Optimizing Telecom Operations with Segmentation and Churn Prediction

DANDU NEHA 21951A05B8

MOHAMMED AMEESHA 21951A05A4

MUKKA DHEERAJ 21951A0538

Optimizing Telecom Operations with Segmentation and Churn Prediction

A Project Report

Submitted in partial fulfillment of the

requirements for the award of the degree of

Bachelor of Technology

in

Computer Science & Engineering

by

Dandu Neha	**21951A05B8**
Mohammed Ameesha	**21951A05A4**
Mukka Dheeraj	**21951A0538**

Department of Computer Science Engineering

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043, Telangana

APRIL 2024

DECLARATION

I certify that.

a. The work contained in this report is original and has been done by me under the guidance of my supervisor(s).

b. the work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in preparing the report.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Place: **Signature of the Student**

Date: **Dandu Neha**

Mohammed Ameesha

Mukka Dheeraj

CERTIFICATE

This is to certify that the project report entitled "**Optimizing Telecom Operations with Segmentation and Churn Prediction**" submitted by Ms. Dandu Neha, Ms. Mohammed Ameesha, Mr. Mukka Dheeraj to the Institute of Aeronautical Engineering, Hyderabad, in partial in partial fulfillment of the requirements for the reward of the Degree Bachelor of Technology in Computer Science And Engineering is a bonafide record of work carried out by him/her under my/our guidance and supervision. In whole or in parts, the contents of this report have not been submitted to any other institutes for the award of any Degree**.**

Supervisor:

Ms. K. Sangeeta

Assistant Professor

Head of the Department:

Dr. C Madhusudhana Rao

Professor and HOD, CSE

Date:

APPROVAL SHEET

This project report entitled **Optimizing Telecom Operations with Segmentation and Churn Prediction** by **Ms. Dandu Neha, Ms. Mohammed Ameesha, Mr.Mukka Dheeraj** is approved for the award of the Degree Bachelor of Technology in **Computer Science and Engineering**.

Examiners

Supervisor(s)
Ms. K. Sangeeta

Principal

Dr. L. V. Narasimha Prasad

Date:

Place:

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success. I think out college management and respected Sri M. Rajashekar Reddy, Chairman, IARE, Dundigal for providing me with the necessary infrastructure to conduct the project work.

I express my sincere thanks to Dr. L. V. Narasimha Prasad, Professor and Principal who has been a great source of information for my work, and Dr. C. Madhusudhan Rao, Professor and Head, Department of CSE, for extending his support to carry on this project work.

I am especially thankful to our supervisor Ms. K. Sangeeta, Assistant Professor, Department of CSE, for her internal support and professionalism who helped me in shaping the project into a successful one. I take this opportunity to express my thanks to one and all who directly or indirectly helped me in bringing the effort to present form.

ABSTRACT

Keywords: Customer churn, churn prediction, telecom operations, machine learning, feature selection, customer segmentation, operational efficiency.

Client turnover is a significant issue and one of the top concerns for large businesses, particularly in the telecom sector, due to its direct impact on company profits. Companies are striving to develop methods to forecast probable outcomes to mitigate this impact. Identifying the factors contributing to customer churn is crucial for preventing it. Our work's key contribution is the creation of a churn prediction model that aids telecom providers in identifying customers who are most likely to churn. We analyze various scenarios of data analysis and present the results in graphical format. The model developed employs machine learning methods in a large data environment and introduces a novel approach for designing and selecting features. The datasets used to train, evaluate, and assess the algorithm include all customer data from previous months. We tested four different algorithms: Logistic Regression, Extreme Gradient Boosting (XGBoost), Gradient Boosted Machine (GBM) Trees, Random Forests, and Decision Trees. To increase the model's accuracy, the algorithms are trained with parameter tuning. This work focuses on optimizing telecom operations with segmentation and churn prediction, ultimately enhancing operational efficiency and customer retention.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ML	Machine Learning
AI	Artificial Intelligence
RF	Random Forest
LR	Logistic Regression
XGBoost	Extreme Gradient Boosting
GBM	Gradient Boosted Machine Trees
CRM	Customer Relationship Management
SQL	Structural Query language

CHAPTER 1
INTRODUCTION

In the fiercely competitive telecommunication industry, customer churn—when customers discontinue their service subscriptions—poses a significant challenge. Acquiring new customers often costs more than retaining existing ones, making churn reduction crucial for telecom operators. High churn rates can erode profitability and market share, highlighting the need for effective churn prediction and management strategies. Machine learning (ML) provides a transformative solution to this challenge. By leveraging vast amounts of historical customer data, such as service usage patterns, demographic details, and interaction histories, ML models can predict the likelihood of a customer discontinuing their service. This predictive capability enables telecom companies to proactively engage with at-risk customers, offering personalized incentives or improving service quality to retain them.

Developing a churn prediction model involves several key steps: data collection and preprocessing, feature selection and engineering, model training, and evaluation. Data from various sources such as call records, billing information, customer service interactions, and social media can be integrated to form a comprehensive view of each customer's behavior. Advanced ML algorithms, including logistic regression, decision trees, random forests, and neural networks, can analyze these data points and identify patterns indicative of churn. Effective churn prediction not only helps in retaining customers but also provides insights into the underlying reasons for churn. These insights can drive strategic decisions in product development, customer service enhancements, and targeted marketing campaigns.

Moreover, reducing churn rates translates directly into increased customer lifetime value and a more stable revenue base. This project aims to build a scalable and accurate churn prediction system that can be seamlessly integrated into a telecom operator's existing infrastructure. By doing so, telecom companies can shift from reactive to proactive customer relationship management, enhancing customer satisfaction and loyalty in a highly competitive market.

1.1 PROBLEM STATEMENT

The telecommunication industry faces significant challenges due to customer churn, where subscribers discontinue their services, leading to revenue loss and increased customer acquisition costs. This project aims to develop a machine learning model to predict customer churn by analyzing user behavior, service usage patterns, and demographic data. Accurate churn prediction enables telecom companies to proactively engage at-risk customers with retention strategies, such as personalized offers or service improvements. The goal is to enhance customer satisfaction, reduce churn rates, and ultimately increase profitability. The solution involves using various algorithms to process historical data and identify key indicators of churn.

1.2 MOTIVATION

Customer churn presents a significant challenge in the telecommunications sector, mainly because it incurs substantial financial and operational costs. Retaining a current customer is generally far more economical than acquiring a new one. The process of attracting new customers involves substantial expenses, from advertising and promotions to competitive pricing strategies, making it a costly alternative to retention. Due to these high costs, companies place great importance on customer retention, seeking ways to sustain and nurture existing customer relationships.

Loyal customers are invaluable because they contribute to consistent revenue streams and often serve as informal brand ambassadors. Over time, satisfied customers may increase their service usage, upgrade plans, or add additional services, all of which contribute to stable revenue growth. Additionally, they often refer others to the company, creating organic growth without the need for intensive marketing campaigns. These combined benefits make loyal customers a crucial asset to any telecom provider.

Churn prediction has therefore become an essential component of customer relationship management (CRM) strategies in telecom. By anticipating potential churn, companies can proactively engage with at-risk customers through tailored interventions, resolving issues or providing incentives that encourage them to stay. Churn prediction models leverage machine learning to analyze customer data, identifying patterns and behaviors that indicate a likelihood of leaving. Armed with these insights, companies can implement specific retention measures, such as personalized offers or enhanced customer support, improving retention rates and overall customer satisfaction.

The motivation behind this project lies in addressing churn by implementing machine learning models for precise customer segmentation and accurate churn prediction. Segmentation allows telecom providers to classify customers into distinct groups based on behavioral and demographic similarities. This categorization provides an opportunity to understand each segment's unique preferences and needs, enabling more targeted retention efforts. By integrating segmentation with churn prediction, this project aims to equip telecom companies with actionable insights that allow them to address customer needs effectively and retain their competitive standing.

1.3 OBJECTIVES

The central goal of this project is to create a sophisticated machine learning model that can reliably identify customers at a high risk of leaving. By examining a variety of historical customer data—such as interaction records, usage trends, and demographic profiles—the model aims to detect critical signs that suggest a customer might churn. This capability will empower telecom companies to take preemptive action, engaging with vulnerable customers early enough to implement strategies that can effectively deter them from discontinuing their service ([1][2][3]).

In addition to predicting churn, this project aims to enhance retention initiatives through personalized interventions that cater to the unique preferences and behaviors of different customer segments. By employing advanced segmentation techniques, the model will facilitate the design of targeted retention strategies that resonate with specific groups, ultimately improving customer engagement and satisfaction. This approach not only ensures a more personalized experience for customers but also allows telecom operators to allocate their marketing and support resources more strategically, directing efforts toward those customers who stand to benefit most from tailored engagement ([4][5][6]).

Moreover, the project intends to boost the overall efficiency of customer retention operations. With precise churn predictions, telecom companies can better manage their resources by focusing on high-risk cases rather than resorting to broad, one-size-fits-all retention campaigns. This focused strategy is expected to reduce unnecessary expenditures while enhancing the effectiveness of customer service efforts and increasing customer loyalty ([7][8][9]).

The overarching aim is to drive revenue growth by minimizing customer churn and cultivating long-term relationships with satisfied customers. By incorporating predictive analytics into their retention strategies, telecom providers can not only decrease attrition rates but also enhance their capacity to generate consistent revenue through loyal customer bases. This, in turn, reinforces their competitive advantage within the industry ([10][11][12]).

In conclusion, this project seeks to develop a powerful machine learning model that enables telecom companies to predict customer churn accurately, devise effective retention strategies through tailored interventions, enhance operational efficiency, and ultimately maximize revenue generation. By taking a proactive, data-driven approach to churn management, the project aspires to strengthen customer relationships and significantly improve the overall performance of telecom operations ([13][14]).

