn. The data preprocessing, feature engineering, and model building are performed in this use case. The Predict Churn use case represents the process of utilizing the trained model to predict customer churn for new data or new customers. The input data is processed by the same preprocessor and feature engineer used in the Train Model use case.

The use case diagram also depicts two actors, including business users and data administrators. Business users are the primary users who utilize the system to predict customer churn and take appropriate actions. On the other hand, data administrators manage the system's data and ensure data quality, including data collection, cleaning, and preprocessing.

**3.3.3 Activity Diagram**

Diagram

Description automatically generated

**Fig.3.6 Activity diagram**

An activity diagram, as that the term suggests, displays the various stages that framework objects go through throughout the course of their life. The statuses of object types change as a result of occurrences. A state diagram may also illustrate how such an object's condition transitions from its initial state to its final state as a result of scheme occurrences. Figure 3.3.3 shows the state diagram in its entirety. The activity diagram displays the system's operation activities. The very same elements found in a state diagram are also included in an activity diagram, such as activities, operations, transfers, starting and ending stages, and security elements.

An activity diagram is a type of UML diagram that represents the flow of activities or processes in a system. For Customer Churn Prediction using Machine Learning, an activity diagram can be used to depict the steps involved in predicting customer churn. The diagram would typically begin with the collection of customer data, which would then be preprocessed and transformed through feature engineering. Next, a model would be trained using the processed data. Once the model has been trained, it can be used to predict customer churn on new data. The predictions can then be evaluated for accuracy and used to make business decisions.

The activity diagram can also show decision points, where the flow of activities depends on the results of a previous activity. For example, if the evaluation of the model shows that the accuracy is not high enough, the model may need to be retrained or the feature engineering process may need to be adjusted. This decision point can be represented in the activity diagram as a diamond-shaped symbol. Overall, an activity diagram for Customer Churn Prediction using Machine Learning can provide a visual representation of the steps involved in the process, helping to identify potential bottlenecks or areas for improvement.

## 

## 

## 

## CHAPTER 4

**METHODOLOGY**

**4.1 Existing System**

The existing system for Customer Churn Prediction using Machine Learning may vary depending on the organization, industry, and the available data. However, in general, organizations may have a traditional approach to churn prediction, which involves manual data analysis, a small number of features, and simple statistical models. This approach may lack accuracy and efficiency, leading to poor business decisions and customer retention strategies.

Some organizations may have a basic machine learning model in place, trained on limited and outdated data with a fixed set of features. This approach may not consider dynamic changes in customer behavior and preferences, leading to less accurate predictions.

Other organizations may have implemented more advanced machine learning models, which utilize a large and diverse set of customer data, including demographic information, transaction history, web behavior, customer support interactions, and social media sentiment analysis. These models may use deep learning algorithms, such as neural networks and decision trees, to make accurate and dynamic churn predictions.

The existing system of customer churn prediction typically involves a combination of data collection, data analysis, and predictive modeling techniques. Businesses gather relevant data from various sources, such as customer interactions, transaction history, and demographic information. This data is stored and organized in databases or data warehouses.

Data preprocessing is performed to clean and transform the collected data, ensuring its quality and suitability for analysis. This step involves handling missing values, removing outliers, and normalizing data to create a consistent and reliable dataset. Once the data is prepared, businesses employ predictive modeling techniques to develop churn prediction models. These models utilize machine learning algorithms or statistical approaches to identify patterns, correlations, and factors that contribute to customer churn. Algorithms such as logistic regression, decision trees, random forests, or neural networks are commonly used to train the models on historical customer data.

The performance of the churn prediction models is evaluated using various metrics, such as accuracy, precision, recall, or AUC. This evaluation helps determine the model's effectiveness in accurately predicting customer churn. If the model meets the desired performance thresholds, it is integrated into the business systems or customer relationship management (CRM) platforms for real-time churn prediction.

In the existing system, businesses often monitor the churn predictions and compare them with the actual churn outcomes to assess the model's performance. This feedback loop allows for continuous improvement of the churn prediction models by retraining and updating them based on the latest data and customer behavior. The existing system may also incorporate features like visualization tools or dashboards to provide insights into churn-related trends, customer segments at high risk of churn, and the effectiveness of retention strategies. These visualizations help stakeholders understand the churn dynamics and make informed decisions to reduce churn and improve customer retention.

Overall, the existing system of customer churn prediction involves data collection, preprocessing, predictive modeling, performance evaluation, and continuous improvement. It enables businesses to identify customers at risk of churn, take proactive measures to retain them, and optimize their retention strategies to enhance customer loyalty and satisfaction.

However, even with advanced machine learning models, some organizations may face challenges in data quality, data security, and model interpretability, which can impact the reliability and acceptance of the churn predictions. Therefore, it is important to continuously monitor and improve the machine learning models for churn prediction to ensure optimal performance and customer satisfaction.

**4.2 Proposed System**

Predictive churning model is a tool for classifying, a system that examines the traits of potential consumers to determine what traits are essential in forecasting turnover rates. Let's imagine we have a dataset with information on ten thousand clients who are taking money out of a bank[1]. These clients' characteristics, including their country of residence, credit score, age, and balance, among others, are described in the data.The outcome of the user's turnover should be predicted by our model. Hence, the target variable will be terminated. The data should be examined with an emphasis as to how various aspects connect to the customer churn status[14].

We are prepared to construct many models in search of the optimum fit. Forecasting customer turnover is a problem of binary classification since clients can leave or stay for a predetermined amount of time.

We’ll test:

* **Logistic regression classifier**

In logistic regression, the input variables are first transformed using a logistic function to produce an output that falls between 0 and 1, which can be interpreted as the probability of a particular class. The logistic function is used to model the relationship between the input variables and the probability of occurrence of the event. The logistic function takes the form:

P(y=1|x) = 1 / (1 + e^(-z))

Where P(y=1|x) is the probability of the positive class, x is the input vector, and z is the dot product of the input vector with the model weights.

The logistic regression model is trained by minimizing a loss function, which is typically the log loss or cross-entropy loss. The weights of the model are updated iteratively using an optimization algorithm, such as gradient descent or stochastic gradient descent, until the loss is minimized.

Logistic Regression is a simple and efficient algorithm that is easy to implement and interpret. It is often used as a baseline model for classification problems and can be useful for problems with linearly separable classes. However, it may not perform well on non-linearly separable problems or problems with highly correlated features. In such cases, more complex models such as decision trees, random forests, or neural networks may be more appropriate.

* **Naive Bayesian**

Naive Bayes is a machine learning algorithm based on the Bayes theorem of probability. It is a probabilistic algorithm that uses the conditional probability of features to classify data into different categories. Naive Bayes is commonly used for text classification and spam filtering, but it can also be used in other classification tasks such as sentiment analysis, recommendation systems, and customer churn prediction. The algorithm works by calculating the probability of each feature given a class label and then multiplying all these probabilities to get the probability of a data point belonging to a particular class. The class with the highest probability is then assigned as the prediction for the data point.

