CUSTOMER CHURN PREDICTION IN SUBSCRIPTION SERVICES

A PROJECT REPORT

Submitted by

Srishti Raj (20BCS4391)

Pooja Verma (20BCS4439)

Harsh Kumar Deo (20BCS4393)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE WITH SPECIALIZATION BIG DATA ANALYTICS

Under the Supervision of:

Mr. Pulkit Dwivedi

Chandigarh University

Gharuan, Punjab-140413

January 2024

BONAFIDE CERTIFICATE

Certified that the report "**Customer Churn Prediction in Subscription Services**" is the bonafide work of **Srishti Raj(20BCS4391)**,**Pooja Verma(20BCS4439)** , and **Harsh Kumar Deo(20BCS4393)** who carried out the project work under my guidance and supervision.

SIGNATURE

Dr. Aman Kaushik

HEAD OF THE DEPARTMENT

Apex Institute of Technology

SIGNATURE

Mr. Pulkit Dwivedi

SUPERVISOR

Assistant Professor

Apex Institute of Technology

Submitted for the project viva-voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The Satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless co-operation made it possible, whose constant guidance and encouragement crown all efforts with success.

We would like to express our deep gratitude to our project guide Prof. Pulkit Dwivedi,sir Assistant Professor, Department of Computer Science and Engineering, CU, for his guidance with unsurpassed knowledge and immense encouragement.

I would like to express my heartful thanks to our parents for their unflinching support and constant encouragement throughout the period of our project work to make it a successful one. I would like to thank all the teaching and non-teaching staff members of Computer Science and Engineering, who have extended their full cooperation during the course of my minor project. I thank all my friends who helped me sharing knowledge and by providing material to complete the project in time.

Submitted By:

Srishti Raj - 20BCS4391

Pooja Verma -20BCS4439

Harsh Kumar Deo-20BCS4393

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Customer churn, the loss of subscribers to a service, poses a significant challenge for telecommunication companies in today's competitive landscape. Retaining existing customers is demonstrably more cost-effective than acquiring new ones. Predicting churn allows telecom providers to proactively intervene and develop targeted strategies to retain at-risk customers.

This paper explores the application of machine learning (ML) techniques for customer churn prediction in telecommunication subscription services. We delve into the various factors influencing churn, including customer demographics, billing information, service usage patterns, and customer support interactions. Techniques for data acquisition, pre-processing, and feature engineering are discussed to prepare the data for effective ML model training.

A comparative analysis of different ML algorithms commonly employed for churn prediction, such as Logistic Regression, Decision Trees, Random Forests, and Gradient Boosting Machines, is presented. The strengths and weaknesses of each approach are explored, along with considerations for model selection and evaluation. The paper emphasizes the importance of model interpretability, enabling telecom companies to understand the key drivers of churn and tailor retention strategies accordingly. The effectiveness of churn prediction models is assessed using relevant metrics like accuracy, precision, recall, and F1-score.

This ensures telecom companies remain equipped to address customer churn proactively and secure long-term customer relationships in a competitive environment.

ABBREVIATIONS

- LLM - logit leaf model
- KNN - K-Nearest Neighbour
- CCP - Customer Churn Prediction
- ANN - Artificial Neural Network
- SVM - Support Vector Machine
- LLP - Lower Limiting Point
- ULP - Upper Limiting Point
- IQR - Inter Quartile Range
- CRM - Customer Relationship Management
- ROC - Receiver Operating Characteristic Curve
- CSV - Comma Separated Value
- API - Application Programming Interface
- EDA - Exploratory Data Analysis

CHAPTER NO - 1

INTRODUCTION

In an era of fierce competition and ever-evolving technological landscapes, the telecommunication industry faces a constant battle: retaining its valuable customer base. Customer churn, the phenomenon of subscribers terminating their service plans, represents a significant financial loss for telecom companies. Studies reveal that acquiring new customers can be five to ten times more expensive than retaining existing ones [1]. This underscores the critical need for proactive strategies to identify customers at risk of churning and implement targeted interventions to foster loyalty and prevent defection.

One of the most powerful tools in this fight against churn is customer churn prediction. By leveraging the vast amount of customer data collected by telecommunication companies, advanced machine learning (ML) techniques can identify patterns and trends that indicate a subscriber's propensity to churn. This predictive power allows companies to anticipate customer dissatisfaction and take preventive measures before churn actually occurs.

The impact of customer churn on a telecommunication company's bottom line is undeniable. Lost subscriber revenue directly translates into decreased profits. However, the financial implications go beyond immediate monetary losses. High churn rates can damage a company's reputation, deterring potential customers and hindering future growth. Furthermore, churn disrupts the delicate balance of network utilization, potentially leading to capacity issues and negatively impacting service quality for remaining subscribers.

The factors influencing customer churn in the telecommunication industry are multifaceted. Customer demographics, such as age, income level, and location, can play a role. Billing-related factors like pricing plans, payment history, and hidden fees can also contribute to dissatisfaction and churn.

Service usage patterns are another crucial area of analysis. Customers who utilize a limited portion of their subscribed services may be more likely to churn, indicating a mismatch between their needs and the offered plan. Conversely, excessive use of data, minutes, or international calls could signal a need for an upgraded plan, which, if not addressed, might lead to frustration and churn.

Customer support interactions are another valuable source of data for churn prediction. The frequency and nature of these interactions can reveal underlying issues with service

quality, billing disputes, or lack of technical support. By analyzing customer support data, telecommunication companies can identify areas for improvement and proactively address potential churn triggers.

The ability to predict churn empowers telecommunication companies to develop targeted retention strategies. Armed with insights gleaned from churn prediction models, companies can tailor personalized offers to at-risk customers. This could include discounted plans, bundled services, loyalty rewards programs, or improved customer support initiatives. Early intervention demonstrably increases the success rate of retention efforts.

Furthermore, churn prediction models allow for a data-driven approach to customer segmentation. By grouping customers with similar churn risk profiles, companies can allocate resources more effectively, focusing targeted retention efforts on those most likely to churn. This ensures valuable resources are not wasted on low-risk customers.

The benefits of implementing a robust customer churn prediction system extend beyond immediate financial gains. By prioritizing customer retention, telecommunication companies foster loyalty and satisfaction. This translates into positive word-of-mouth marketing, attracting new customers and further enhancing brand image. In a competitive industry, customer loyalty is a powerful differentiator, leading to long-term market share growth and increased profitability.