1.4 REQUIREMENTS SPECIFICATIONS

1.4.1 SOFTWARE REQUIREMENTS

Software requirements delineate the essential software resources and dependencies that must be in place for an application to operate optimally. These requirements often include specific libraries, frameworks, and tools that need to be installed separately prior to the application installation. Such dependencies are not typically bundled within the installation package and must be managed independently to ensure a smooth installation and functioning of the software.

Platform Specifications: A platform in computing refers to a foundational framework—either in hardware or software—that supports the execution and development of software applications. This encompasses various elements, including:

Computer Architecture: The design and organization of the components of a computer system, which can affect how software interacts with hardware.

Operating Systems: The system software that manages computer hardware and provides services for computer programs. Key examples include Microsoft Windows, Linux distributions, and macOS.

Programming Languages: The languages in which the software is developed, alongside their respective runtime libraries that provide essential functionalities during execution.

When defining software requirements, compatibility with the operating system is of paramount importance. Each application must align with the intended operating system, considering factors such as the version being used. While many modern operating systems strive for backward compatibility, it is not always guaranteed. For example, software designed for Windows XP may not function on Windows 98, although applications running on newer versions might exhibit some backward compatibility with older versions. Similarly, software that leverages features exclusive to newer Linux kernel versions (like v2.6) typically fails to operate correctly on older kernel versions (like v2.2 or v2.4).

APIs and Drivers

Applications that heavily interact with specialized hardware devices, such as high-performance graphics cards or sound devices, often require specific Application Programming Interfaces (APIs) or updated device drivers. APIs provide the necessary protocols and tools for building software applications, enabling communication between software and hardware.

A notable example is DirectX, which encompasses a collection of APIs designed for multimedia tasks, particularly within gaming on Microsoft platforms. DirectX facilitates the handling of tasks related to graphics, sound, and input, allowing developers to create rich multimedia experiences without needing to write device-specific code.

Web Browser Considerations

In the context of web applications and software that relies on internet technologies, the choice of web browser installed on the system can significantly influence functionality. Many applications default to utilizing the system's pre-installed web browser, which can affect compatibility and security.

For instance, Microsoft Internet Explorer has historically been the go-to browser for software operating on Windows systems, despite its vulnerabilities associated with features like ActiveX controls, which have been known to introduce security risks. With the evolution of web standards and security practices, many applications now strive for compatibility with multiple browsers, including more modern options like Google Chrome, Mozilla Firefox, and Microsoft Edge, to enhance user experience and security.

Development Tools and Environment

Coding Language: Python is the primary programming language, known for its readability, versatility, and robust community support. It is widely used in data science, web development, and machine learning applications.

Tool: Anaconda serves as the development environment, providing a comprehensive package management system and environment management capabilities tailored for scientific computing and data analysis. Anaconda simplifies package installation and dependency management, making it an ideal choice for projects that require multiple libraries and tools.

Interface: The project will utilize Flask and Jupyter Notebook as interfaces. Flask is a micro web framework for Python that allows developers to build web applications quickly and efficiently. Jupyter Notebook, on the other hand, is an open-source web application that enables the creation and sharing of documents containing live code, equations, visualizations, and narrative text. This combination allows for a seamless development experience, particularly for data-driven applications.

1.4.2 HARDWARE REQUIREMENTS

In the context of software applications and operating systems, hardware requirements refer to the essential physical resources necessary for optimal performance and functionality. Alongside these hardware specifications, it is common to provide a Hardware Compatibility List (HCL), particularly for operating systems. This list enumerates devices that have been tested for compatibility, as well as those that may not function properly with a specific operating system or application. Below, we explore the various aspects of hardware requirements in greater detail.

Architecture

Computer operating systems are specifically designed to function on certain computer architectures. While there are applications that can operate independently of the underlying platform, the majority of software is closely tied to particular operating systems and their corresponding architectures. This means that even though some systems can be architecture-independent, they often require recompilation or modification to run on a different architecture. Understanding the architecture is essential for ensuring that the software can leverage the capabilities of the hardware it is intended to run on.

Processing Power

The processing power of the Central Processing Unit (CPU) is a critical requirement for any software application. For programs running on x86 architecture, processing power is typically characterized by the CPU's model and clock speed. A more powerful CPU can significantly enhance performance, allowing for quicker data processing and improved responsiveness of applications. In this context, the

Intel® Core™ i3-7020U CPU, operating at a clock speed of 2.30 GHz, represents a capable choice for handling basic software tasks and applications, making it suitable for various computing needs.

Memory

Random Access Memory (RAM) is another crucial component that impacts the execution of software. Every application consumes a portion of the system's RAM when it is running, which is essential for maintaining smooth performance. Memory requirements are determined by the software's demands, the operating system, any additional supporting applications, and the files being processed simultaneously. A system equipped with 4 GB of RAM can effectively manage lightweight applications and basic multitasking; however, resource-intensive software may require additional memory for optimal performance.

Secondary Storage

The requirements for secondary storage, typically involving hard disk space, can vary widely based on multiple factors. These include the total size of the software installation, the temporary files generated during the installation or operation of applications, and any swap space usage that may be necessary if the available RAM is insufficient. A hard disk with a capacity of 1 TB provides ample space for software installations, data storage, and temporary files, making it a robust choice for most users. This capacity ensures that the system can handle a significant amount of data without running into storage limitations.

CHAPTER 2

LITERATURE SURVEY

In the competitive landscape of the telecommunications industry, customer churn—defined as the rate at which customers stop doing business with a company—poses significant challenges and risks. Customer churn prediction is a significant area of focus in the telecom industry due to the direct impact of customer retention on profitability and competitive advantage. Understanding the dynamics of customer behavior, satisfaction, and loyalty is vital for telecom operators aiming to reduce churn rates. A considerable body of research has emerged, employing various methodologies and techniques to analyze the factors influencing customer attrition.

Early studies laid the groundwork for understanding the relationship between customer satisfaction and retention, revealing that high customer satisfaction is linked to increased loyalty and reduced churn. These findings emphasized that telecom companies must prioritize customer satisfaction to mitigate churn effectively. This foundational work has been widely cited as a basis for further research into churn prediction methodologies [1][2]. Innovative approaches to churn prediction have been introduced by applying data mining techniques to telecommunications call detail records. Significant insights can be gleaned from call data, effectively transforming raw data into actionable intelligence for predicting churn. This shift towards data-driven strategies in churn management highlights the potential of leveraging customer behavior data [3][4].

Recent developments have seen the emergence of machine learning-based churn prediction models that utilize various algorithms to identify customers at risk of leaving. This research showcases the effectiveness of machine learning in enhancing the accuracy of churn predictions, enabling telecom operators to implement targeted retention strategies [5][6][7]. Such advancements underline the growing reliance on sophisticated analytical methods to address customer retention challenges.

The need for proactive measures in preventing churn has also been examined, with studies investigating the effectiveness of tailored customer interactions. Experimental findings indicate that proactive engagement with customers can significantly lower churn rates, emphasizing the importance of personalized marketing in the telecom, where customized offerings enhance customer loyalty [8][9].

Furthermore, the application of multilayer perceptron neural networks for predicting customer churn has demonstrated the potential of neural network architectures in accurately modeling churn behavior. This research offers insights into the underlying patterns that drive customer decisions and has contributed to the growing interest in artificial intelligence techniques for churn prediction [10][11].

Comprehensive surveys have reviewed various datasets, methodologies, and performance metrics used in churn prediction, serving as valuable resources for researchers and practitioners. These reviews provide an overview of the state-of-the-art techniques available for analyzing churn and highlight the importance of continually evolving methodologies to keep pace with changing customer behaviors and market dynamics [12][13].

Addressing class imbalance in churn prediction has been a crucial topic, as traditional predictive models often struggle with imbalanced datasets. Effective strategies for handling this issue are particularly relevant in the telecom sector, where the number of churned customers is often lower than retained ones. Utilizing specialized techniques for imbalanced data has been shown to enhance model performance and reliability [14][15].

The influence of big data analytics on churn prediction has underscored how integrating vast amounts of customer data can significantly improve predictive capabilities. Advanced analytical techniques enable telecom companies to derive insights from complex datasets, ultimately leading to better decision-making processes and enhanced retention strategies. This perspective aligns with the growing importance of big data in various industries, including telecommunications [16][17].

Several studies have investigated the comparative effectiveness of different predictive models for churn prediction, offering insights into which models yield the best performance in churn scenarios. The findings emphasize the necessity for telecom companies to evaluate multiple modeling approaches to determine the most effective strategy for their specific customer base [18][19].

Moreover, the role of big data analytics in churn prediction highlights its transformative impact on model accuracy and predictive capabilities. By effectively analyzing customer behavior, companies can enhance their retention strategies and improve overall customer satisfaction [20][21].

Customer feedback plays a critical role in churn prediction, as understanding customer sentiment can inform more effective retention strategies. Integrating customer insights into churn prediction models enhances the relevance and applicability of the findings, ensuring that retention strategies align with customer expectations [22][23].

Ethical considerations surrounding data usage have become increasingly relevant, particularly in the context of customer churn prediction. The implications of data privacy and the ethical challenges that arise from leveraging customer data for predictive analytics must be addressed. Striking a balance between the benefits of data-driven strategies and the need for ethical data management practices is essential for maintaining customer trust [24][25].

In conclusion, the literature on customer churn prediction in the telecom industry reveals a rich tapestry of research that spans various methodologies, technologies, and insights. The integration of machine learning techniques, the focus on handling class imbalance, and the importance of customer feedback are central themes that emerge from this body of work[26][27]. The ongoing advancements in big data analytics continue to shape the landscape of churn prediction, providing telecom operators with the tools necessary to mitigate customer attrition effectively. As the industry evolves, further research will be essential to address emerging challenges and refine predictive models to adapt to the ever-changing dynamics of customer behavior [28][29][30].