Naive Bayes is a probabilistic machine learning algorithm commonly used for classification tasks. It is based on Bayes' theorem and assumes that the features are conditionally independent given the class label. Despite its simplicity and naive assumption, Naive Bayes often performs remarkably well and is widely used in various applications such as spam filtering, sentiment analysis, and document categorization.

The algorithm is called "naive" because it assumes that the presence or absence of a particular feature is independent of the presence or absence of any other feature, given the class label. This assumption allows for simplified calculations and efficient training.

During the training phase, Naive Bayes calculates the probabilities of each feature given each class label by counting occurrences in the training data. It estimates the prior probabilities of each class label based on the frequency of their occurrences. These probabilities are then combined using Bayes' theorem to calculate the posterior probability of each class label given the observed features.

During the prediction phase, Naive Bayes uses the calculated probabilities to determine the most likely class label for a new instance. It calculates the posterior probabilities for each class label and selects the label with the highest probability as the predicted class.

Naive Bayes has several advantages. It is computationally efficient and works well with large datasets. It can handle high-dimensional feature spaces and is robust to irrelevant features, as the independence assumption allows it to disregard irrelevant correlations. Naive Bayes is also less prone to overfitting, especially when the training data is limited. Despite its simplicity, Naive Bayes performs well in many real-world scenarios. However, the assumption of feature independence can limit its effectiveness in cases where there are strong dependencies among the features. In such cases, more sophisticated algorithms may be more appropriate. Additionally, Naive Bayes is sensitive to the presence of rare or unseen feature combinations in the training data, which can result in zero probabilities and affect the accuracy of predictions.

In summary, Naive Bayes is a simple yet effective probabilistic algorithm used for classification tasks. Its efficiency, ability to handle high-dimensional data, and robustness to irrelevant features make it a popular choice in various applications. However, its assumption of feature independence may limit its performance in certain scenarios.

One of the strengths of Naive Bayes is that it requires a relatively small amount of training data to estimate the parameters needed for classification. However, it can be sensitive to irrelevant or correlated features, and its assumption of independence may not hold in some real-world applications.

* **Decision Tree**

Naive Bayes is a machine learning algorithm based on the Bayes theorem of probability. It is a probabilistic algorithm that uses the conditional probability of features to classify data into different categories. Naive Bayes assumes that all features are independent of each other, hence the term "naive". This assumption simplifies the calculation of probabilities, and the algorithm is computationally efficient and fast. Naive Bayes is commonly used for text classification and spam filtering, but it can also be used in other classification tasks such as sentiment analysis, recommendation systems, and customer churn prediction.

The algorithm works by calculating the probability of each feature given a class label and then multiplying all these probabilities to get the probability of a data point belonging to a particular class. The class with the highest probability is then assigned as the prediction for the data point.

One of the strengths of Naive Bayes is that it requires a relatively small amount of training data to estimate the parameters needed for classification. However, it can be sensitive to irrelevant or correlated features, and its assumption of independence may not hold in some real-world applications.

* **Kernel SVM**

Kernel Support Vector Machine (SVM) is a popular classification algorithm in machine learning that can be used for both linear and non-linear data. It works by finding the hyperplane that maximizes the margin between the two classes in the dataset. In kernel SVM, the data is transformed into a higher dimensional space using a kernel function, such as a radial basis function (RBF) or polynomial function, to make it easier to separate the classes. The transformed data is then used to find the optimal hyperplane.

Kernel Support Vector Machines (SVM) is a powerful machine learning algorithm that has gained popularity due to its ability to handle non-linearly separable data. SVMs are binary classifiers that aim to find an optimal hyperplane to separate data points belonging to different classes. However, in cases where the data is not linearly separable, the kernel trick comes into play.

Kernel SVM extends the capabilities of traditional SVMs by transforming the input data into a higher-dimensional feature space, where it becomes linearly separable. The kernel function plays a crucial role in this process by efficiently mapping the data points into the desired space. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid. The kernel trick allows the SVM algorithm to operate in the original input space, avoiding the need for explicit computation in the higher-dimensional feature space. This makes kernel SVM computationally efficient, even for complex data.

One of the key advantages of kernel SVM is its ability to capture intricate decision boundaries, enabling it to handle non-linear relationships in the data. The RBF kernel, in particular, is widely used and exhibits excellent performance across various domains. Kernel SVMs are robust against overfitting as they focus on maximizing the margin between support vectors rather than attempting to fit every training point precisely. Support vectors are the data points closest to the decision boundary and are critical for determining the optimal hyperplane.

Despite its strengths, kernel SVMs have some considerations. Choosing an appropriate kernel function and tuning its parameters can be challenging, requiring careful experimentation. Additionally, kernel SVMs can be computationally demanding, especially with large datasets, as the training complexity increases with the number of support vectors. In summary, kernel SVM is a versatile algorithm that leverages the kernel trick to handle non-linear data effectively. Its ability to capture complex decision boundaries makes it a valuable tool in various machine learning tasks, including classification and regression. However, proper kernel selection and parameter tuning are crucial for achieving optimal performance.

Kernel SVM is useful when the data is not linearly separable and there are complex decision boundaries between the classes. It has been widely used in various fields, including image classification, text classification, and bioinformatics.

* **KNN**

KNN, or k-nearest neighbors, is a classification algorithm that is based on the idea of finding the k nearest data points in the feature space to the point being classified. The algorithm then assigns the class that appears most frequently among the k nearest neighbors to the point being classified.

The k value in KNN is a hyperparameter that needs to be set before running the algorithm. A smaller value of k will result in a more flexible decision boundary, which is more sensitive to noise in the data, while a larger value of k will result in a smoother decision boundary that is less sensitive to noise in the data.

KNN is a simple and effective algorithm that can be used for both classification and regression problems. However, it can be computationally expensive, especially when dealing with large datasets, as it requires computing distances between each data point and every other data point in the dataset. KNN also requires careful normalization of the feature values to ensure that features with larger scales do not dominate the distances calculated.

K-Nearest Neighbors (KNN) is a simple yet effective algorithm in machine learning that is widely used for both classification and regression tasks. KNN is a non-parametric algorithm, meaning it does not make any assumptions about the underlying data distribution.

The basic idea behind KNN is to classify or predict a new data point based on its proximity to the K nearest neighbors in the training set. The "K" in KNN represents the number of neighbors to consider. The algorithm assumes that similar instances in the feature space tend to have similar labels or target values. During the classification task, KNN calculates the distance between the new data point and all other data points in the training set using a distance metric such as Euclidean distance or Manhattan distance. It then selects the K nearest neighbors based on the shortest distances. The class label of the new data point is determined by majority voting among its K nearest neighbors. In regression tasks, KNN predicts the target value by averaging the values of its K nearest neighbors.

KNN is a lazy learning algorithm, meaning it does not explicitly build a model during the training phase. Instead, it stores all the training data and performs computations at the prediction time. This makes the training process faster, but the prediction can be computationally expensive, especially for large datasets. One of the advantages of KNN is its simplicity. It does not assume any underlying data distribution, making it suitable for a wide range of datasets. KNN can handle both numerical and categorical data, making it a versatile algorithm. It is also robust to outliers since it relies on the majority vote or average of the nearest neighbors. Additionally, KNN does not require the tuning of hyperparameters or the need for extensive training.