1.1 Problem Identification:

Customer churn is a critical issue for subscription-based businesses. It refers to the phenomenon where customers discontinue using a service and cancel their subscriptions. This can lead to significant revenue loss and hinder a company's growth.By predicting churn, businesses can take proactive steps to retain these customers, such as offering discounts, providing additional value, or addressing any underlying issues that might be causing dissatisfaction.

Here's a breakdown of the problem identification for customer churn prediction:

- **High Cost of Customer Acquisition:** Acquiring new customers is often much more expensive than retaining existing ones. Subscription services rely on a consistent customer base, so churn can significantly impact the bottom line.

- **Missed Opportunities:** Churning customers represent lost potential for upselling, cross-selling, and generating long-term revenue.
- **Negative Impact on Brand Image:** A high churn rate can signal dissatisfaction with the service, potentially deterring potential customers and harming brand reputation.
- **Difficulty in Identifying At-Risk Customers:** Without a system for identifying customers at risk of churn, businesses struggle to take proactive measures to retain them.

Predicting customer churn allows subscription services to:

- **Proactively intervene:** By identifying customers likely to churn, businesses can implement targeted retention strategies like personalized offers, improved customer support, or educational resources.
- **Improve product and service offerings:** Understanding the reasons behind churn helps businesses identify areas for improvement in their product or service, leading to a more satisfied customer base.
- **Optimize marketing and sales efforts:** Resources can be directed towards retaining valuable customers rather than solely focusing on acquiring new ones.

Customer churn is a major challenge for subscription services. By implementing customer churn prediction models, businesses can gain valuable insights into customer behavior and take proactive steps to retain their valuable customer base.

1.2 Purpose

The purpose of customer churn prediction in subscription services is to identify customers who are at high risk of cancelling their subscriptions. This allows businesses to take proactive steps to retain these customers, such as offering discounts, providing additional value, or addressing any underlying issues that might be causing dissatisfaction.

By predicting churn, subscription services can:

- Reduce customer churn rate, which directly translates to increased revenue and customer lifetime value.
- Improve customer satisfaction by identifying areas for improvement and proactively addressing them before customers churn.
- Optimize marketing campaigns by targeting existing customers who are at risk of churning with personalized retention offers.
- Allocate resources more effectively by focusing efforts on retaining valuable customers.

In conclusion, customer churn in the telecommunication industry represents a significant financial and strategic challenge. However, by harnessing the power of customer churn prediction models, telecommunication companies can gain a crucial advantage. This paper will delve deeper into the intricacies of customer churn prediction, exploring the various data sources, machine learning algorithms, and model evaluation metrics employed to identify at-risk customers. Through a comprehensive exploration of these topics, this paper aims to equip telecommunication companies with the knowledge and tools necessary to proactively address customer churn and build a loyal, sustainable customer base in a dynamic and competitive environment.

CHAPTER NO - 2

LITERATURE SURVEY

2.1 Literature Review

Customer churn prediction is a crucial task for subscription-based businesses as it helps in retaining existing customers and maximizing revenue. In this paper, we present a comprehensive review of related work in the domain of customer churn prediction in subscription services. We discuss various approaches, methodologies, and techniques employed by researchers to address this work. D. Manzano. [1]introduce the concept of the architecture of a churn prediction system that utilizes stream mining techniques. Churn prediction refers to the task of identifying customers who are likely to stop using a service or product, commonly observed in subscription-based businesses. Stream mining deals with analyzing data streams in real-time to extract patterns and make predictions. S. Babu [2] introduces a survey focused on understanding the factors influencing churn in the telecommunications industry and how data mining techniques can be applied to analyze and predict churn behavior. A. Idris [3] introduce performance of different tree-based ensemble classifiers in conjunction with three distinct feature selection methods: maximum relevancy and minimum redundancy (mRMR) and F-score based selection schemes. Our focus is on addressing the challenging problem of churn prediction in the telecommunications industry.A.Amin[5] introduce big data analysis techniques to delve into historical churn customer data, aiming to construct a robust churn prediction model. By scrutinizing user characteristics, the study identifies customers at higher risk of churning beforehand. Subsequently, it devises targeted strategies and implements a series of retention activities tailored to retrieve these customers. L. Zhao [6]introduce A new algorithm termed KLMM (K-local maximum margin) has been introduced for feature extraction. This method delves into diversification subspace partition rules, thereby constructing a corresponding potential field structure. By scrutinizing the data source in terms of scalability dimensions, it uncovers the inherent connection between data attributes and classification outcomes. The derived features exhibit a capability to diminish the dimensionality of churn prediction in telecom data, offering potential advancements in this domain.B. Shah [7] introduce capitalizes on the notion of data certainty to refine churn prediction models. Data certainty

denotes the reliability and confidence level attributed to the data at hand. Through integrating metrics of data certainty into their predictive frameworks, the authors strive to heighten the precision of identifying customers prone to churn.S. A. Qureshi[9] introduce a churn prediction model for businesses aiming to identify customers who are likely to leave and take proactive steps to retain them. However, one common challenge faced in building such models is class imbalance, where the number of churners is significantly lower than non-churners. To address this issue, various re-sampling methods can be employed. These methods involve techniques like over-sampling the minority class, under-sampling the majority class, or using more advanced techniques like SMOTE (Synthetic Minority Over-sampling Technique). By employing these methods, the model can achieve a better balance between the classes, resulting in more accurate predictions and effective retention strategies, ultimately leading to improved customer retention rates and business success.V. Lazarov[10] introduce machine learning, data mining, and hybrid approaches. These methods play a pivotal role in business decision-making and Customer Relationship Management (CRM) by enabling the identification, anticipation, and retention of churning customers. Among these techniques, decision trees stand out as a widely recognized tool for predicting issues associated with client turnover. Joao B.[14] This study introduces an innovative framework for Customer Churn Prediction (CCP) within the banking sector, particularly addressing the challenge of rare churn events that persist over time. The rarity of these events often undermines the effectiveness of traditional techniques designed for binary classification. Our objectives are outlined as follows: to present and validate a data preprocessing phase that integrates various approaches, including Feature Engineering (FE) tailored to the retail banking context, Inverse Density Tree (IDT) oversampling (IDT-over), and IDT undersampling (IDT-under).Nikita Khandelal[15] introduce Various machine learning classification models, including Random Forest, Logistic Regression, K Nearest Neighbor (KNN), AdaBoost, Decision Tree, Support Vector Machines (SVM), and Artificial Neural Networks (ANN), have been utilized for predicting customer churn. These models analyze historical customer data encompassing demographics, transaction history, customer interactions, and usage trends to forecast future churn. The selection of the most suitable model should be guided by the unique features of the industry under consideration. Factors such as data availability, client behavior, and industry-specific nuances necessitate tailored strategies for businesses operating in diverse sectors like healthcare, banking,