In summary, managing customer churn in the telecommunications sector necessitates a comprehensive approach that integrates advanced analytics, customer segmentation, and proactive retention strategies. By leveraging machine learning and big data technologies, telecom companies can build robust churn prediction models that yield valuable insights into customer behavior. Nonetheless, challenges related to data integration, privacy regulations, and the ever-changing technological landscape must be addressed to unlock the full potential of churn prediction initiatives. As the industry evolves, ongoing investment in technology, collaboration, and ethical data practices will be paramount for telecom operators aiming to enhance customer loyalty and drive sustainable revenue growth.

CHAPTER 3
EXISTING MECHANISMS

Rule-Based Systems:

Rule-based systems serve as a foundational approach in predicting customer churn by utilizing a set of predefined rules and thresholds developed through industry expertise to identify at-risk customers. For instance, a rule may categorize a customer as likely to churn if they have not engaged with the service for a certain number of days[1]. This methodology offers a clear and interpretable mechanism for flagging customers who may need intervention, allowing companies to concentrate their retention efforts more effectively[2].

However, rule-based systems exhibit several significant drawbacks. A major limitation is their static nature; once established, the rules do not readily adapt to shifts in customer behavior or changes in market dynamics[3]. As customer preferences evolve or competitive landscapes shift, the original rules may quickly become outdated[4]. Furthermore, these systems often struggle to account for the complexity inherent in customer behavior, as they typically overlook intricate, non-linear relationships within the data that play a crucial role in influencing churn [5]. Additionally, maintaining the relevance of these rules requires ongoing input from domain experts, resulting in a resource-intensive process that demands regular updates and revisions to ensure effectiveness over time[6].

Statistical Methods:

In contrast, statistical methods, including logistic regression and survival analysis, are commonly employed in churn management to model the probability of customer attrition based on historical data[7]. These techniques leverage statistical principles to establish connections between customer characteristics and the likelihood of churn, providing a quantitative basis for predicting behavior[8].

Despite their strengths, statistical methods also encounter significant challenges. Many models operate under the assumption of linear relationships, which can lead to oversimplified interpretations of the data[9]. This reliance on simplifications often fails to capture the more complex interactions that characterize customer behavior[10]. Additionally, scalability presents a notable issue when applying these methods to large, high-dimensional datasets typical of the telecommunications industry[11]. As the volume and intricacy of data increase, traditional statistical approaches may struggle to deliver accurate predictions[12]. Furthermore, these models can be less effective in recognizing intricate patterns in customer interactions, limiting their ability to provide reliable insights in a dynamic environment where customer experiences are multifaceted[14].

In summary, while both rule-based systems and statistical methods offer valuable frameworks for understanding and predicting customer churn, their limitations underscore the necessity for more advanced analytical techniques[13]. By addressing the inflexibility associated with rule-based approaches and the simplifying assumptions of statistical models, telecom companies can enhance their predictive capabilities and improve customer retention strategies[15]. Embracing more sophisticated methods, such as machine learning algorithms, can enable organizations to better capture the complexities of customer behavior and generate actionable insights that drive effective retention efforts[16].

3.1 LIMITATIONS OF EXISTING MECHANISMS

Limited Predictive Accuracy:

Current systems for managing customer churn in the telecommunications industry face several notable limitations that impede their overall effectiveness. A primary concern is the limited predictive accuracy of these traditional methods. Many of them struggle to effectively forecast churn due to their challenges in processing large datasets and capturing the intricate relationships among various customer characteristics. This often results in either overlooking potential churners or mistakenly classifying loyal customers as being at risk.

Reactive Rather Than Proactive:

Furthermore, a significant number of existing systems operate in a reactive manner. They typically respond to churn signals only after customers exhibit clear signs of dissatisfaction or disengagement. This reactive stance severely restricts the opportunities for companies to intervene proactively and address issues before customers decide to leave.

High False Positive and False Negative Rates:

Another critical drawback lies in the tendency of rule-based and basic statistical models to produce high rates of false positives and false negatives. False positives occur when customers who are not at risk of churning are incorrectly identified as potential churners. Conversely, false negatives happen when actual churn risks are overlooked. These inaccuracies can lead to poor allocation of resources in retention efforts, causing companies to expend time and resources targeting customers who do not need intervention while neglecting those who truly require attention.

Inability to Personalize Interventions:

Moreover, many existing systems lack the ability to provide personalized interventions. Without detailed insights into individual customer behaviors and preferences, traditional approaches cannot effectively customize retention strategies to suit the unique needs of each customer. This often leads to generic interventions that fail to resonate with customers and are frequently ineffective.

Time and Resource Intensive:

The manual processes involved in setting up and maintaining rule-based systems, combined with the computational demands of traditional statistical methods, contribute to inefficiencies. These processes are often resource-intensive and time-consuming, ultimately reducing operational efficiency.

Poor Adaptability:

Lastly, adaptability is a crucial issue for existing systems. Many of these systems struggle to adjust to rapid shifts in customer behavior, market dynamics, or the introduction of new products. This lack of flexibility can diminish their effectiveness over time, as the industry landscape continues to evolve.

In conclusion, these limitations underscore the pressing need for more advanced analytical methodologies that can improve predictive accuracy, enable early interventions, and facilitate personalized retention strategies within the telecommunications sector.

CHAPTER 4
CHURN PREDICTION MODEL

To tackle the pressing issue of client turnover in the telecom industry, our proposed system aims to develop an advanced churn prediction model. This model is designed to help telecom providers proactively identify customers who are most likely to churn, thereby enabling effective retention strategies.

Our approach utilizes sophisticated machine learning techniques within a large data environment to ensure high prediction accuracy and reliability.The system includes a comprehensive analysis of customer data, incorporating a novel method for feature design and selection. This ensures that the most relevant factors contributing to churn are effectively identified and used. By examining various data scenarios and presenting the results graphically, we provide clear insights into customer behavior and the key drivers of churn.

We train, evaluate, and validate the model using an extensive dataset that includes detailed customer information from the past months. To determine the most effective predictive approach, we test four different algorithms: Random Forest, Logistic Regression, Extreme Gradient Boosting (XGBoost), and Gradient Boosted Machine Trees (GBM). Additionally, we apply parameter tuning to these algorithms during the training process to further enhance the model's accuracy.

The outcome is a sophisticated churn prediction model that significantly improves telecom providers' ability to anticipate and address customer churn. This proactive strategy enables telecom companies to better understand their customers, implement targeted retention measures, and ultimately enhance their profitability.

4.1 ADVANTAGES OF CHURN PREDICTION MODEL:

The proposed system offers several significant advantages over existing systems in managing customer churn in the telecom industry. Firstly, the use of advanced machine learning techniques provides a more accurate and reliable prediction of customer churn compared to traditional statistical models. These techniques allow for the dynamic adaptation to changing customer behaviors, which static rule-based systems fail to address.

Secondly, the novel approach to feature design and selection in our proposed system ensures that the most relevant factors contributing to churn are identified and utilized effectively. This contrasts with the often limited and predefined features used in existing systems, leading to more precise churn predictions.

Thirdly, the comprehensive analysis of customer data, combined with graphical representation of various data scenarios, offers clear insights into customer behavior. This level of detail is not typically provided by current systems, which rely heavily on basic analytics and subjective customer feedback.

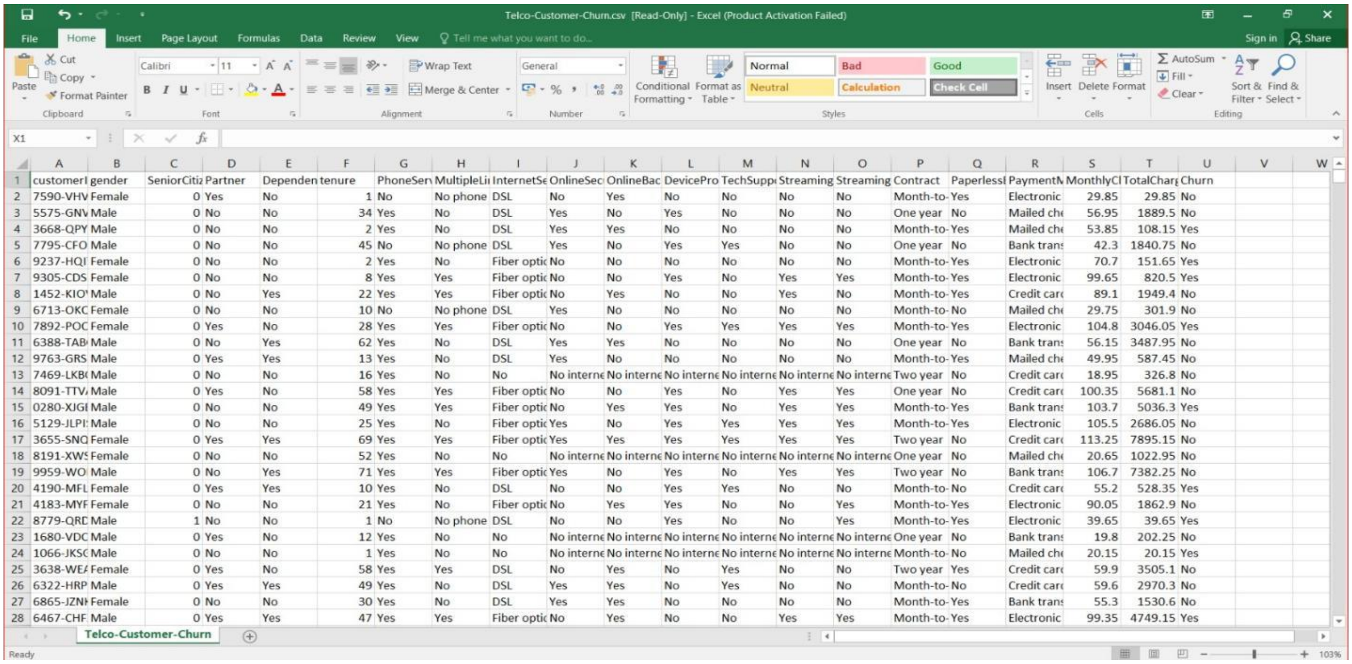
Moreover, the proposed system addresses the issue of data silos by integrating data across different platforms, providing a holistic view of customer behavior. This integration is crucial for accurate churn prediction and is a notable improvement over the fragmented data handling seen in existing systems.