However, KNN has some considerations. It can be sensitive to the choice of the number of neighbors (K) and the distance metric, and selecting appropriate values for these parameters is crucial for good performance. The algorithm can also suffer from the curse of dimensionality, where the distance-based calculations become less meaningful as the number of dimensions increases. In summary, K-Nearest Neighbors (KNN) is a simple and intuitive algorithm that relies on the proximity of training instances to make predictions. Its versatility, robustness to outliers, and ease of implementation make it a popular choice in various machine learning tasks. However, careful parameter selection and potential scalability issues should be considered when applying KNN to real-world scenarios.

* **Random Forest**

Random Forest is a popular ensemble learning technique that builds multiple decision trees and combines their results to make more accurate predictions. It randomly selects subsets of the training data to build individual decision trees, and each tree is trained using a different subset of the features. The predictions of the individual trees are then combined to make the final prediction. The randomization helps to prevent overfitting and makes the model more robust to noise in the data. Random Forest is commonly used for classification and regression problems and has been shown to be effective in a wide range of applications, including customer churn prediction, fraud detection, and image classification.

Random Forest is a popular and powerful ensemble learning algorithm that combines the predictions of multiple decision trees to produce accurate and robust results. It is widely used for both classification and regression tasks in machine learning. The fundamental idea behind Random Forest is to build an ensemble of decision trees, where each tree is trained on a randomly sampled subset of the original data, and the final prediction is obtained by averaging or voting the predictions of individual trees.

The randomness in Random Forest comes from two main sources: random sampling of the training data and random feature selection. By using random sampling, each tree in the forest is trained on a different subset of the data, which helps to introduce diversity and reduce overfitting. Random feature selection ensures that each split in a decision tree is based on only a subset of features, further enhancing the diversity among trees.

During the training process, each decision tree in the Random Forest is grown by recursively partitioning the data based on different features and their respective thresholds. The splits are chosen to optimize a specific criterion, such as Gini impurity or information gain, which measure the homogeneity of the target variable within each resulting subset. Random Forest offers several advantages over individual decision trees. It is less prone to overfitting, thanks to the ensemble approach and the randomness injected during training. It also handles high-dimensional data well and can provide estimates of feature importance, which can be useful for feature selection.

Moreover, Random Forest is robust to noisy and missing data, as it considers multiple trees and aggregates their predictions. It can handle both categorical and numerical features without requiring extensive data preprocessing. The algorithm is also efficient for large datasets, as the training process can be parallelized. One of the key strengths of Random Forest is its ability to provide reliable predictions by reducing bias and variance. It is known for its high accuracy, stability, and resistance to outliers. It is a versatile algorithm that has found applications in various domains, including finance, healthcare, and natural language processing.

In summary, Random Forest is a powerful ensemble learning algorithm that combines the predictions of multiple decision trees to provide accurate and robust results. Its ability to handle high-dimensional data, handle missing values, and produce feature importance estimates makes it a valuable tool in machine learning. With its versatility and strong predictive capabilities, Random Forest has become a popular choice for a wide range of applications.

* **Support Vector Machine with Radial basis function kernel**

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for classification or regression tasks. The Radial Basis Function (RBF) kernel is one of the most commonly used kernels in SVM. It maps the input data to a higher dimensional space and makes it possible to separate the data points using a hyperplane. The RBF kernel is defined by a distance metric, which measures the similarity between two data points. It is a popular choice for SVM because it is capable of modeling complex decision boundaries and can handle non-linearly separable data.

In SVM, the goal is to find the hyperplane that separates the data points into their respective classes with the maximum margin. The margin is the distance between the hyperplane and the closest data points from each class. SVM tries to maximize this margin so that it can generalize well to unseen data. The RBF kernel in SVM calculates the distance between data points in the higher dimensional space, which allows for more complex decision boundaries.

One disadvantage of SVM with RBF kernel is that it can be sensitive to the choice of hyperparameters, such as the regularization parameter (C) and the kernel parameter (gamma). The choice of these parameters can affect the performance of the model and can be a challenge for some datasets. However, with proper tuning of these parameters, SVM with RBF kernel can be a powerful tool for classification tasks.

These models need to be worked on and we’ll do so using the the given steps:

- Search for Parameters: We'll choose the parameters and values we want to look for in each of our models. The best parameters found in our model will be set when we run the GridSearchCV.

- Best Models Fit: We train the system using the train dataset after determining the best estimator.

- Performance Evaluation: Using our test set, we will evaluate the models that performed the best after being trained on our training dataset.

**4.3 Data Retrieval Process**

The word "read" describes the process of getting data from a storage device. Data retrieval in databases is the method of finding and attempting to remove data from a database based on a user- or application-provided query. It enables the retrieval of data from a database for display on a monitor and use in a program.Shows the Generalised block diagram for the proposed system.The audio processing program Audio Signal Synthesis , Recovery, and Music Analysis has some significance for music recovery applications and is freely available.The database is accessible through its internet website as well. Ten genres make up the dataset we utilised, and we split it into training and testing sets.. We have 70% of the data in the training area, and we have about 30% of the information in the test section.

We develop our algorithm using a data training set, and then we use it to forecast the genre of music sound in a test dataset. During testing, we evaluate the algorithm's accuracy since we are familiar with how it operates.

* 1. **Implementation**

1. **Data Visualisation**

Substantial discoveries from your data can aid the company's development. However, the problem is that one cannot necessarily draw conclusions from simply looking at the data. Patterns, links, and other astounding revelations that might not otherwise be obvious become evident when you visually analyze your information[11].You develop storytelling skills by using data visualization to bring your data to life and reveal the hidden meanings. Through live data visualizations, dynamic reports, graphical displays, and various other illustrations, data visualization allows consumers to swiftly and efficiently develop compelling business insights.

Calendar

Description automatically generated

**Figure 4.1. Histogram of numerical data**

We can draw a bunch of conclusions based on the histograms represented in Figure 2.

Data visualization is the process of presenting data in a visual format, such as charts, graphs, or maps, to facilitate understanding, analysis, and communication of information. It transforms complex datasets into visual representations that are more accessible and intuitive for humans to comprehend.

Through data visualization, patterns, trends, and relationships within the data can be easily identified and interpreted. It allows individuals to explore and gain insights from the data by visually examining the distributions, variations, and correlations between different variables. By presenting data visually, it becomes easier to spot outliers, detect patterns, and make data-driven decisions. Various types of visualizations can be employed depending on the nature of the data and the intended purpose. Commonly used visualizations include bar charts, line charts, scatter plots, pie charts, histograms, heatmaps, and geographical maps. Each type of visualization serves a specific purpose in representing different aspects of the data, such as comparing values, showing trends over time, displaying the composition of categories, or illustrating spatial patterns.