online retail, and telecommunications.A.D.Caigny[16] introduce Decision trees and logistic regression are widely used algorithms in customer churn prediction due to their strong predictive performance and interpretability. However, decision trees may struggle with linear relationships between variables, while logistic regression may face challenges with interaction effects among variables. To address these limitations, a novel hybrid algorithm called the logit leaf model (LLM) is proposed.The LLM aims to enhance classification accuracy while retaining model comprehensibility. It operates in two stages: segmentation and prediction.This hybrid approach is compared against traditional methods such as decision trees, logistic regression, random forests, and logistic model trees in terms of predictive performance and comprehensibility. By leveraging segmentation and individualized modeling, the LLM offers a promising solution to improve churn prediction accuracy while maintaining interpretability.S. Sivakumar[17] introduce the fresh perspective on churn predictors by integrating them with an organizational competitiveness strategy. Through factor analysis, the model establishes connections between key churn predictors and the overarching competitive strategy of the organization. This innovative approach sheds light on the intricate relationship between customer churn and strategic decision-making, providing valuable insights for businesses aiming to enhance their competitive edge.Makhtar.M[20]introduce a novel classification model grounded in Rough Set Theory for categorizing customer churn. The findings of the research demonstrate that the proposed Rough Set classification model surpasses existing models, resulting in substantial improvements in accuracy. This study thus presents a promising advancement in the realm of churn prediction, offering a more effective approach for businesses to identify and manage customer churn.Amin A.[21] introduce six prominent sampling techniques and conduct a comparative analysis of their performances. These techniques include the mega-trend diffusion function (MTDF), synthetic minority oversampling technique, adaptive synthetic sampling approach, couples top-N reverse k-nearest neighbor, majority weighted minority oversampling technique, and immune centroids oversampling technique. Additionally, we evaluate four rules-generation algorithms: the learning from example module, version 2 (LEM2), covering, exhaustive, and genetic algorithms, utilizing publicly available datasets. Our empirical findings indicate that MTDF and rules-generation based on genetic algorithms exhibit superior predictive performance compared to the other evaluated oversampling methods and rule-generation algorithms.P.T.Noi[26] introduce

the performance comparison of Random Forest (RF), k-Nearest Neighbor (kNN), and Support Vector Machine (SVM) classifiers for land use/cover classification using Sentinel-2 image data. The research focuses on a 30 × 30 km2 area in the Red River Delta of Vietnam, covering six land use/cover types. Fourteen different training sample sizes, ranging from 50 to over 1250 pixels per class, were utilized, including both balanced and imbalanced datasets. Across all classification results, high overall accuracy (OA) was observed, ranging from 90% to 95%.Among the three classifiers and 14 sub-datasets, SVM exhibited the highest OA with the least sensitivity to training sample sizes, followed by RF and kNN. Regardless of sample size, all three classifiers achieved similar high OA values (over 93.85%) when the training sample size exceeded 750 pixels per class or approximately 0.25% of the total study area. Notably, this high accuracy was consistently attained across both imbalanced and balanced datasets. L. Almuqren[30]introduce a novel method leveraging social media mining to forecast customer churn within the telecommunications sector. Notably, it marks the pioneering use of Arabic Twitter mining for predicting churn specifically in Saudi Telecom companies. The efficacy of this newly proposed approach was validated through diverse standard metrics and a thorough comparison with ground-truth actual outcomes provided by a telecom company.

2.2 Proposed System

Customer churn, the phenomenon of customers discontinuing their subscription, poses a significant challenge for subscription-based businesses. By employing machine learning algorithms, we aim to develop predictive models that can identify customers at risk of churn, enabling proactive retention strategies to be implemented.

In the problem statement of Subscription Services ,telecommunication companies face a significant challenge in retaining their customer base due to voluntary churn, where customers switch to competitors' services. This poses a considerable cost, as acquiring new customers requires substantial marketing and sales efforts compared to retaining existing ones. Thus, understanding and mitigating customer churn is crucial for maintaining profitability and market competitiveness.The challenge lies in accurately predicting which subscribers are likely to discontinue their services and implementing effective retention strategies to mitigate churn rates. Leveraging machine learning techniques, this project aims to develop a predictive model that can identify potential churners among telecommunication

subscribers. By analyzing diverse datasets encompassing subscriber demographics, usage patterns, service subscriptions, and customer interactions, the goal is to build a robust churn prediction system. Key objectives include addressing data heterogeneity, feature selection, handling imbalanced datasets, and ensuring scalability and interpretability of the machine learning solution within the telecommunication infrastructure. Successful implementation of this predictive model will empower telecommunication companies to proactively retain subscribers, enhance customer satisfaction, and optimize marketing efforts, thereby bolstering business performance in the competitive subscription services market. This paper outlines the proposed methodology, which involves data feature selection, Support Vector Classifier. Various machine learning algorithms such as logistic regression, decision trees, random forests,(KNN)K Nearest Neighbor Classifier,AdaBoost Classifier,Gradient Boosting Classifier,Voting Classifier.We will be explored and compared for their effectiveness in churn prediction. The proposed methodology seeks to contribute to the development of robust churn prediction systems that can assist subscription services in reducing customer churn rates and improving overall customer retention strategies.

2.3 Problem Definition

Telecommunication companies experience a significant challenge due to customer churn, which refers to subscribers cancelling their subscriptions. This churn results in lost revenue and increased customer acquisition costs. Identifying customers at high risk of churning before they cancel their subscriptions is critical for maintaining a healthy customer base. However, this is a complex task due to the various factors influencing churn behavior. Develop a robust customer churn prediction model that can accurately identify subscribers with a high probability of churning.

This model will empower telecommunication companies to:

- Proactively retain at-risk customers through targeted interventions.
- Improve customer satisfaction and loyalty.
- Optimize marketing campaigns and resource allocation.
- Gain a competitive advantage by reducing churn and increasing customer lifetime value.

There are many challenges:

- **Data complexity:** Telecommunication customer data is vast and multifaceted, encompassing demographics, billing information, usage patterns, and support interactions. Extracting meaningful features and preparing clean data for model training is crucial.
- **Model interpretability:** Understanding the key factors driving churn is essential for designing effective retention strategies. The churn prediction model should provide insights beyond just churn probability.
- **Model selection and evaluation:** Choosing the most suitable machine learning algorithm for churn prediction requires careful consideration of factors like data characteristics, model complexity, and computational resources. Evaluating model performance with appropriate metrics is essential.
- **Dynamic customer behavior:** Customer needs and preferences evolve over time. The churn prediction model needs to be adaptable and continuously updated to maintain accuracy in the face of changing market conditions.