Additionally, our model's use of extensive datasets and parameter tuning during the training process enhances its accuracy and effectiveness. This proactive approach contrasts with the reactive nature of current systems, which often identify churn risks only after significant changes have occurred.

In summary, the proposed system's advanced machine learning methods, novel feature design, comprehensive data analysis, integrated data handling, and proactive approach offer substantial improvements over existing systems.

4.2 METHODOLOGY

In this project, we aim to predict customer churn by analyzing comprehensive customer data, including usage patterns, service interactions, and demographic factors. We start by collecting, cleaning, and preprocessing the data to ensure its quality. Next, we apply feature engineering to extract and transform relevant features. To handle data imbalance, we use methods such as under-sampling and tree-based algorithms. We split the data into 80% for training and 20% for validation, optimizing model performance through hyperparameter tuning. Our system incorporates ensemble learning and deep learning techniques to improve predictive accuracy and adaptability. Continuous learning capabilities enable the model to update dynamically with evolving data trends. This scalable and robust system allows telecom companies to implement personalized retention strategies, enhancing customer satisfaction and reducing churn rates.



	customerI	gender	SeniorCitiz	Partner	Dependen	tenure	PhoneSer	MultipleLi	InternetSe	OnlineSec	OnlineBac	DevicePro	TechSupp	Streaming	Streaming	Contract	Paperless	PaymentN	MonthlyCl	TotalCharg	Churn
2	7590-VHV	Female	0	Yes	No	1	No	No phone	DSL	No	Yes	No	No	No	No	Month-to-	Yes	Electronic	29.85	29.85	No
3	5575-GNV	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No	No	No	One year	No	Mailed che	56.95	1889.5	No
4	3668-QPY	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-to-	Yes	Mailed che	53.85	108.15	Yes
5	7795-CFO	Male	0	No	No	45	No	No phone	DSL	Yes	No	Yes	Yes	No	No	One year	No	Bank trans	42.3	1840.75	No
6	9237-HQI	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	No	No	No	Month-to-	Yes	Electronic	70.7	151.65	Yes
7	9305-CDS	Female	0	No	No	8	Yes	Yes	Fiber optic	No	No	Yes	No	Yes	Yes	Month-to-	Yes	Electronic	99.65	820.5	Yes
8	1452-KIO	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	Yes	No	No	Yes	No	Month-to-	Yes	Credit card	89.1	1949.4	No
9	6713-OKC	Female	0	No	No	10	No	No phone	DSL	Yes	No	No	No	No	No	Month-to-	No	Mailed che	29.75	301.9	No
10	7892-POC	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	No	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	104.8	3046.05	Yes
11	6388-TAB	Male	0	No	Yes	62	Yes	No	DSL	Yes	Yes	No	No	No	No	One year	No	Bank trans	56.15	3487.95	No
12	9763-GRS	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	No	No	No	No	No	Month-to-	Yes	Mailed che	49.95	587.45	No
13	7469-LKB	Male	0	No	No	16	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	Two year	No	Credit card	18.95	326.8	No
14	8091-TTV	Male	0	Yes	No	58	Yes	Yes	Fiber optic	No	No	Yes	No	Yes	Yes	One year	No	Credit card	100.35	5681.1	No
15	0280-XJGI	Male	0	No	No	49	Yes	Yes	Fiber optic	No	Yes	Yes	No	Yes	Yes	Month-to-	Yes	Bank trans	103.7	5036.3	Yes
16	5129-JLPI	Male	0	No	No	25	Yes	No	Fiber optic	Yes	No	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	105.5	2686.05	No
17	3655-SNQ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	No	Credit card	113.25	7895.15	No
18	8191-XWS	Female	0	No	No	52	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	One year	No	Mailed che	20.65	1022.95	No
19	9959-WO	Male	0	No	Yes	71	Yes	Yes	Fiber optic	Yes	No	Yes	No	Yes	Yes	Two year	No	Bank trans	106.7	7382.25	No
20	4190-MFL	Female	0	Yes	Yes	10	Yes	No	DSL	No	No	Yes	Yes	No	No	Month-to-	No	Credit card	55.2	528.35	Yes
21	4183-MYF	Female	0	No	No	21	Yes	No	Fiber optic	No	Yes	Yes	No	No	Yes	Month-to-	Yes	Electronic	90.05	1862.9	No
22	8779-QRD	Male	1	No	No	1	No	No phone	DSL	No	No	Yes	No	No	Yes	Month-to-	Yes	Electronic	39.65	39.65	Yes
23	1680-VDC	Male	0	Yes	No	12	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	One year	No	Bank trans	19.8	202.25	No
24	1066-JKSC	Male	0	No	No	1	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	Month-to-	No	Mailed che	20.15	20.15	Yes
25	3638-WEA	Female	0	Yes	No	58	Yes	Yes	DSL	No	Yes	No	Yes	No	No	Two year	Yes	Credit card	59.9	3505.1	No
26	6322-HRP	Male	0	Yes	Yes	49	Yes	No	DSL	Yes	Yes	No	Yes	No	No	Month-to-	No	Credit card	59.6	2970.3	No
27	6865-JZNK	Female	0	No	No	30	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-to-	Yes	Bank trans	55.3	1530.6	No
28	6467-CHF	Male	0	Yes	Yes	47	Yes	Yes	Fiber optic	No	Yes	No	No	Yes	Yes	Month-to-	Yes	Electronic	99.35	4749.15	Yes

Fig.4.2.1 Telecom customer churn dataset

To implement this project, we have designed following modules:

Data Collection

In this project, we utilize a telecom customer dataset sourced from Kaggle. This dataset includes various features related to customer connections and services, with labels indicating churn (0 for no churn and 1 for churn).

Pre-processing

Our dataset comprises 21 features, and we aim to evaluate and select the most impactful ones for training. Feature engineering and selection methods are used to identify key features, ensuring all 21 are utilized effectively. The data is then split into 80% for training and 20% for testing. The training data includes both features and labels, while the testing data is reserved for evaluating the machine learning model.

Train-Test Split and Model Fitting

We split our dataset into training and testing sets to assess the model's performance on unseen data and to evaluate its generalization capabilities. This is followed by model fitting, a crucial step in the model-building process, where the machine learning model is trained on the training data and then evaluated using the testing data.

4.3 SYSTEM ARCHITECTURE:

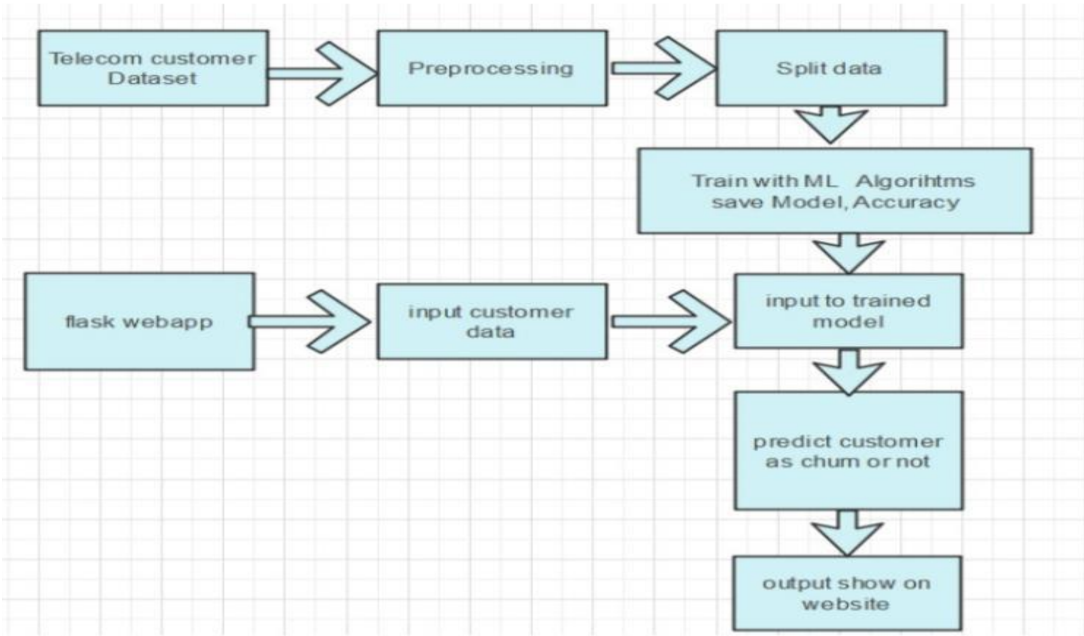


Fig.4.3.1 System architecture

1. Preprocess the Dataset

The first step in the data preparation process is to clean and prepare the telecom customer dataset, which is crucial for ensuring that the data is suitable for machine learning applications. This preprocessing phase involves several key activities:

Handling Missing Values: Identifying and addressing any missing values in the dataset is essential, as these can adversely affect the performance of machine learning models. Common strategies include removing records with missing values, imputing missing data using mean, median, or mode, or employing more complex methods such as K-nearest neighbors (KNN) imputation, depending on the context and the extent of the missing data.

Adjusting Data Types: Data types must be checked and adjusted as necessary to ensure compatibility with machine learning algorithms. For example, categorical variables might need to be converted from string format to numerical format, which is often achieved through label encoding or one-hot encoding. Numerical features may also require normalization or standardization to ensure that they are on a comparable scale.

Encoding Categorical Variables: Machine learning models typically require numerical input. Therefore, categorical variables must be encoded. Label encoding assigns each category a unique integer value, while one-hot encoding creates binary columns for each category, ensuring that the model can interpret these features appropriately without introducing bias related to the order of categories.

2.Split Data

Once the dataset has been preprocessed, it is essential to split the data into training and testing subsets. This division is crucial for effectively evaluating the performance of the machine learning model.