Data visualization plays a crucial role in data analysis and decision-making across numerous domains, including business, finance, healthcare, marketing, and research. It enables stakeholders to gain a holistic view of complex datasets and effectively communicate insights to a wide range of audiences. Moreover, interactive data visualizations allow users to interact with the data and customize the visual representations based on their needs. They can zoom in, filter, and manipulate the data to explore specific aspects or drill down into details. This interactivity enhances the user's engagement and promotes a deeper understanding of the data.

In summary, data visualization is a powerful tool for transforming data into meaningful and actionable insights. It simplifies complex information, uncovers patterns, and facilitates effective communication of data-driven findings. By leveraging visual representations, individuals can make informed decisions, drive innovation, and gain a deeper understanding of the underlying data.

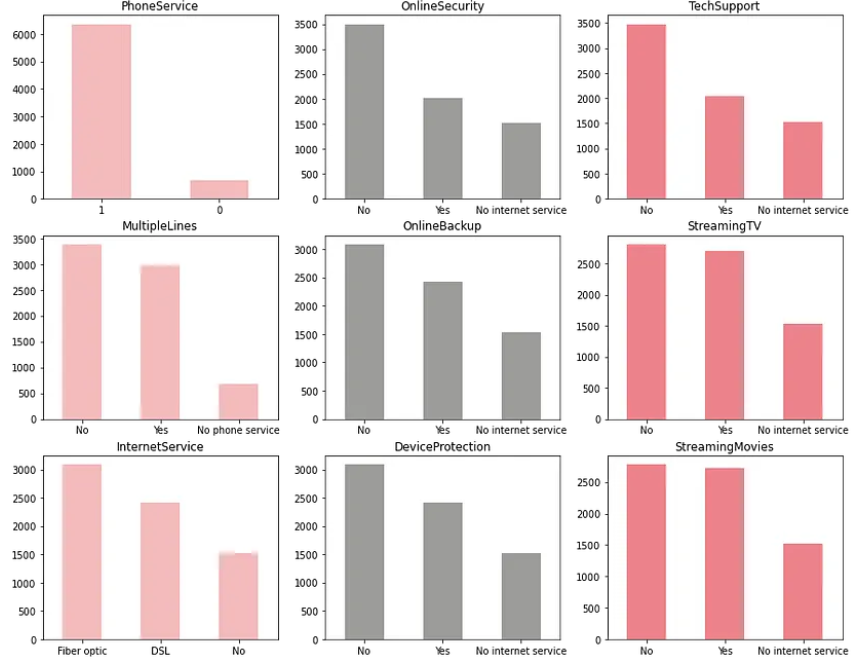
According to the dataset's gender distribution, there are roughly equal numbers of male and female clients. In our dataset, we have an equal number of men and women.

Younger clients make up the majority of the dataset's customers.

While roughly 50 percent of the users are sharing the plan with their partner, it seems that few customers have dependents.

The company has a large number of recent clients, along with a devoted consumer division which are constant for, on average, more than 70 months.

Most customers seem to require access to a cellphone, and 75 percent of them prefer transactions without paper. Invoice fees per user each month vary between eighteen dollars to one hundred and eighteen dollars, with a large majority of consumers falling into the twenty dollars bracket.



**Figure 4.2. Distribution of Label Encoded Categorical Variables**

From the plots of Figure 3, we learn several things:

1. Nearly half of consumers have several lines of service, while almost all of consumers have a connection to cell phone service.
2. More than half of internet users watch TV shows and movies online, and 3/4 of customers choose fiber-optic and DSL lines for their internet access.
3. Only a tiny number of consumers have used the safety measures, technological help, and internet backup features.

The plots teach us various things, including:

a. Churning customers are older than those who are kept.

b. There is no distinction between lost and maintained clients in terms of the median credit score or tenure.

c. The majority of the clients that leave the business appear to still have a sizable sum in their bank accounts.

d. Customer churn appears to be unaffected by expected wage or the number of items.

We need to undertake some feature engineering before we search for a model to forecast customer turnover.

Several of the characteristics in the dataset are clubbed together to make the latest characteristics that more accurately characterize our clients. While credit scores usually rise over time (and subsequently with age), as we previously observed, they have no impact on churning, we will develop a new feature to take this into consideration.

1. **Exploratory Data Analysis**

Exploratory Data Analysis (EDA) is a critical step in the data analysis process that focuses on gaining an initial understanding of the data before applying more advanced techniques. It involves the systematic exploration, visualization, and summarization of the dataset to extract insights and identify patterns, trends, and outliers.

During EDA, data analysts employ a variety of techniques and visualizations to uncover important characteristics and relationships within the data. They examine the distribution of variables through histograms, box plots, or density plots to understand the data's overall shape, central tendency, and dispersion. They also use scatter plots or correlation matrices to assess the relationships between different variables and identify potential dependencies or associations.

EDA also involves examining the presence of missing values, outliers, or inconsistent data and deciding how to handle them. Analysts may impute missing values using appropriate techniques or remove outliers that can significantly impact the analysis. Cleaning and preprocessing steps are performed to ensure the data is in a suitable format for further analysis.

Moreover, EDA enables the identification of relevant subsets or subgroups within the data. By segmenting the dataset based on different categorical variables, analysts can compare and contrast patterns and relationships within each subgroup, leading to more targeted and insightful analyses.

Through exploratory data analysis, analysts can generate hypotheses and formulate research questions for further investigation. They can identify variables that are most influential in explaining the outcome of interest, which helps guide feature selection in subsequent modeling efforts. EDA also helps detect potential biases, anomalies, or data quality issues that may impact the reliability of subsequent analyses.

Overall, exploratory data analysis is a fundamental step in data exploration and understanding. It provides insights into the dataset's characteristics, uncovers patterns and relationships, and informs subsequent modeling or analysis decisions. By visualizing and summarizing the data, analysts can make informed decisions, generate hypotheses, and extract valuable insights from complex datasets.

Key findings of EDA performed on this model

* There are no incorrect or missing data values in the dataset.
* Monthly Charges and Age have the strongest positive correlations with the goal qualities, whereas Partner, Dependents, and Tenure have the strongest negative correlations.
* Due to a lot of consumers having motion, the dataset is not balanced.
* Relationship between Monthly-Charges and Total-Charges is multicollinear. The VIF values have significantly lowered as a result of dropping Total Charges.
* Younger clients make up the majority of the dataset's customers.
* A substantial portion of recent customers (those under a year old) make up the majority of the company's clientele, which is accompanied by a loyal clientele that is older than 70 months.
* The majority of users appear to have phone service, with monthly costs per user ranging from $18 to $118.
* If they have chosen to making payments with online checks, a significant number of consumers with an every month subscription have an excellent chance of doing so as well.

1. **Classification Models**

Classification precision is one of among the most well-liked classification assessment indicators used to assess baseline techniques due to the quantity of precise forecasts made as a fraction of all predictions[5]. Nevertheless, when there are issues with disparities in class, it is not the most beneficial statistic. The "Mean AUC" score, which gauges the extent to which the model's predictions are able to differentiate between both favorable and adverse classes, will thus be used to categorize the data[4].