Success Criteria are the below:

- The churn prediction model achieves high accuracy, precision, recall, and F1-score in identifying customers at risk of churn.
- The model provides interpretable insights into the key drivers of churn, enabling targeted retention strategies.
- The model is scalable and can be integrated effectively into existing telecommunication infrastructure.
- The model demonstrates a positive impact on customer churn rates and contributes to increased customer lifetime value.

2.4 Goals and Objectives

To significantly reduce customer churn in subscription services by proactively identifying at-risk customers and implementing targeted retention strategies.The main goals are :

- **Reduce customer churn:** The primary goal is to significantly decrease the number of customers who cancel their subscriptions with the service. This translates to higher customer retention rates and increased revenue for the company.
- **Improve customer satisfaction and loyalty:** By proactively identifying at-risk customers, targeted interventions can be implemented to address their concerns and improve their overall experience. This fosters customer loyalty and enhances brand reputation.
- **Optimize marketing and customer service resources:** By identifying churn-prone customers, resources can be strategically allocated to those most likely to benefit from targeted marketing campaigns or personalized customer service interactions. This leads to a more efficient use of resources and potentially lower customer acquisition costs.
- **Gain a competitive advantage:** In a saturated subscription service market, effectively predicting and mitigating customer churn can give a company a significant edge over competitors.

Develop a high-performing customer churn prediction model to proactively identify subscribers at risk of cancelling their subscriptions.
This objective emphasizes the following key points:

- **Develop a robust customer churn prediction model:** This model should accurately identify customers with a high probability of churning based on historical data and customer attributes.
- **Achieve high model performance:** The churn prediction model should strive for high accuracy, precision, recall, and F1-score to ensure reliable identification of at-risk customers.
- **Enhance model interpretability:** The model should not only predict churn but also provide insights into the key factors driving churn behavior. This allows for targeted interventions based on the specific needs and concerns of at-risk customers.

- **Implement a scalable and integrated solution:** The churn prediction model should be seamlessly integrated into existing systems and processes to enable efficient and timely action on churn risk.
- **Continuously monitor and improve the model:** Customer behavior and market dynamics evolve over time. The model should be regularly monitored and updated with new data to maintain its effectiveness in identifying churn risk.

By achieving these goals and objectives, customer churn prediction can become a powerful tool for subscription service providers, leading to increased customer retention, improved customer satisfaction, and ultimately, a stronger and more profitable business.

2.5 Literature Review Summary

Year and Author	Article	Tools/Software	Technique	Source	Evalution Parameter
R. K. Peddarapu, S. Ameena, S. Yashaswini, N. Shreshta and M. PurnaSahithi [2022]	Customer Churn Prediction and Subscription Using ML	• Decision tree • Ensemble learning (Random Forest) • Logistic regression	• Support Vector Machine (SVM) • XGBoost	*6th International Conference on Electronics, Communication and Aerospace Technology*, Coimbatore, India, 2022	Employ a variety of models, such as decision trees, SVM, ensemble learning (Random Forest), logistic regression, and XGBoost, to achieve accurate predictions.
P. Bhuse, A. Gandhi, P. Meswani, R. Muni and N. Katre, [2020]	Customer Churn Prediction and Subscription Using ML	• Classification algorithms • Decision trees • Random forests • Predictive Model	• Support Vector Machine (SVM) • XGBoost	*3rd International Conference on Intelligent Sustainable Systems (ICISS)*, Thoothukudi, India, 2020	Telecommunications, predictive modeling, and the application of machine learning and deep learning techniques in understanding and predicting customer churn in the telecom industry.
A. O. Akinrotimi, R. O. Ogundokun, M. A. Mabayoje, R. A. Oyekunle and M. O. Adebiyi [2023]	Customer Churn Prediction and Subscription Using ML	• Numerical models; • Churn prediction Oversampling technique • Dimensionality reduction;	Synthetic Minority Over-sampling Technique (SMOTE).	*International Conference on Science, Engineering and Business for Sustainable Development Goals (SEB-SDG)*, Omu-Aran, Nigeria, 2023	Focus on dealing with imbalanced data, which is a common challenge in churn prediction scenarios.
P. K. Dalvi, S. K. Khandge, A. Deomore, A. Bankar and V. A. Kanade [2016]	Customer Churn Prediction and Subscription Using ML	• Logistics • Regression tree analysis • Telecommunications • Data mining • Logistic Regression • Decision Trees • CRM	• Data mining • Predictive models	*2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, Indore, India, 2016,	Customer Relationship Management (CRM) is mentioned as a relevant aspect, indicating a likely exploration of how churn prediction contributes to managing customer relationships.
O. R. Devi, S. K. Pothini, M. P. Kumari, S. V and U. N. S. Charan [2023]	Customer Churn Prediction and Subscription Using ML	• Over-the-top media services (OTT) • Training Machine learning • Gain measurement	• Over-the-top media services (OTT) • Support Vector Machine (SVM)	*2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*,Salem, India, 2023	Related to over-the-top media services, training, predictive models, gain measurement, and subscription renewal in the OTT industry.

Table 2.1: Literature Review Summary Table

CHAPTER NO - 3

METHODOLOGY

3.1 Methodology Used

Customer churn, the loss of subscribers, poses a significant challenge for subscription services. Predicting churn allows proactive intervention and development of targeted strategies to retain at-risk customers. This methodology outlines the key steps for building a customer churn prediction model in the context of telecommunication subscription services.

3.1.1 Data Acquisition and Preprocessing:

- **Identify relevant data sources:** Customer demographics, billing information, service usage patterns (call history, internet usage), customer support interactions, and contract details.
- **Address missing values:** Techniques like imputation (filling missing entries) or data deletion might be necessary depending on the data quality and quantity.
- **Feature engineering:** Create new features that might be more informative for churn prediction. Examples include average monthly spend, number of support tickets raised, and churned customer referrals.
- **Data normalization:** Scale numerical features to a common range to prevent them from unduly influencing the model.
- **Data cleaning:** Identify and handle outliers or inconsistencies within the data.

3.1.2 Model Selection and Training:

- **Choose appropriate machine learning algorithms:** Common options include Logistic Regression, Random Forests, Gradient Boosting Machines, and deep learning architectures like Recurrent Neural Networks (RNNs) for sequential data analysis.
- **Split data into training and testing sets:** A common split is 80/20, with 80% for training the model and 20% for evaluation. Consider using techniques like k-fold

cross-validation for a more robust evaluation.