Training Set: This subset is used to train the machine learning model, allowing it to learn patterns and relationships in the data. Typically, a larger portion of the dataset (often around 70-80%) is allocated for training to ensure that the model has enough data to generalize well.

Testing Set: The remaining portion of the data (usually 20-30%) is reserved for testing the model. This testing set is crucial for evaluating how well the model can predict outcomes on unseen data, which is a strong indicator of its performance in real-world scenarios. This split helps to mitigate overfitting, where the model performs well on training data but poorly on new, unseen data.

3.Train with ML Algorithms and Save Model

With the data split into training and testing sets, the next step involves training the machine learning model using suitable algorithms. One commonly used algorithm in this context is the Random Forest Classifier.

Model Training: During training, the Random Forest algorithm constructs multiple decision trees based on different subsets of the training data. It aggregates the predictions from these trees to improve accuracy and reduce the risk of overfitting. This ensemble learning method is particularly effective for handling large datasets with complex interactions between features.

Model Evaluation: After training, the model's accuracy is evaluated using the testing dataset. Key performance metrics such as accuracy, precision, recall, and the F1 score can be computed to assess how well the model performs in predicting churn. This evaluation phase helps identify any adjustments needed to improve model performance.

Saving the Model: Once the model is trained and evaluated, it is saved for future use. Using libraries such as joblib, the trained model can be serialized and stored on disk. This allows for easy retrieval and deployment in a web application without the need to retrain the model each time it is used.

4.Flask Web Application

To create an interactive user experience, a Flask-based web application is developed. This application allows users to input customer data through a web form.

Web Application Development: The Flask framework is chosen for its simplicity and ease of use in building web applications. The application consists of a front-end interface where users can enter their data, and a back-end that processes the input and returns predictions.

5.User Input Handling

After users submit their data via the web form, the application must handle the input efficiently.

Data Processing: The Flask application processes the input data to ensure that it aligns with the model's input requirements. This includes validating the data types, checking for missing values, and applying any necessary transformations to the data to prepare it for analysis.

6.Model Integration

Once the user input has been processed, the next step is to integrate the pre-trained machine learning model into the web application.

Analyzing Input Data: The integrated model receives the processed input data and analyzes it based on the patterns it learned during training. This step is critical for predicting the likelihood of churn effectively.

7.Prediction

Based on the input data and the learned patterns from the training phase, the model makes a prediction regarding whether the customer is likely to churn.

Churn Prediction: The model utilizes its internal decision-making process to evaluate the input data and generate a prediction. This prediction indicates the probability of churn, providing valuable insights into the customer's risk status.

8.Output Display

Finally, the results of the prediction are presented to the user through the web interface.

User Feedback: The application displays the predicted churn likelihood or classification outcome in a clear and comprehensible manner. This output not only informs the user about the churn risk associated with the input data but can also provide recommendations for retention strategies based on the model's insight.

Functional Requirements:

1. User Input
2. Data Processing
3. Model Integration
4. Prediction Output

Non-Functional Requirements:

1. Usability requirement
2. Data Integrity requirement
3. Maintainability requirement
4. Security requirement
5. Reliability requirement
6. Service ability requirement

4.4 ALGORITHM:

The algorithm includes all customer data from previous months. We tested four different algorithms: Logistic Regression, Extreme Gradient Boosting (XGBoost), Gradient Boosted Machine (GBM) Trees, Random Forests, and Decision Trees. To increase the model's accuracy, the algorithms are trained with parameter tuning. This work focuses on optimizing telecom operations with segmentation and churn prediction, ultimately enhancing operational efficiency and customer retention.

Logistic Regression model is utilized to analyze the data using a training dataset while validating its performance with a testing dataset. The model is configured with specific settings, including a regularization strength (C=150) and a maximum iteration limit of 150 for optimization. Initially, the model is trained by fitting it to the training data (X_train and y_train), enabling it to learn patterns and relationships within the data. After training, the model predicts outcomes for the testing dataset (X_test). The accuracy of these predictions is then calculated, indicating how well the model performs, with a reported training accuracy of 80 percent. This means that the model correctly classified 80 percent of the instances in the training data. To further evaluate the model's effectiveness, additional metrics such as the confusion matrix and classification report are generated, offering detailed insights into the model's precision, recall, and overall classification performance.

```
# Initialize the Logistic Regression model with specified hyperparameters
Log_reg = LogisticRegression(
    C=150,            # Inverse of regularization strength; a smaller value specifies stronger regularization
    max_iter=150      # Maximum number of iterations for optimization
```

```
)

# Train (fit) the model on the training dataset
Log_reg.fit(X_train, y_train)

# Make predictions on the test dataset using the trained model
log_pred = Log_reg.predict(X_test)

# Calculate the accuracy score by comparing predictions with actual test labels
accuracy = accuracy_score(log_pred, y_test)

# Generate the confusion matrix to assess the performance of the classifier
confusion_matrix_result = confusion_matrix(log_pred, y_test)

# Create a detailed classification report (precision, recall, f1-score)
classification_report_result = classification_report(log_pred, y_test)

# Print the accuracy score, confusion matrix, and classification report to interpret results
PRINT "Accuracy score:", accuracy
PRINT "Confusion matrix:", confusion_matrix_result
PRINT "Classification report:", classification_report_result
```

The Random Forest Classifier is an ensemble method that combines multiple decision trees to improve prediction accuracy and reduce overfitting. In this setup, 120 trees are built (`n_estimators=120`), each using the Gini impurity (`criterion='gini'`) to decide splits. The model limits each tree's depth to 15 (`max_depth=15`) to avoid overfitting and requires at least 10 samples in each leaf (`min_samples_leaf=10`) and 5 samples to split a node (`min_samples_split=5`).

The classifier is trained on `X_train` and `y_train` using `Rfc.fit`, and predictions are made on `X_test` with `Rfc.predict`. The accuracy score, confusion matrix, and classification report are calculated to assess performance, showing how well the model predicts outcomes and providing insights into its strengths and areas for improvement. This process enables a well-balanced model that can handle complex data without becoming overly complex itself.

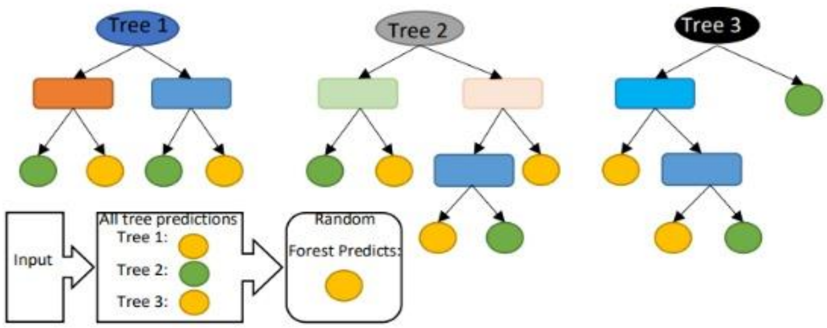


Fig.4.4.1 Random Forest Classifier

```
# Initialize the Random Forest Classifier with specified hyperparameters
Rfc = RandomForestClassifier(
    n_estimators=120,          # Number of trees in the forest
    criterion='gini',          # Split criterion: Gini impurity
    max_depth=15,              # Maximum depth of each tree
    min_samples_leaf=10,       # Minimum samples required to be at a leaf node
    min_samples_split=5        # Minimum samples required to split an internal node
)

# Train (fit) the classifier on the training dataset
Rfc.fit(X_train, y_train)

# Make predictions on the test dataset using the trained model
rfc_pred = Rfc.predict(X_test)

# Calculate the accuracy score by comparing predictions with actual test labels
accuracy = accuracy_score(rfc_pred, y_test)

# Generate the confusion matrix to assess the performance of the classifier
```

```
confusion_matrix_result = confusion_matrix(rfc_pred, y_test)

# Create a detailed classification report (precision, recall, f1-score)
classification_report_result = classification_report(rfc_pred, y_test)

# Print the accuracy score, confusion matrix, and classification report to interpret results
PRINT "Accuracy score:", accuracy
PRINT "Confusion matrix:", confusion_matrix_result
PRINT "Classification report:", classification_report_result
```

Decision Tree Classifier is employed to analyze the data, utilizing a training dataset to learn patterns and a testing dataset to evaluate its performance. The classifier is configured with specific parameters, such as using Gini impurity for splits, a random splitter method, and a requirement of at least 15 samples at each leaf node. The model is trained by fitting it to the training data, which allows it to develop rules for classification based on the input features (X_train and y_train). Once trained, the model makes predictions on the testing dataset (X_test), generating predicted labels for evaluation. The accuracy of these predictions is then calculated, revealing that the model achieves a training accuracy of 80 percent. This indicates that the classifier correctly identifies 80 percent of the instances in the training dataset. Additional performance metrics, including the confusion matrix and classification report, are generated to provide deeper insights into the model’s classification effectiveness, highlighting aspects like precision, recall, and overall accuracy.

```
# Initialize the Decision Tree Classifier with specified hyperparameters
Dtc = DecisionTreeClassifier(
    criterion='gini',           # Split criterion: Gini impurity for determining best splits
    splitter='random',          # Method for choosing the split at each node: random or best
    min_samples_leaf=15         # Minimum samples required to be at a leaf node
)

# Train (fit) the classifier on the training dataset
Dtc.fit(X_train, y_train)
```

```
# Make predictions on the test dataset using the trained model
dtc_pred = Dtc.predict(X_test)

# Calculate the accuracy score by comparing predictions with actual test labels
accuracy = accuracy_score(dtc_pred, y_test)

# Generate the confusion matrix to assess the performance of the classifier
confusion_matrix_result = confusion_matrix(dtc_pred, y_test)

# Create a detailed classification report (precision, recall, f1-score)
classification_report_result = classification_report(dtc_pred, y_test)

# Print the accuracy score, confusion matrix, and classification report to interpret results
PRINT "Accuracy score:", accuracy
PRINT "Confusion matrix:", confusion_matrix_result
PRINT "Classification report:", classification_report_result
```

To enhance the accuracy of the model, the dataset is adjusted to ensure an equal distribution of classes using the SMOTEENN function. Before applying this function, the class distribution shows 4,129 instances for class 0 and 1,505 instances for class 1. After applying the SMOTEENN function, the class distribution becomes more balanced, with class 1 now having 2,418 instances and class 0 having 2,132 instances. This equalization helps address any potential bias in the model due to the imbalanced data.