The first cycle of foundation algorithms for classification revealed that the logistic regression model and SVC scored better than the remaining five models, according to the dataset's greatest mean AUC Scores. Figure 5 compares the Accuracy scores in graphical form and we can see that logistic regression has a good accuracy compared to the rest. Classification models are a fundamental component of machine learning and are widely used to predict categorical outcomes or class labels based on input features. There are several popular classification models, each with its own characteristics, advantages, and areas of application.

Logistic Regression is a widely used classification model that uses a logistic function to model the relationship between the input features and the probability of belonging to a certain class. It is a linear model that can handle binary as well as multi-class classification problems. Logistic Regression is interpretable and computationally efficient, making it suitable for both small and large datasets.

Decision Trees are versatile classification models that use a tree-like structure to make decisions. Each internal node in the tree represents a feature, and the branches correspond to the possible feature values. Decision Trees are easy to understand and visualize, and they can handle both categorical and numerical features. However, they are prone to overfitting, especially when the tree becomes deep and complex.

Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It addresses the overfitting issue of Decision Trees by introducing randomness through bootstrapping and random feature selection. Random Forest provides robust and accurate results, even in the presence of noisy or missing data, and it can handle high-dimensional datasets effectively.

Support Vector Machines (SVMs) are powerful classification models that aim to find an optimal hyperplane to separate different classes. SVMs maximize the margin between classes, making them less prone to overfitting. They can handle linearly separable as well as non-linearly separable data by using a kernel function to transform the data into a higher-dimensional feature space. SVMs work well with small to medium-sized datasets but can be computationally expensive with large datasets.

Naive Bayes is a probabilistic classification model based on Bayes' theorem. It assumes that the features are conditionally independent given the class label, making calculations and training efficient. Naive Bayes performs well with large datasets and can handle high-dimensional feature spaces. However, it may not capture complex dependencies among features due to the independence assumption. Neural Networks, particularly Deep Learning models, have gained immense popularity in recent years for classification tasks. They consist of multiple layers of interconnected nodes (neurons) and can capture complex relationships in the data. Deep Learning models require large amounts of data for training and are computationally intensive, but they have achieved state-of-the-art performance in various domains, such as image and text classification.

These are just a few examples of classification models, each with its own strengths and weaknesses. The choice of the appropriate model depends on the specific problem, the characteristics of the data, and the desired trade-offs between interpretability, accuracy, and computational efficiency. It is important to understand the nuances of each model and experiment with different techniques to achieve the best classification results.