- **Train the model:** The chosen algorithm is trained on the training data, learning to identify patterns associated with customer churn.
- **Hyperparameter tuning:** Adjust model parameters to optimize its performance on the training data. Techniques like grid search or randomized search can be employed.

3.1.3 Model Evaluation and Interpretation:

- Evaluate model performance on the testing set: Metrics like accuracy, precision, recall, F1-score, and AUC-ROC are used to assess the model's ability to correctly classify churning and non-churning customers.
- Interpretability: Analyze the model to understand the key factors influencing churn predictions. Techniques like feature importance scores can be used to identify the most impactful features.
- Feature ablation (removing features) can also be used to assess the contribution of specific features to the model's performance.

3.1.4 Model Deployment and Monitoring:

Integrate the model into the telecommunication service infrastructure to generate churn risk scores for existing customers.

- **Define churn risk thresholds:** Customers exceeding a certain risk score are flagged as high-risk and targeted for retention efforts.
- **Develop targeted retention strategies:** This might include personalized offers, improved customer service experiences, or loyalty programs tailored to address the specific reasons for churn identified by the model.
- **Monitor model performance over time:** Customer behavior and market dynamics can evolve. Regularly retrain the model with new data to maintain its accuracy and effectiveness.

- **Cost-benefit analysis:** Evaluate the cost of implementing and maintaining the churn prediction system against the potential benefits of reduced churn and increased customer lifetime value.

By following this methodology, telecommunication companies can leverage machine learning to build robust customer churn prediction models. This allows for proactive customer retention strategies, ultimately leading to increased customer satisfaction, loyalty, and revenue growth.

3.2 Hardware Requirements

The hardware requirements are as follows:

1. **Processor:**

   a) A multi-core CPU is essential for efficient computation, especially during tasks like training the model.

   b) A quad-core processor or higher is recommended to handle resource-intensive calculations effectively.

2. **Memory (RAM):**

   a) A minimum of 8 GB RAM is essential for managing large datasets and running deep learning algorithms smoothly.

   b) Consider upgrading to 16 GB RAM or more for improved performance, particularly when dealing with complex neural network models.

3. **Storage:**

   a) Allocate at least 100 GB of available storage space to accommodate datasets, software, and project files.

   b) Using a Solid-State Drive (SSD) instead of a Hard Disk Drive (HDD) can significantly enhance data access speed and overall system responsiveness.

4. **Graphics:**

   a) While not mandatory, a dedicated graphics card (GPU) can expedite the training of the dataset and improve visualization performance.

   b) GPUs from NVIDIA (GeForce or Quadro series) or AMD (Radeon series) are preferred for their parallel processing capabilities.

5. **Internet Connectivity:**

   a) An active internet connection is necessary for downloading datasets, libraries, documentation, and updates.

6. **Monitor:**

   a) A high-resolution monitor with a size of 22 inches or more is recommended to comfortably view code, visualizations, and dashboards.

7. **Operating System:**

   a) The project can be executed on various operating systems, including Windows, macOS, or Linux.

3.3 Software Requirements

The software requirements are as follows:

1. **Python Programming Environment:**

   a) Python is the primary programming language for this project. Ensure you have Python 3.x installed on your system.

2. **Integrated Development Environment (IDE):** a) Choose an IDE to write and run Python code. Popular options include:

   i) PyCharm

   ii)Visual Studio Code

   iii) Jupyter Notebook (for interactive coding and visualization)

   iv)Google colab

3. **Python Libraries and Packages:**

   a) Install the required libraries using pip, a Python package installer. Important libraries include:

   i) NumPy (for numerical computations)

   ii)Open CV(real-time optimized Computer Vision)

   iii) Pandas (for data manipulation)

   iv) Matplotlib and Seaborn (for data visualization)

   v) Scikit and Sklearn (for calculating mathematical value)

4. **Version Control (Optional but Recommended):**

   a) Utilize Git for version control to track changes and collaborate effectively with team members.

5. **Command Line or Terminal:**

   a) Basic command-line or terminal proficiency is useful for running scripts, managing packages, and navigating directories.

6. **Web Browsers:**

   a) Ensure you have a modern web browser (e.g., Google Chrome, Mozilla Firefox)to visualize dashboards and online documentation.

7. **Text Editor:** a) While an IDE is recommended, having a simple text editor (e.g., Notepad++, Sublime Text) is useful for viewing and editing code files.

8. **Virtual Environment (Optional but Recommended):** a) Create a virtual environment to manage project-specific libraries and dependencies, ensuring a clean and isolated development environment.

3.4 Functional Requirements

The functions of software systems are defined in functional requirements and the behavior of the system is evaluated when presented with specific inputs or conditions which may include calculations, data manipulation and processing and other specific functionality.

- Our system should be able to read the data and preprocess data.
- It should be able to analyze the data.
- It should be able to group data based on hidden patterns.
- It should be able to assign a label based on its data groups.
- It should be able to split data into train set and test set.
- It should be able to train model using train set.
- It must validate trained model using test set.
- It should be able to classify the fake and real data.

3.5 Non-Functional Requirements

Nonfunctional requirements illustrate how a system must behave and create constraints of its functionality. This type of constraints is also known as the system's quality features. Attributes such as performance, security, usability, compatibility are not the feature of the system, they are a required characteristic. They are "developing" properties that emerge from the whole arrangement and hence we can't compose a particular line of code to execute them. Any attributes required by the user are described by the specification. We must contain only those needs that are appropriate for our design. Some Non-Functional Requirements are as follows:

- Reliability
- Maintainability
- Performance
- Portability
- Scalability
- Flexibility

CHAPTER NO - 4

DESIGN FLOW AND PROCEDURE

4.1 Flow Chart

The proposed process follows the steps of loading libraries ,data collection and preparation, feature scaling and selection , data transformation, parameter and model selection, training, and machine learning model evaluation.