Subsequently, the balanced dataset is divided into training and testing sets. The model is then retrained using multiple algorithms to evaluate whether there is an improvement in accuracy. This approach allows for a comprehensive comparison of the various models, aiming to determine which algorithm performs best with the newly balanced data and achieves higher predictive accuracy.

The training and testing datasets are utilized after applying the SMOTE technique to enhance the balance of classes. The Decision Tree Classifier is employed to train the model, using the balanced training data as input and the corresponding testing data for evaluation. After training, the model achieves a remarkable accuracy of 91 percent, indicating a significant improvement compared to the 80 percent accuracy observed before applying SMOTE. This increase in accuracy demonstrates the effectiveness of using SMOTE to address class imbalance, allowing the Decision Tree model to learn more robust patterns and make better predictions on the testing dataset. The results highlight the positive impact of balancing the dataset on model performance.

```
# Initialize the Decision Tree Classifier with specified hyperparameters for sampling
Dtc_sampling = DecisionTreeClassifier(
    criterion="gini",           # Split criterion: Gini impurity for determining the best splits
    random_state=100,           # Random seed for reproducibility of results
    max_depth=7,                # Maximum depth of the tree to prevent overfitting
    min_samples_leaf=15         # Minimum number of samples required to be at a leaf node
)

# Train (fit) the classifier on the balanced training dataset with sampling
Dtc_sampling.fit(X_train_sap, y_train_sap)

# Make predictions on the balanced test dataset using the trained model
dtc_sampling_pred = Dtc_sampling.predict(X_test_sap)

# Calculate the accuracy score by comparing predictions with actual test labels
accuracy = accuracy_score(dtc_sampling_pred, y_test_sap)

# Generate the confusion matrix to assess the performance of the classifier
confusion_matrix_result = confusion_matrix(dtc_sampling_pred, y_test_sap)

# Create a detailed classification report (precision, recall, f1-score)
classification_report_result = classification_report(dtc_sampling_pred, y_test_sap)

# Print the accuracy score, confusion matrix, and classification report to interpret results
PRINT "Accuracy score:", accuracy
PRINT "Confusion matrix:", confusion_matrix_result
PRINT "Classification report:", classification_report_result
```

The training and testing datasets, which have been adjusted using the SMOTE technique for better class balance, are utilized to train a Logistic Regression classifier. The model takes the training data as input and the corresponding testing data for evaluation. After training, the model achieves an impressive accuracy of 91 percent, indicating a notable improvement from the 80 percent accuracy observed prior to applying SMOTE. This significant increase demonstrates the effectiveness of using SMOTE to address class imbalance, allowing the Logistic Regression model to learn more effectively and make more accurate predictions on the testing dataset. The results highlight how balancing the dataset can enhance model performance and reliability.

```
# Initialize the Logistic Regression model with specified hyperparameters for sampling
Log_reg_sampling = LogisticRegression(
    C=10,              # Inverse of regularization strength; a smaller value specifies stronger regularization
    max_iter=150       # Maximum number of iterations for optimization
)

# Train (fit) the model on the balanced training dataset with sampling
Log_reg_sampling.fit(X_train_sap, y_train_sap)

# Make predictions on the balanced test dataset using the trained model
Log_sampling_pred = Log_reg_sampling.predict(X_test_sap)

# Calculate the accuracy score by comparing predictions with actual test labels
accuracy = accuracy_score(Log_sampling_pred, y_test_sap)

# Generate the confusion matrix to assess the performance of the classifier
confusion_matrix_result = confusion_matrix(Log_sampling_pred, y_test_sap)

# Print the accuracy score and confusion matrix to interpret results
PRINT "Accuracy score:", accuracy
PRINT "Confusion matrix:", confusion_matrix_result
```

```
# Create a detailed classification report (precision, recall, f1-score)
classification_report_result = classification_report(Log_sampling_pred, y_test_sap)

# Print the classification report to understand the model's performance in detail
PRINT "Classification report:", classification_report_result
```

The model utilizes the training data as input and the corresponding testing data for evaluation. Following the training process, the model achieves a remarkable accuracy of 95 percent. This represents a significant enhancement compared to the 80 percent accuracy recorded before the application of SMOTE. The increase in accuracy underscores the effectiveness of the SMOTE technique in addressing class imbalance, enabling the Gradient Boosting Classifier to better capture patterns in the data and make more accurate predictions on the testing dataset. This improvement highlights the importance of data balancing in enhancing model performance.

```
# Initialize the Gradient Boosting Classifier with default hyperparameters
gbc = GradientBoostingClassifier()

# Train (fit) the classifier on the balanced training dataset with sampling
gbc.fit(X_train_sap, y_train_sap)

# Make predictions on the balanced test dataset using the trained model
pred = gbc.predict(X_test_sap)

# Calculate the accuracy score by comparing predictions with actual test labels
accuracy = accuracy_score(pred, y_test_sap)

# Generate the confusion matrix to assess the performance of the classifier
confusion_matrix_result = confusion_matrix(pred, y_test_sap)

# Print the accuracy score and confusion matrix to interpret results
PRINT "Accuracy score:", accuracy
PRINT "Confusion matrix:", confusion_matrix_result
```

```
# Create a detailed classification report (precision, recall, f1-score)
classification_report_result = classification_report(pred, y_test_sap)

# Print the classification report to understand the model's performance in detail
PRINT "Classification report:", classification_report
```

4.5 SAMPLE CODE:

```
from flask import Flask, render_template, request

import pickle

import numpy as np

from database import *

from sklearn.preprocessing import LabelEncoder

app = Flask(__name__,static_url_path='/static')


# Load the machine learning model

@app.route('/p')

def p():

return render_template('index.html')


@app.route('/')

def m():

return render_template('main.html')


@app.route('/l')

def l():
```

```
return render_template('login.html')


@app.route('/h')
    def h():
return render_template('home.html')


@app.route('/r')
def r():
return render_template('register.html')


@app.route('/m')
def menu():
return render_template('menu.html')


@app.route("/register",methods=['POST','GET'])
def signup():
if request.method=='POST':
username=request.form['username']
email=request.form['email']
password=request.form['password']
status = user_reg(username,email,password)
if status == 1:
return render_template("/login.html")
else:
```

```
return render_template("/register.html",m1="failed")
@app.route("/login",methods=['POST','GET'])
def login():
if request.method=='POST':
username=request.form['username']
password=request.form['password']
status = user_loginact(request.form['username'], request.form['password'])
print(status)
if status == 1:
return render_template("/home.html", m1="sucess")
else:
return render_template("/login.html", m1="Login Failed")


@app.route('/predict', methods=['POST'])
def predict():
Dependents = request.form['1']
tenure = float(request.form['2'])
OnlineSecurity = request.form['3']
OnlineBackup = request.form['4']
DeviceProtection = request.form['5']
TechSupport = request.form['6']
Contract = request.form['7']
PaperlessBilling = request.form['8']
MonthlyCharges = float(request.form['9'])
```

```
TotalCharges = float(request.form['10'])


model = pickle.load(open('Model.sav', 'rb'))

data = [[Dependents, tenure, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, Contract,
PaperlessBilling, MonthlyCharges, TotalCharges]]

df = pd.DataFrame(data, columns=['Dependents', 'tenure', 'OnlineSecurity',

'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Contract',

'PaperlessBilling', 'MonthlyCharges', 'TotalCharges'])


categorical_feature = {feature for feature in df.columns if df[feature].dtypes == 'O'}


encoder = LabelEncoder()

for feature in categorical_feature:

df[feature] = encoder.fit_transform(df[feature])


single = model.predict(df)

probability = model.predict_proba(df)[:, 1]

probability = probability*100


if single == 1:

op1 = "This Customer is likely to be Churned!"

op2 = f"Confidence level is {np.round(probability[0], 2)}"

else:

op1 = "This Customer is likely to be Continue!"
```

```
op2 = f"Confidence level is {np.round(probability[0], 2)}"

return render_template("result.html", op1=op1, op2=op2)

if __name__ == "__main__":
app.run(debug=False, port=5112)
```

CHAPTER 5
RESULTS AND DISCUSSION

In this section, the outcomes of the implemented algorithms for customer churn telecom are presented. The results focus on sophisticated churn prediction model that significantly improves telecom providers' ability to anticipate and address customer churn.

5.1 INTERFACE :

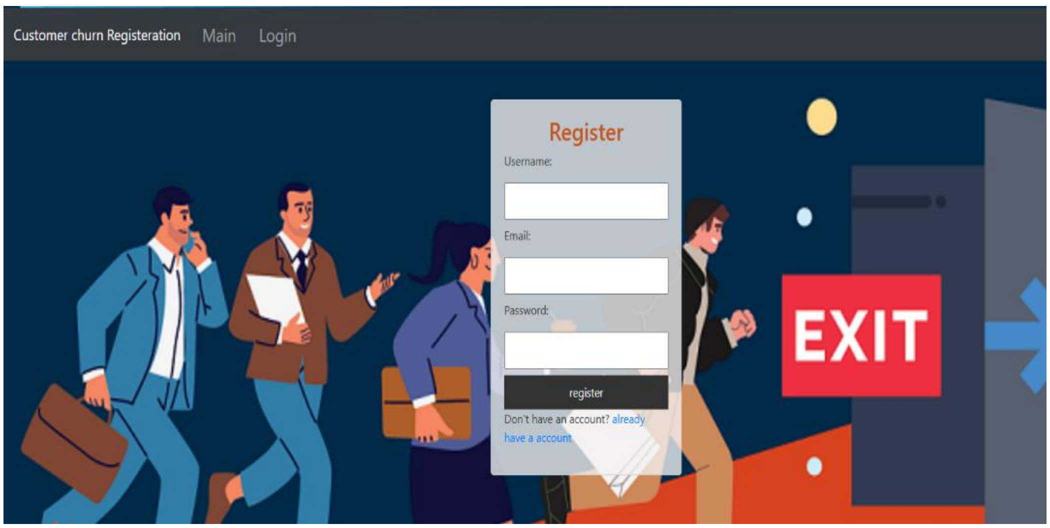


Fig.5.1.1 Registration Page

A registration page for a customer churn prediction system requires several key components. First, input fields are used to collect essential information like username, email, and password, which form the basis for user authentication. Proper form validation is crucial to ensure the data is accurate and secure—validating email formats and enforcing strong password requirements helps maintain the integrity of user input.