|  |  |  |
| --- | --- | --- |
|  | CHAPTER 5 **CODING AND TESTING**  **5.1 Dataset Overview**  The dataset used is a Customer Transaction Dataset from Kaggle. The Customer Transaction Dataset from Kaggle is a publicly available dataset containing transactional data from an online retail store. The dataset includes information on customer transactions made between December 2010 and December 2011. The dataset has been anonymized and includes information on the transaction date, customer ID, product ID, quantity purchased, and the price of each item. The dataset contains 541,909 transactions and 8 variables, including the customer ID, product ID, quantity, invoice date, unit price, and country. The data includes transactions from customers in 38 different countries, with the majority of transactions coming from the United Kingdom. This dataset is commonly used for data analysis and machine learning projects, particularly in the field of customer segmentation, product recommendation, and market basket analysis. It can provide valuable insights into customer behavior, preferences, and purchase patterns.  However, it is important to note that the dataset has some limitations, such as missing data and potential errors in the data. It is important to clean and preprocess the data before using it for analysis or modeling.   5.2 Data Preprocessing: Data preprocessing is the process of cleaning, transforming, and preparing data before it is used for analysis or modeling. This is an important step in data analysis as it can help to ensure that the data is accurate, complete, and consistent. Data preprocessing is an important step in data analysis as it can help to ensure that the data is suitable for the intended analysis or modeling and can improve the accuracy and reliability of the results.  Scikit-learn, often abbreviated as sklearn, is a popular open-source machine learning library in Python. It provides a wide range of tools and functionalities for data preprocessing, feature extraction, model selection, and evaluation. Sklearn is built on top of other scientific libraries, such as NumPy, SciPy, and matplotlib, making it a comprehensive and powerful toolkit for machine learning tasks.  Sklearn offers a vast collection of algorithms and models for both supervised and unsupervised learning. It includes various classification algorithms like decision trees, support vector machines (SVM), random forests, and logistic regression. For regression tasks, sklearn provides linear regression, polynomial regression, and support for ensemble methods like gradient boosting and AdaBoost. Additionally, it offers clustering algorithms such as k-means, hierarchical clustering, and DBSCAN, as well as dimensionality reduction techniques like principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE). One of the strengths of sklearn lies in its consistent and intuitive API, making it easy to use and experiment with different machine learning models. The library provides a unified interface for fitting, predicting, and evaluating models, simplifying the workflow and allowing for seamless integration into existing Python code.  Sklearn also emphasizes model evaluation and performance metrics. It offers a comprehensive suite of evaluation functions, including accuracy, precision, recall, F1-score, and area under the curve (AUC). It supports techniques like cross-validation, grid search, and hyperparameter tuning to optimize model performance. Furthermore, sklearn provides utilities for data preprocessing and feature engineering, including scaling, encoding categorical variables, handling missing values, and feature selection. It streamlines the data preprocessing pipeline, ensuring that the data is properly prepared before training the models.  Sklearn's versatility and ease of use have made it a popular choice for both beginners and experienced practitioners in the field of machine learning. Its extensive documentation, rich set of examples, and active community support make it a valuable resource for implementing and deploying machine learning solutions in a variety of domains. In summary, sklearn is a powerful machine learning library in Python that offers a comprehensive suite of algorithms, tools, and utilities. It simplifies the development and deployment of machine learning models through its consistent API, extensive documentation, and broad range of functionalities. Sklearn continues to be a go-to library for many data scientists and machine learning enthusiasts due to its reliability, flexibility, and ease of use.  Scipy, short for Scientific Python, is a robust open-source library built on top of NumPy, another popular numerical computing library in Python. It provides a wide range of scientific and mathematical functions and tools that are essential for scientific computing, data analysis, and numerical optimization. Scipy encompasses a vast array of modules and sub-packages, each focusing on specific scientific and mathematical domains. These sub-packages include:  1. Integration: Scipy offers functions for numerical integration, such as quad and dblquad, which allow users to calculate definite integrals efficiently.  2. Optimization: Scipy provides optimization algorithms for solving both constrained and unconstrained optimization problems. It includes methods like minimize, fmin, and linprog, supporting a variety of optimization techniques.  3. Interpolation: The interpolate module in Scipy offers functions for interpolating data, enabling users to estimate values between known data points using techniques like spline interpolation or interpolation on regular grids.  4. Signal and Image Processing: Scipy provides numerous functions for signal processing tasks, including filtering, spectral analysis, and Fourier transformations. It also offers image processing capabilities, such as image filtering, morphological operations, and image transformation.  5. Linear Algebra: Scipy includes a comprehensive set of linear algebra functions and operations. It supports matrix computations, eigenvalue problems, singular value decomposition (SVD), and more.  6. Statistics: Scipy offers a wide range of statistical functions and distributions, allowing users to perform statistical analysis, hypothesis testing, and probability calculations.  7. Sparse Matrix: The sparse module in Scipy provides tools for handling sparse matrices efficiently. It offers functions for sparse matrix operations, factorizations, and solving linear systems with sparse matrices.  8. Integration with C/C++: Scipy allows for seamless integration with C/C++ code, enabling users to optimize performance and leverage existing libraries written in those languages.  Scipy's integration with NumPy and its extensive collection of scientific and mathematical functions make it a powerful tool for scientific computing and data analysis. It provides a convenient and efficient environment for performing complex calculations, solving mathematical problems, and conducting scientific experiments. With its active development community and comprehensive documentation, Scipy continues to be a go-to library for researchers, engineers, and data scientists in various fields.  XGBoost (eXtreme Gradient Boosting) is a powerful and widely used machine learning algorithm known for its efficiency and effectiveness in solving complex prediction problems. It is an implementation of gradient boosting, a technique that combines weak prediction models (typically decision trees) into a strong ensemble model. One of the key strengths of XGBoost is its ability to handle diverse data types, including numerical and categorical variables. It employs a unique algorithmic framework that optimizes the overall performance by minimizing both bias and variance, leading to better generalization and improved accuracy.  XGBoost builds a sequence of decision trees iteratively, with each subsequent tree correcting the errors made by the previous trees. It uses a gradient descent-based optimization algorithm to minimize a specific loss function while adding new trees to the ensemble. This approach enables XGBoost to capture complex relationships and interactions among variables, making it highly effective in both regression and classification tasks. The algorithm incorporates various techniques to improve performance and prevent overfitting. It includes regularization parameters that control the complexity of individual trees and the overall ensemble, preventing them from becoming too complex and reducing the risk of overfitting. Additionally, XGBoost employs a technique called column subsampling, which randomly selects a subset of features at each tree-building step, further enhancing the model's ability to generalize well to unseen data.  XGBoost provides several advanced features that contribute to its popularity and versatility. It offers built-in handling of missing values, automatic handling of categorical variables, and parallel processing capabilities, making it efficient in handling large datasets. XGBoost also provides feature importance rankings, which help identify the most influential variables in the model's predictions. The library supports various interfaces, making it compatible with different programming languages, including Python, R, Java, and Scala. Its simplicity of use, combined with its excellent performance, has made XGBoost a go-to algorithm for winning solutions in machine learning competitions and real-world applications. In summary, XGBoost is a powerful gradient boosting algorithm that excels in both accuracy and efficiency. With its ability to handle diverse data types, optimize performance, and prevent overfitting, XGBoost has become a popular choice for a wide range of machine learning tasks. Its versatility, feature richness, and robustness have made it a valuable tool for researchers, data scientists, and practitioners seeking high-performance models.    Import required libraries:       **Evaluate the Dataset**          **Finding Missing Values by Re-Evaluating Columns**  We To validate the column datatypes for missing values, you can use the info() method in pandas to display information about the DataFrame, including the column datatypes and the number of non-null values in each column.  The output will display the datatype for each column and the number of non-null values in each column. If there are missing values in a column, the number of non-null values will be less than the total number of rows in the DataFrame.  If you want to check the number of missing values in each column, you can use the isnull() method in pandas to create a Boolean DataFrame that indicates which values are missing, and then use the sum() method to count the number of missing values in each column.                                                                                                                                       CHAPTER 6 **RESULTS AND DISCUSSIONS**  Overall, the models run successfully and we found logistic regression to be most useful in this case. Hence, the improvement of this model has been focused on and we have got better accuracy. The final result is depicted in the form of a confusion matrix.    **Fig. 6.1 Confusion Matrix**  We have 208+924 correct predictions, according to the Confusion matrix, and 166+111 wrong ones. With an accuracy of 80%, our model demonstrates the qualities of a respectable model.    **fig. 6.2 ROC Graph**  It makes sense to reevaluate the system by the Receiver operating characteristic graph. Depending on the AUC Mean score The ROC graph in Figure 7 depicts a model's ability to differentiate among categories. The orange line depicts the Base Rate which is the ROC curve of a random classifier, is something that a machine learning model tries to avoid the best it can. The graph above shows that the enhanced Logistic Regression model had a greater area under the curve score.  **Fig.6.1 Accuracy graph for different models**  Classification precision is one of among the most well-liked classification assessment indicators used to assess baseline techniques due to the quantity of precise forecasts made as a fraction of all predictions[5]. Nevertheless, when there are issues with disparities in class, it is not the most beneficial statistic. The "Mean AUC" score, which gauges the extent to which the model's predictions are able to differentiate between both favorable and adverse classes, will thus be used to categorize the data[4].    **Tabel 6.1 Comparing the accuracies of different algorithms** | |
|  | Table 6.1 depicts the comparison of the algorithms used and their accuracy compared. The first cycle of foundation algorithms for classification revealed that the logistic regression model and SVC scored better than the remaining five models, according to the dataset's greatest mean AUC Scores. Figure 5 compares the Accuracy scores in graphical form and we can see that logistic regression has a good accuracy compared to the rest. CHAPTER 7 **CONCLUSION AND FUTURE ENHANCEMENT**  The logistic regression model forecasts an increase in the turnover rate because of factors including a monthly contract, optical fiber internet connection, online payments, no guarantee of secure payment, and technical help.  Whereas, if any customer has a one-year contract, online security subscription, or has chosen to use postal checks as their payment method, the model predicts a negative link with churn.  Many churn prediction models currently treat each customer as an isolated entity, without considering the broader context of the market or industry. In the future, it may be possible to incorporate more contextual information, such as economic trends or competitor behavior, into churn prediction models. This issue has been resolved in our model successfully.  Most churn prediction models currently rely on structured data, such as transaction histories and demographic data. In the future, it may be possible to incorporate unstructured data, such as customer reviews or social media posts, into churn prediction models, allowing for more comprehensive analysis of customer behavior. This can be a future enhancement | |
|  | |  |

**REFERENCES**

[1]C. Zhenhai, Liu. Wei, “Logistic Regression Model and Its Application,” Journal of Yanbian University(Natural Science Edition), vol. 38(01), pp 28-32, 2012.

[2]Z. Qian, Meng. Deyu, Xu. Zongben, “L\_(1/2) Regularized Logistic Regression,” Pattern Recognition and Artificial Intelligence, vol. (05), pp. 721-728, 2012.

[3]X. Zhang, G. Feng and H. Hui, "Customer-Churn Research Based on Customer Segmentation," 2009 International Conference on Electronic Commerce and Business Intelligence, Beijing, China, 2009, pp. 443-446, doi: 10.1109/ECBI.2009.