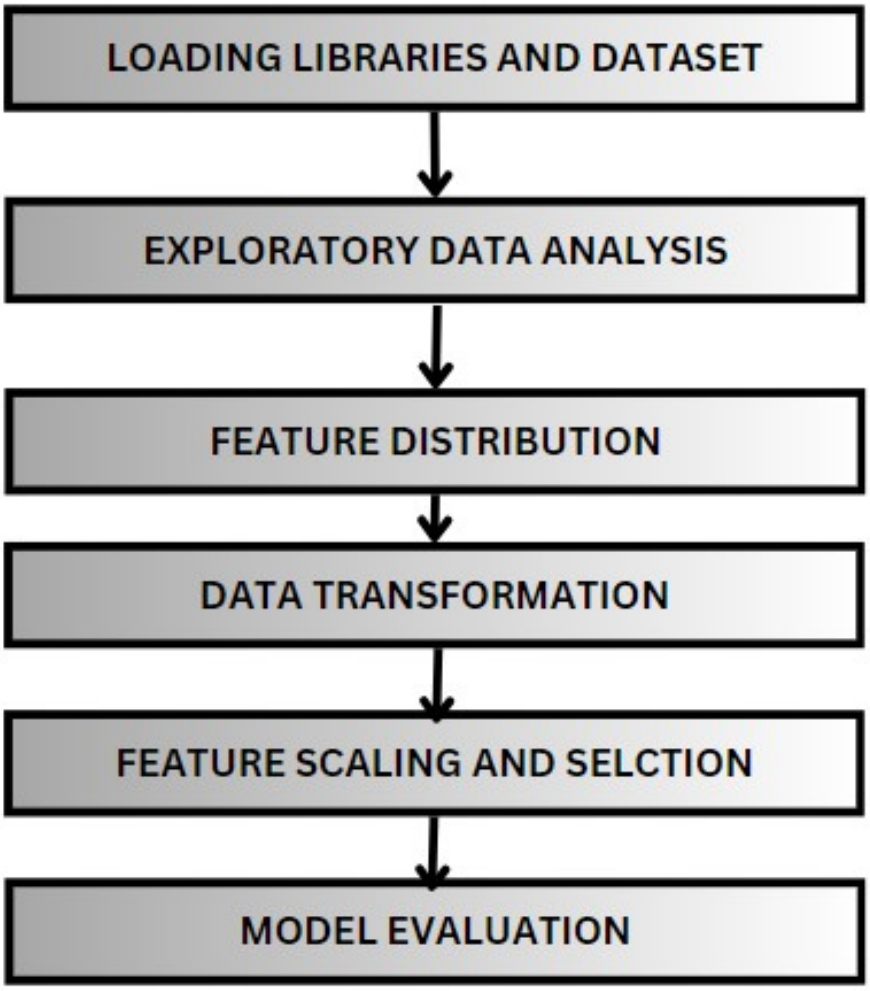

Figure 4.1: Breakdown Steps

4.2 Module Process

In a customer churn prediction project for subscription services, different modules work together in a specific process to achieve the desired outcome. The data acquisition module focuses on gathering relevant customer data from various sources within the telecommunication company.It identifies data sources like customer demographics, billing information, service usage logs (calls, internet), customer support interactions, and contract details.Data Pre-processing module handles the raw data to make it suitable for the machine learning model.It addresses missing values, creates new informative features (e.g., average monthly spend), normalizes numerical features, and cleans inconsistencies within the data.Model Selection and Training module focuses on choosing and training the machine learning model for churn prediction.It involves selecting an appropriate algorithm (Logistic Regression, Random Forest, etc.), splitting the data into training and testing sets, training the model on the training data, and hyperparameter tuning for optimal performance.Model Evaluation and Interpretation module assesses the model's effectiveness and provides insights into churn drivers. It evaluates the model's performance using metrics like accuracy, precision, recall, and F1-score. Additionally, it analyzes feature importance to understand the key factors influencing churn predictions.Model Deployment and Monitoring module integrates the model into the telecommunication system and monitors its performance over time.It integrates the model for real-time churn risk scoring of existing customers. Based on pre-defined risk thresholds, customers are flagged for potential churn. Additionally, it monitors model performance and triggers retraining with new data to maintain accuracy.

4.2.1 Importing Libraries and Loading Dataset

The first step is to import the necessary libraries at the beginning of your Python script. These libraries offer functions and classes that you'll use throughout the analysis.Python offers a rich ecosystem of libraries for data manipulation, analysis, and machine learning.The specific libraries you import will depend on the chosen tools for your churn prediction project. Here are some commonly used libraries:

- **Pandas:** A powerful library for data manipulation and analysis. It excels at working with tabular data, allowing you to load, clean, and explore your customer data.

- **NumPy:** Provides foundational structures like arrays and matrices for numerical computations. It often works hand-in-hand with pandas for data processing.
- **scikit-learn:** A popular machine learning library offering a wide range of algorithms for classification, regression, and other tasks. It provides tools for model training, evaluation, and hyperparameter tuning.
- **Matplotlib or Seaborn:** Libraries for data visualization, allowing you to create informative plots and charts to explore your data and understand churn patterns.

```
#import Libraries
import pandas as pd
import sklearn
import numpy as np
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt

%matplotlib inline
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder

from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve
from sklearn.metrics import classification_report,accuracy_score ,confusion_matrix
```

Figure 4.2: Importing Libraries

Once you import the necessary libraries, you can load your customer data into a usable format.

The format of your data source will determine the loading method:

- **CSV (Comma-Separated Values):** A common format for tabular data. pandas provides the pd.read_csv function to load CSV files into a DataFrame object, a central data structure in pandas.
- **Excel sheets:** pandas also offers pd.read_excel to import data from Excel files.

- **Databases:** Libraries like SQLalchemy can be used to connect to databases and retrieve data into pandas DataFrames.
- **APIs:** If your data resides in an API, you might need to use web scraping libraries like requests or BeautifulSoup to fetch the data and potentially parse it into a structured format (e.g., CSV) before loading it with pandas. By importing the

```
df = pd.read_csv('/home/Telecom-Customer-Churn.csv')
```

Figure 4.3: Load CSV files into a DataFrame

required libraries and loading your customer churn dataset, you lay the groundwork for further analysis and model building. This allows you to explore the data, understand customer behavior patterns, and ultimately train a model to predict churn effectively.

4.2.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a crucial initial step in any data analysis project. It's a process of investigating, summarizing, and visualizing data to understand its characteristics, identify patterns, and uncover potential relationships between variables.
Goals of EDA:

- **Gain initial insights:** EDA helps you get a feel for the data before diving into complex modeling or analysis.
- **Identify patterns and trends:** By examining the data through various visualizations, you can discover underlying patterns and trends that might inform further analysis.
- **Uncover outliers and anomalies:** EDA can help you identify data points that fall outside the expected range, which might require further investigation or cleaning.
- **Formulate hypotheses:** Based on the insights gained from EDA, you can formulate initial hypotheses about relationships between variables that can be tested through statistical analysis or modeling later.