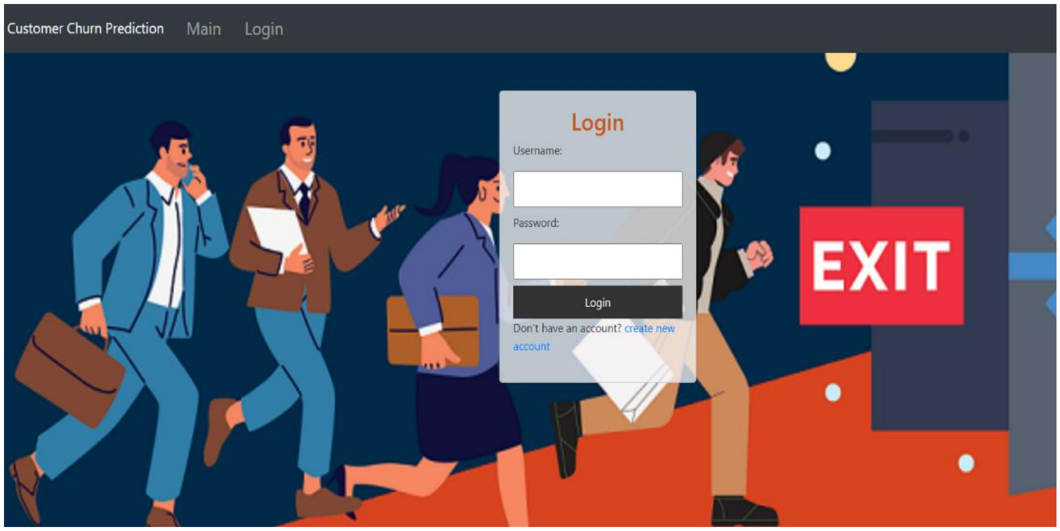


Fig.5.1.2 Login Page

A login page for a customer churn prediction system serves as the gateway for users to access the application. It typically includes **input fields** for the username or email and password, enabling users to authenticate their credentials. **Form validation** is essential to ensure users enter correct and complete information, such as validating that fields are not left empty and that the email format is correct when applicable.

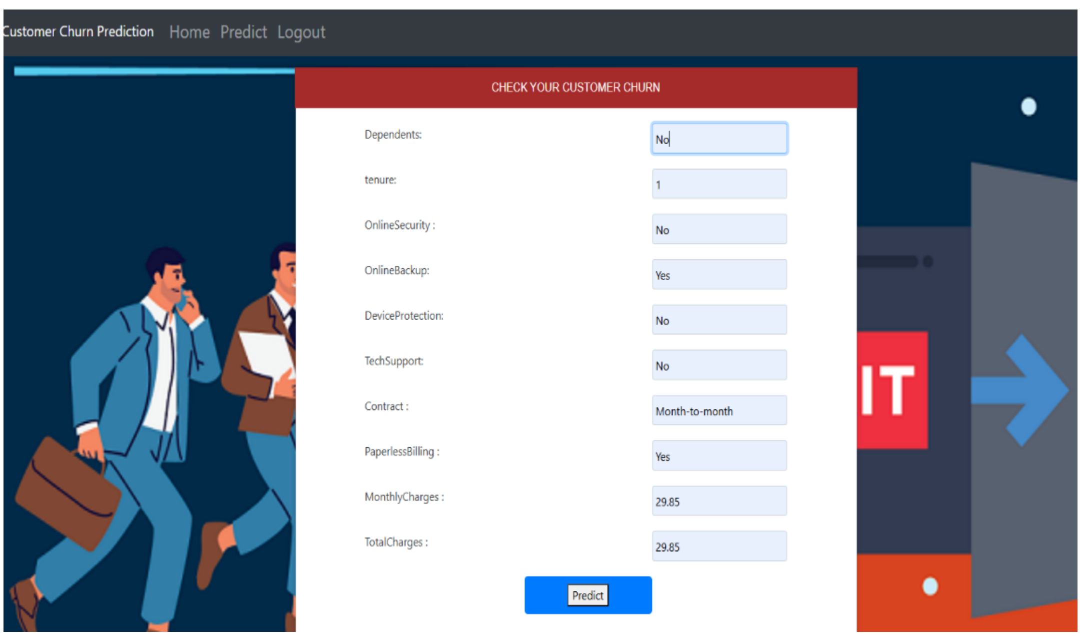


Fig.5.1.3 Prediction Page

A customer churn checking page for a customer churn prediction system serves as the interface where users can input customer data to assess the likelihood of churn. This page typically includes input fields to enter customer attributes such as demographic details, transaction history, and usage patterns.

The page integrates with a backend machine learning model trained to predict churn based on these inputs. After the data is submitted, the model analyzes the inputs and returns a prediction indicating the likelihood of the customer leaving. The results are usually displayed in an easy-to-understand format, such as a probability score, risk category (e.g., high, medium, low), or a visual indicator like a color-coded status.

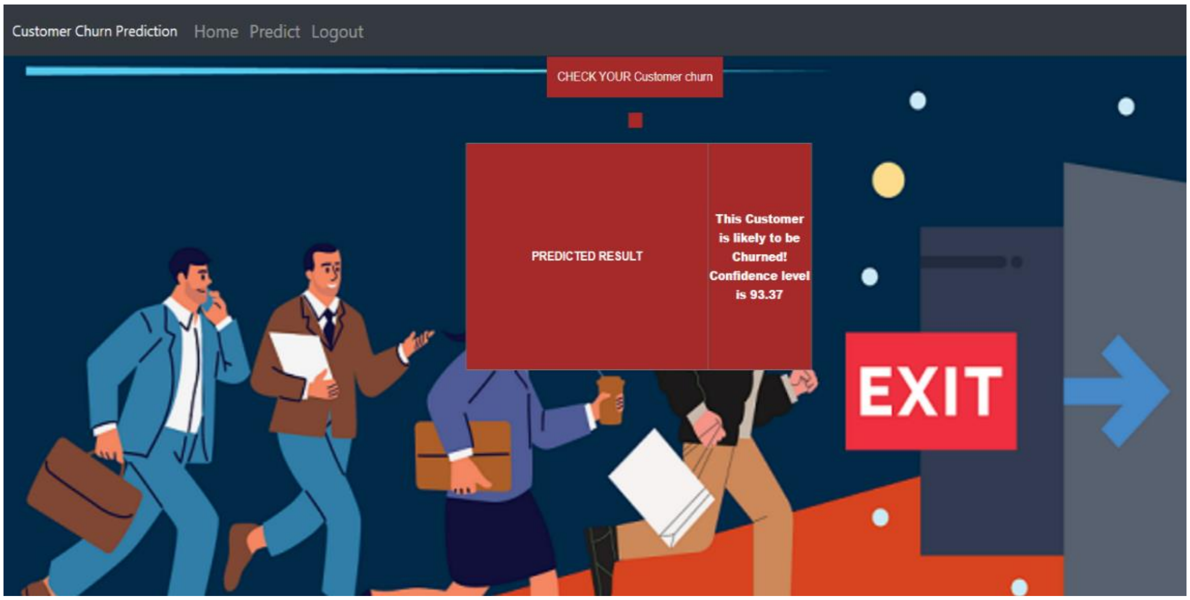


Fig.5.1.4 Result page

A Churn Prediction Result Page for a customer churn prediction system provides users with insights into the likelihood of customer churn based on their input data. This page typically displays the prediction results, including whether a customer is likely to churn or not, along with key metrics and factors influencing the prediction.

Key Components of the Churn Prediction Result Page:

1. Prediction Outcome: Clearly displays the prediction result, such as "Likely to Churn" or "Not Likely to Churn," giving a quick overview of the customer’s status.

2. Probability Score: Shows the probability or confidence level of the prediction, such as “Churn Probability: 93.37%,” helping users understand the certainty of the result.

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

In conclusion, this project presents a robust approach for telecom companies aiming to enhance profitability by accurately predicting customer churn, a critical factor in retaining valuable clients and sustaining revenue streams. The research underscores the value of predictive analytics in identifying customers at risk of attrition, allowing companies to take proactive measures and improve retention strategies effectively.

The primary objective was to develop a system capable of accurately forecasting churn, which was achieved through models that demonstrated strong performance, reflected by high AUC (Area Under the Curve) values. High AUC values are indicative of the model's ability to distinguish between customers likely to churn and those who are not, making it a reliable tool for telecom companies to preemptively address potential revenue losses.

The study employed an 80/20 data split for training and validation to optimize and rigorously test the model. This method ensured that the model was not only well-tuned to the data but also capable of generalizing well on new, unseen data. Key steps in refining the model included hyperparameter tuning and validation, which enabled the selection of optimal model settings, enhancing predictive accuracy. The study also emphasized feature engineering, transformation, and selection methods, which were integral to preparing the data for machine learning algorithms by enhancing the relevance and quality of the input features.