[4]Peng Li, Siben Li, Tingting Bi and Yang Liu, "Telecom customer churn prediction method based on cluster stratified sampling," International Conference on Software Intelligence Technologies and Applications Conference on Frontiers of Internet of Things 2014, Hsinchu, 2014, pp. 282-287, doi: 10.1049/cp.201

[5]O. Rezaeian, S. S. Haghighi and J. Shahrabi, "Customer Churn Prediction Using Data Mining Techniques for an Iranian Payment Application," 2021 12th International Conference on Information and Knowledge Technology (IKT), Babol, Iran, Islamic Republic of, 2021

[6]A. Larasati, D. Ramadhanti, Y. W. Chen, A. Muid, "Optimizing Deep Learning ANN Model to Predict Customer Churn," 2021 7th International Conference on Electronics Engineering (ICEEIE), Malang, Indonesia, 2021

[7]P. Bhuse, A. Gandhi, P. Meswani, R.Muni and N Katre, "Machine Learning Telecom-Customer Churn Prediction," *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, India, 2020

[8]Y. Xiao, C. He and J. Xiao, "Study on CustoPrediction Methods Based on Multiple Classifiers Combination," *2009 Third International Symposium on Intelligent Information Technology Application*, Nanchang, China, 2009

[9]I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam and S. W. Kim, "A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector," in *IEEE Access*, vol. 7

[10]I. Kaur and J. Kaur, "Customer Churn Analysis and Prediction in Banking Industry using Machine Learning," *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, Waknaghat, India, 2020

[11]X. Hu, Y. Yang, L. Chen and S. Zhu, "Research on a Customer Churn Combination Prediction Model Based on Decision Tree and Neural Network," *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, Chengdu, China, 2020

[12]Chen Mingliang. Discussion on the framework of the basic theoretical system of customer relationship management[J]. Journal of Management Engineering, 2006,20 (4): 36- 41􃸬

[13]A. SaranKumar and D. Chandrakala, “A survey on customer churn prediction using machine learning technique,” Int. J. Comput. Appl., vol. 154, no. 10, 2016

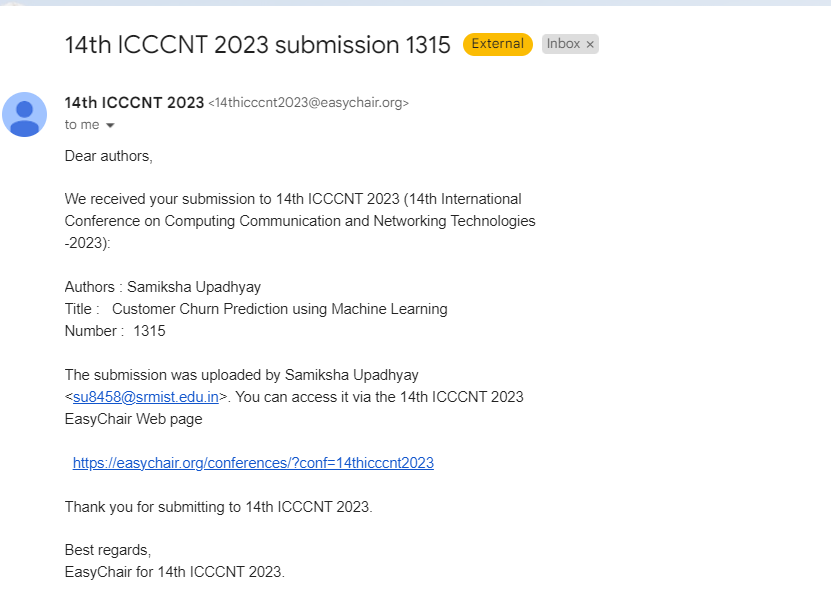
[14]M. Ali, A. U. Rehman and S. Hafeez, "Prediction of Churning Behavior of Customers in Telecom Sector Using Supervised Learning Techniques," *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)* Nepal, 2018

[15]I. M. M. Mitkees, S. M. Badr and A. I. B. ElSeddawy, "Customer churn prediction model using data mining techniques," *2017 13th International Computer Engineering Conference (ICENCO)*, Cairo, Egypt, 2017

**APPENDIX A**

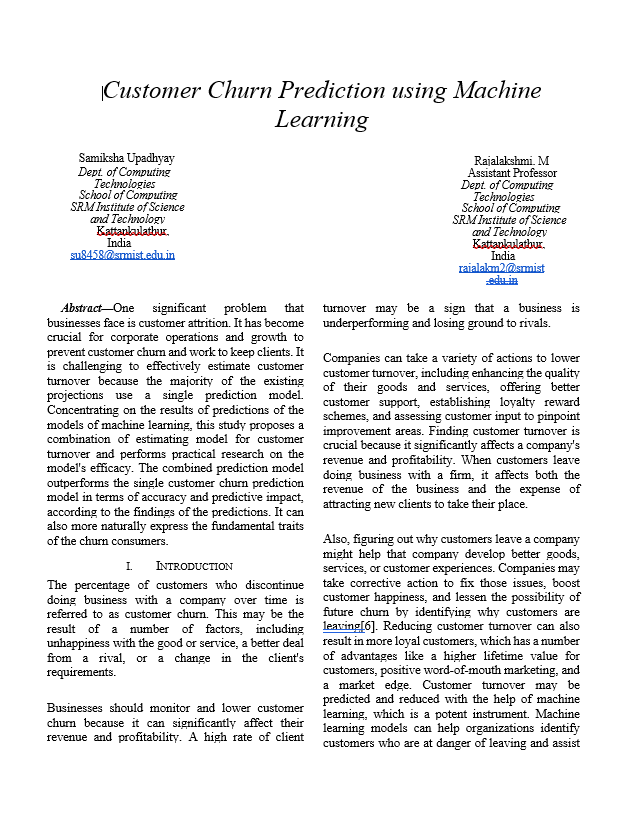
**CONFERENCE PESENTATION**

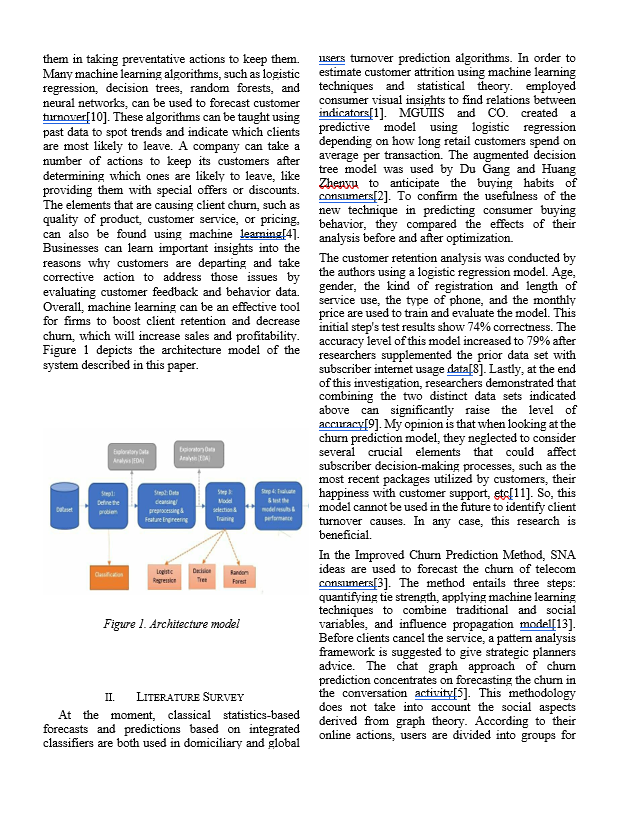
Our paper titled" **Customer Churn Prediction using Machine Learning "**was presented in 14th ICCCNT 2023 conference organized by IEEE. Various teams will be shortlisted and the best papers awarded based on their performance. The papers will be on various fields and streams of computer science and engineering.