- **Prepare data for modeling:** EDA often involves cleaning and transforming data to make it suitable for subsequent modeling steps.

```
df.columns
```

```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7039 entries, 0 to 7038
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7039 non-null   object 
 1   gender            7039 non-null   object 
 2   SeniorCitizen     7039 non-null   int64  
 3   Partner           7039 non-null   object 
 4   Dependents        7039 non-null   object 
 5   tenure            7039 non-null   int64  
 6   PhoneService      7039 non-null   object 
 7   MultipleLines     7039 non-null   object 
 8   InternetService   7039 non-null   object 
 9   OnlineSecurity    7039 non-null   object 
 10  OnlineBackup      7039 non-null   object 
 11  DeviceProtection  7039 non-null   object 
 12  TechSupport       7039 non-null   object 
 13  StreamingTV       7039 non-null   object 
 14  StreamingMovies   7039 non-null   object 
 15  Contract          7039 non-null   object 
 16  PaperlessBilling  7039 non-null   object 
 17  PaymentMethod     7039 non-null   object 
 18  MonthlyCharges    7039 non-null   float64
 19  TotalCharges      7039 non-null   object 
 20  Churn             7039 non-null   object 
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

Figure 4.4: Dataset Aanlysis

Common Techniques in EDA:

- **Descriptive statistics:** This involves calculating summary measures like mean, median, standard deviation, and frequency distributions to understand the central tendency and spread of data.

- **Visualization:** Creating various charts and graphs like histograms, scatter plots, boxplots, and heatmaps helps visualize the distribution of data, relationships between variables, and potential outliers.
- **Data cleaning:** EDA might reveal missing values, inconsistencies, or errors in the data that need to be addressed before proceeding with further analysis.
- **Feature engineering:** New features can be derived from existing ones based on domain knowledge or exploratory findings to potentially improve model performance.

Benefits of EDA:

- **Improved understanding of data:** EDA provides a foundational understanding of the data you're working with, making it easier to choose appropriate analysis techniques and interpret results effectively.
- **Informed modeling:** Insights from EDA can guide the selection of suitable machine learning algorithms and feature engineering strategies for building robust models.
- **Data quality checks:** EDA helps identify data quality issues that could negatively impact analysis results, allowing for early rectification.
- **Hypothesis generation:** EDA can spark new ideas and guide the formulation of hypotheses to be tested later in the analysis process.

Overall, EDA is an essential first step in data science projects. By dedicating time to explore and understand your data, you can lay a strong foundation for subsequent analysis and modeling, ultimately leading to more reliable and insightful results.

4.2.3 Feature Distribution

1. **Numerical Feature Distribution:** Numerical features are those that contain numerical values, such as age, weight, income, or number of purchase. We may also visualize the distribution using histograms, box plots to gain insights into the data central tendency, spread, and skewness.

   Tenure, Monthlycharges and TotalCharges are the three numerical features columns here.

```
df[numerical_features].describe()
```

	tenure	MonthlyCharges	TotalCharges
count	7039.000000	7039.000000	7028.000000
mean	32.376332	64.762963	2284.005827
std	24.561896	30.087756	2267.193201
min	0.000000	18.250000	18.800000
25%	9.000000	35.500000	401.250000
50%	29.000000	70.350000	1397.950000
75%	55.000000	89.850000	3796.912500
max	72.000000	118.750000	8684.800000

Figure 4.5: Numerical Value

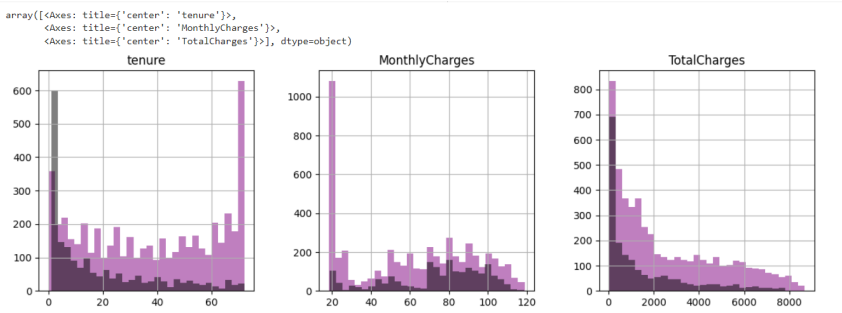

Figure 4.6: Comparison of NotChurned(purple) and Churned(black)

2. **Categorical Feature Distribution:** It represents discrete categories or groups, such as gender, product categories, or customer segments. Categorical features in this dataset are gender, SeniorCitizen, Partner, Dependents, PhoneService, MultipleLines, InternetService,online security, OnlineBackup, DeviceProtection,TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod,To analyze the category distribution, we typically calculate frequencies or proportions for each category and visualize them using bar charts or pie charts.

   Here we ae taking contract columns to analyze not churn and churned through month-to-month, two-year, and one-year parameters. Below the bar graph represents for contract feature.

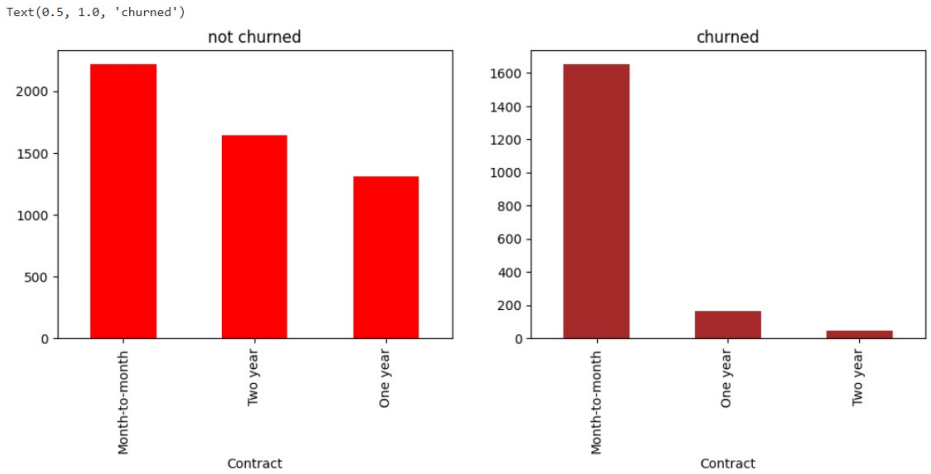

Figure 4.7: Contract Feature(NotChurned= red, Churned= brown)

3. **Target Variable Distribution:** The distribution of the target variable refers to how the values of the variable are spread out or distributed across different categories or values. In many machine learning or statistical modeling tasks, the target variable is the variable that we are trying to predict or understand. Target variable distribution shows that we are dealing with an imbalanced problem as there are many more non-churned as compare to churned users.The target feature is churn here.

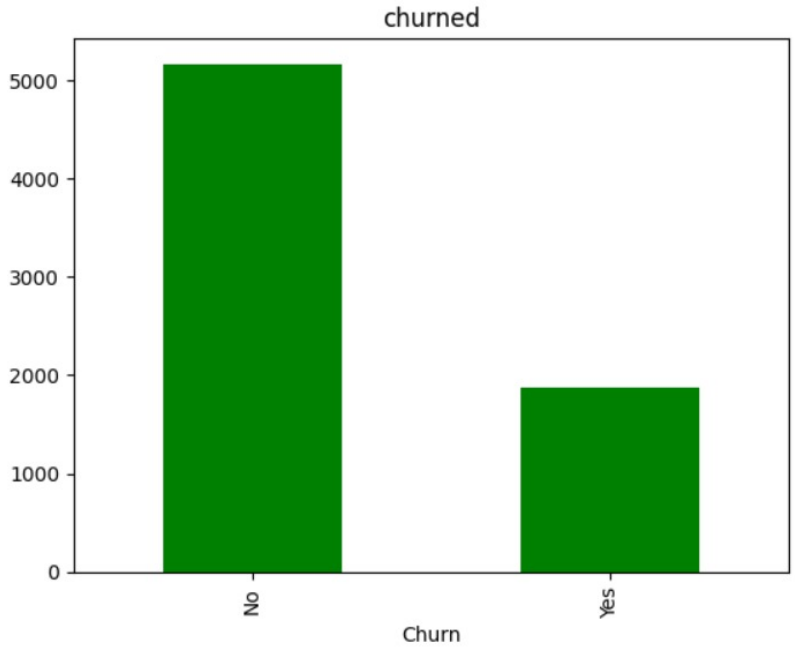

Figure 4.8: Target Variable Distribution

4.2.4 Outliers using IQR method

The IQR (Inter Quartile Range) method is a statistical technique used to identify and handle outliers in a dataset. It involves calculating the difference between the third quartile (Q3) and the first quartile (Q1) of the dataset. The IQR is then used to detect outliers by defining a range around the median within which most of the data points lie. Lower Limiting Point (LLP):