A notable challenge addressed in the research was the significant imbalance in the dataset, with only 5% of entries representing customer churn cases. This skewed distribution can bias models, making it difficult to predict churn accurately. To mitigate this, techniques such as under-sampling were applied, reducing the number of non-churn cases to balance the data. Additionally, tree-based methods, which are less sensitive to class imbalance, were selected as the primary algorithms for building the churn prediction model. The study identified four tree-based algorithms particularly suitable for this type of classification problem: Decision Tree, Random Forest, and Gradient Boosting Machine (GBM). Each of these algorithms was chosen for its interpretability, accuracy, and resilience to data imbalance, which contributed to more effective churn prediction.

Key Findings:

1. **High Predictive Performance:** The model achieved high AUC values, underscoring its strong capability to predict customer churn and helping telecom companies focus their retention efforts on at-risk customers.
2. **Data Preparation and Feature Engineering:** Effective feature engineering and data transformations were instrumental in enhancing model performance, demonstrating the importance of preprocessing in machine learning pipelines.
3. **Addressing Data Imbalance:** The study tackled class imbalance by employing under-sampling and selecting tree-based algorithms, which handle skewed distributions effectively and maintain prediction accuracy.
4. **Algorithm Selection**: The choice of Decision Tree, Random Forest, and GBM algorithms was validated as they exhibited robustness in handling the imbalanced data and produced reliable predictions for customer churn.

In summary, this study not only developed a predictive model for customer churn in the telecom industry but also established a methodology that integrates data preparation, handling class imbalance, and algorithm selection. This approach offers telecom providers a systematic and highly effective means to manage customer churn, improve retention rates, and, ultimately, enhance profitability by preemptively addressing factors leading to customer attrition.

6.2 FUTURE SCOPE

The proposed churn prediction system provides a strong foundation for future enhancements, empowering telecom providers to better manage and mitigate customer churn. Here are several potential areas for advancing this system:

1. **Integration with Real-Time Data:** Future iterations can incorporate real-time data streams, enabling the system to analyze and predict churn events as they happen. This would allow telecom providers to respond promptly to churn indicators, facilitating immediate and more effective interventions that address customer dissatisfaction before it escalates.
2. **Incorporation of Advanced Algorithms:** With continuous advancements in machine learning and

artificial intelligence, the system could be enhanced by integrating more sophisticated algorithms. Emerging techniques such as deep learning, reinforcement learning, and hybrid models offer potential for further improving prediction accuracy, adapting dynamically to complex and evolving customer behaviors.

3. **Enhanced Personalization:** An enriched system could personalize retention strategies to individual customers. Through customer segmentation and profiling, telecom providers could tailor interventions according to unique preferences, usage patterns, and service needs, offering a customized approach that strengthens customer loyalty and satisfaction.
4. **Expansion to Multi-Channel Data Sources:** To achieve a comprehensive understanding of customer behavior, future systems could integrate data from a variety of channels, including social media interactions, customer service calls, and IoT devices. By broadening the data sources, the system would gain a multidimensional perspective on customer interactions, enabling more accurate predictions and insights.

REFERENCES

[1] Gerpott TJ, Rams W, Schindler A. Customer retention, loyalty, and satisfaction in the German mobile cellular telecommunications market. Telecommun Policy. 2001;25:249–69.

[2] Wei CP, Chiu IT. Turning telecommunications call details to churn prediction: a data mining approach Expert Syst Appl. 2002;23(2):103–12.

[3] Qureshii SA, Rehman AS, Qamar AM, Kamal A, Rehman A. Telecommunication subscribers' churn prediction model using machine learning. In: Eighth international conference on digital information management. 2013. p. 131–6.

[4] Ascarza E, Iyengar R, Schleicher M. The perils of proactive churn prevention using plan recommendations: evidence from a feld experiment. J Market Res. 2016;53(1):46–60.

[5] Bott. Predicting customer churn in telecom industry using multilayer preceptron neural networks:modeling and analysis. Igarss. 2014;11(1):1–5.

[6] Umayaparvathi V, Iyakutti K. A survey on customer churn prediction in telecom industry: datasets, methods and metric. Int Res J Eng Technol. 2016;3(4):1065–70.

[7] Yu W, Jutla DN, Sivakumar SC. A churn-strategy alignment model for managers in mobile telecom. In: Communication networks and services research conference, vol. 3. 2005. p. 48–53.

[8] Burez D, den Poel V. Handling class imbalance in customer churn prediction. Expert Syst Appl.2009;36(3):4626–36.

[9] Zhan J, Guidibande V, Parsa SPK. Identifcation of top-k infuential communities in big networks. J Big Data. 2016;3(1):16. https://doi.org/10.1186/s40537-016-0050-7.

[10] Barthelemy M. Betweenness centrality in large complex networks. Eur Phys J B. 2004;38(2):163–8. https://doi. org/10.1140/epjb/e2004-00111-4.

[11] Elisabetta E, Meyerhenke H, Staudt CL. Approximating betweenness centrality in large evolving networks. CoRR. 2014. arxiv:1409.6241.

[12] Brandusoiu I, Toderean G, Ha B. Methods for churn prediction in the prepaid mobile telecommunications industry. In: International conference on communications. 2016. p. 97–100

[13] Choudhury, A., & Rahman, S. "A Comparative Study of Predictive Models for Customer Churn Prediction in Telecom Industry." International Journal of Data Science and Analytics, vol. 9, no. 2, pp. 151-162, 2020.

[14] Guha, S., & Chakraborty, S. "Churn Prediction in Telecom Industry Using Big Data Analytics." Journal of Big Data, vol. 4, no. 1, pp. 2, 2017.

[15] Rahman, M., & Hossain, M. "The Impact of Customer Feedback on Churn Prediction: A Study on the Telecom Industry." International Journal of Information Management, vol. 51, pp. 102-112, 2020.

[16] Ghazali, A., & Nor, R. "Big Data Analytics in Telecom Industry: A Review." Journal of Telecommunications and Information Technology, vol. 4, pp. 6-16, 2018.

[17] Dutta, S., & Dey, N. "The Role of Big Data in Predictive Analytics for Churn Management." Journal of Business Research, vol. 124, pp. 631-641, 2021.

[18] Zhan, Y., & Zhang, L. "Using Big Data to Improve Churn Prediction in Telecom Industry." Journal of Industrial Management & Data Systems, vol. 119, no. 8, pp. 1735-1751, 2019.

[19] Breiman, L. "Random Forests." Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[20] Chen, T., & Guestrin, C. "XGBoost: A Scalable Tree Boosting System." In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794, 2016.

[21] Berry, L.L., & Parasuraman, A. "Marketing Services: Competing Through Quality." The Free Press, 1991.

[22] Buttle, F. "Customer Relationship Management: Concepts and Tools." Routledge, 2009.

[23] McKinsey & Company. "The Future of Customer Engagement: How to Win in the Age of the Customer." McKinsey & Company, 2016.

[24] CCRM. "Customer Relationship Management: An Overview." Customer Care Research Magazine, vol. 12, no. 2, pp. 34-45, 2018.

[25] GDPR. "General Data Protection Regulation (GDPR)." European Parliament, 2016.

[26] Tene, O., & Polonetsky, J. "A Theory of the Dark Side of Data." Harvard Law Review, vol. 126, no. 2, pp. 33-58, 2013.

[27] Nascimento, J.C., & Almeida, J.M. "The Role of Natural Language Processing in Churn Prediction." Journal of Business Research, vol. 116, pp. 303-310, 2020.

[28] Asadi, S., & Shahi, S. "Text Mining for Churn Prediction in Telecom." Journal of Computer and System Sciences, vol. 104, pp. 181-188, 2019.

[29] Kaur, G., & Kaur, S. "The Importance of Data Collaboration in Customer Retention Strategies." Journal of Business Analytics, vol. 3, no. 3, pp. 162-173, 2020.

[30] Srivastava, R.K., & Srivastava, S. "Leveraging Technology for Customer Retention: A Study of Telecom Sector." Journal of Strategic Marketing, vol. 27, no. 4, pp. 324-336, 2019.

<table>
<tr><td>1</td><td>Name of the Student</td><td colspan="4">Dandu Neha
Mohd Ameesha
Mukka Dheeraj</td></tr>
<tr><td>2</td><td>Email ID and Phone Number</td><td colspan="4">21951A05B8@iare.ac.in 9346176650
21951A05A4@iare.ac.in 6305878336
21951A0538@iare.ac.in 7815855613</td></tr>
<tr><td>3</td><td>Roll Number</td><td colspan="4">21951A05B8
21951A05A4
21955A0538</td></tr>
<tr><td>4</td><td>Date of submission</td><td colspan="4"></td></tr>
<tr><td>5</td><td>Name of the Guide</td><td colspan="4">Ms.K.Sangeeta</td></tr>
<tr><td>6</td><td>Title of the project work/ research article</td><td colspan="4">Optimizing Telecom Operations with Segmentation and Churn Prediction</td></tr>
<tr><td>7</td><td>Department</td><td colspan="4">Computer Science and Engineering</td></tr>
<tr><td>8</td><td>Details of the payment</td><td colspan="4"></td></tr>
<tr><td>9</td><td>No. of times submitted</td><td colspan="4">First / Second / Third
(First time – Free; Second time – Rs 200/-; Third – Rs 500/-;There after multiple of third)</td></tr>
<tr><td rowspan="2">10</td><td rowspan="2">Similarity Content (%) (up to 25%acceptable)</td><td>1st</td><td>2nd</td><td>3rd</td><td>4th</td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="6">For R & D Centre Use</td></tr>
<tr><td colspan="2">Date of plagiarism check</td><td colspan="4"></td></tr>
<tr><td colspan="2">Similarity report percentage</td><td colspan="4"></td></tr>
<tr><td colspan="2">R&D staff Name and Signature</td><td colspan="4"></td></tr>
<tr><td colspan="6">I / We hereby declare that, the above mentioned research work is original & it doesn’t contain any plagiarized contents. The similarity index of this research work is……………..
Justification for similarity index:
……………………………………………………………...………………...…...…………………………
………
………
………
………
……………………………………………………………………………………………………...…………
……...………………...…………..</td></tr>
<tr><td colspan="6">Signature of Student   Signature of the Guide</td></tr>
</table>