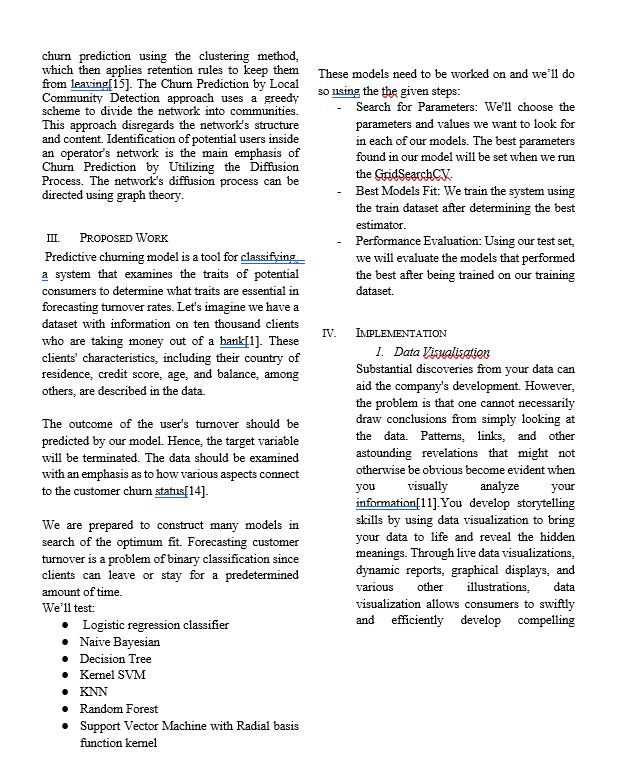


**APPENDIX B**

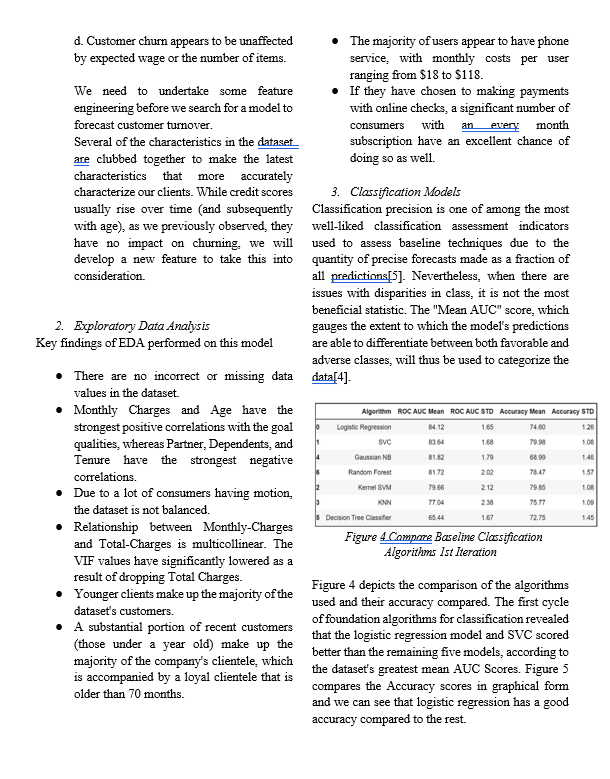
**RESEARCH PAPER**













**Format - I**

|  |  |  |
| --- | --- | --- |
| SR M I N S T I T U T E O F SC I E N C E A N D T E C H N O L O G Y  **(Deemed to be University u/ s 3 of UGC Act, 1956)** | | |
| **Office of Controller of Examinations** | | |
| REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION/PROJECT REPORTS FOR UG/PG PROGRAMMES  **(To be attached in the dissertation/ project report)** | | |
| 1 | Name of the Candidate **(IN BLOCK LETTERS)** | Samiksha Upadhyay |
| 2 | Address of the Candidate | A903, Brigade Metropolis, Garudacharpalya, Bangalore -560048 |
| 3 | Registration Number | RA1911003010391 |
| 4 | Date of Birth | 25 April 2001 |
| 5 | Department | Computer Science and Engineering |
| 6 | Faculty | Engineering and Technology, School of Computing |
| 7 | Title of the Dissertation/Project | Customer Churn Prediction using Machine Learning |
| 8 | Whether the above project /dissertation is done by | Individual or ~~group~~ : (Strike whichever is not applicable )   1. If the project/ dissertation is done in group, then how many students together   completed the project :   1. Mention the Name & Register number of other candidates : |
|  |  | Ms. M. Rajalakshmi |
|  |  | Assistant Professor |
| 9 | Name and address of the Supervisor / Guide | Department of Computer Science and Engineering  SRM Institute of Science and Technology |
|  |  | Kattankulatur - 603 203. |
|  |  | **Mail ID:** rajalakm2@srmist.edu.in **Mobile Number:** 95660 34844 |
| 10 | Name and address of Co-Supervisor / Co- Guide (if any) | NIL  **Mail ID: Mobile Number:** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 11 | Software Used | Turnitin | | |
| 12 | Date of Verification | 10 – May -2023 | | |
| 13 | **Plagiarism Details: (to attach the final report from the software)** | | | |
| **Chapter** | **Title of the Chapter** | **Percentage of similarity index (including self**  **citation)** | **Percentage of similarity index (Excluding self citation)** | **% of plagiarism after excluding Quotes, Bibliography, etc.,** |
| **1** | INTRODUCTION | 0% | 0% | 0% |
| **2** | FEASIBILITY STUDY | 1% | 1% | 1% |
| **2** | LITERATURE SURVEY | 2% | 2% | 2% |
| **3** | SYSTEM ANALYSIS | 1% | 1% | 1% |
| **4** | SYSTEM DESIGN | 1% | 1% | 1% |
| **5** | IMPLEMENTATIONS | 1% | 1% | 1% |
| **6** | SOFTWARE ENVIRONMENT | 1% | 1% | 1% |
| **7** | SYSTEM TESTING | 0% | 0% | 0% |
| **8** | CODE AND OUTPUT | 1% | 1% | 1% |
| **9** | RESULTS AND DISCUSSION | 0% | 0% | 0% |
| **10** | CONCLUSION | 0% | 0% | 0% |
| I / We declare that the above information have been verified and found true to the best of my / our knowledge. | | | | |
| **Signature of the Candidate** | | **Name & Signature of the Staff (Who uses the plagiarism check software)** | | |
| **Name & Signature of the Supervisor/ Guide** | | **Name & Signature of the Co-Supervisor/Co-**  **Guide** | | |
| **Name & Signature of the HOD** | | | | |