```
x = ['tenure','MonthlyCharges']
def count_outliers(data,col):
        q1 = data[col].quantile(0.25,interpolation='nearest')
        q2 = data[col].quantile(0.5,interpolation='nearest')
        q3 = data[col].quantile(0.75,interpolation='nearest')
        q4 = data[col].quantile(1,interpolation='nearest')
        IQR = q3 -q1
        global LLP
        global ULP
        LLP = q1 - 1.5*IQR
        ULP = q3 + 1.5*IQR
        if data[col].min() > LLP and data[col].max() < ULP:
            print("No outliers in",i)
        else:
            print("There are outliers in",i)
            x = data[data[col]<LLP][col].size
            y = data[data[col]>ULP][col].size
            a.append(i)
            print('Count of outliers are:',x+y)
global a
a = []
for i in x:
    count_outliers(df,i)
```

```
No outliers in tenure
No outliers in MonthlyCharges
```

Figure 4.9: Outlier Analysis

$$\text{LLP} = Q1 - 1.5 \times \text{IQR}$$

Upper Limiting Point (ULP):

$$\text{ULP} = Q3 + 1.5 \times \text{IQR}$$

4.2.5 Data Cleaning and Transforming

Identify and rectify errors, inconsistencies, and missing data within a dataset to ensure its accuracy and reliability.This could involve techniques like imputation (filling in missing entries) or data deletion depending on the data and its importance. This might include typos, formatting errors, or logical inconsistencies within the data.Extreme values that deviate significantly from the rest of the data can be addressed through techniques like capping or winsorizing (replacing outliers with specific values). Identify and eliminate duplicate data entries that can skew analysis results.Here we don't need the customerID column. So, we will remove it from the dataset

```
df.drop(['customerID'],axis = 1)
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService
0	Female	0	Yes	No	1	No
1	Male	0	No	No	34	Yes
2	Male	0	No	No	2	Yes
3	Male	0	No	No	45	No
4	Female	0	No	No	2	Yes

Figure 4.10: Data Cleaning

Convert data from its raw format into a format more suitable for analysis or modeling. This often involves manipulating, restructuring, and combining data to extract meaningful insights.Creating new features from existing ones that might be more informative for your specific task.For example, in customer churn prediction, you might create a feature for "average monthly spend" from billing data.Scaling numerical features to a common range prevents features with larger scales from dominating the analysis. Techniques like min-max scaling or z-score normalization are commonly used.

Encoding categorical variables is the process of transforming categorical data (like customer types) into numerical representations suitable for machine learning algorithms. Techniques like one-hot encoding or label encoding can be employed.

Data aggregation is combining data points to create summaries or higher-level insights. For

instance, you might aggregate call history data to calculate "total monthly call duration" per customer.

One-hot Encoding: One Hot Coding is a technique used in machine learning and data preprocessing, particularly for categorical variables. It converts categorical variables into a binary vector representation, where each category is represented by a binary value in a vector. Dummy variables are used to represent categorical data in a format that can be used for machine learning algorithms, which typically require numerical input. They are especially common in regression analysis, where categorical predictors need to be incorporated into the model.

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges	gender_Female	gender_Male	Partner_No	Partner_Yes	Dependents_No	Dependents_Yes
0	0	1	29.85	29.85	1	0	0	1	1	0
1	0	34	56.95	1889.50	0	1	1	0	1	0
2	0	2	53.85	108.15	0	1	1	0	1	0
3	0	45	42.30	1840.75	0	1	1	0	1	0
4	0	2	70.70	151.65	1	0	1	0	1	0

5 rows × 47 columns

Figure 4.11: One hot encoding

Rearranging Columns: Rearranging columns refers to changing the order in which columns appear in a table or spreadsheet. Organizing columns based on logical flow or category can make the data easier to understand and analyze. Placing related columns next to each other allows for quicker visual comparisons and identification of patterns.Specific column arrangements might be more suitable for creating charts and graphs that effectively represent the data.

Feature Scaling: Feature scaling, also known as data normalization or standardization, is a crucial data preprocessing technique used in machine learning. Its primary purpose is to transform the features within a dataset to a common range. This ensures that all features contribute equally to the model's learning process and avoids situations where features with larger magnitudes dominate those with smaller values.StandardScaler is a technique used in machine learning for data preprocessing, specifically for features that are measured on different scales. It addresses the issue of features having varying ranges or units, which can negatively impact the performance of machine learning algorithms.

Feature Selection: Feature selection is identifying and selecting the most relevant features from the original dataset. This process involves reducing the dimensionality of the feature space by discarding irrelevant or redundant features, which can lead to

improved model performance, reduced computational complexity, and enhanced interpretability. The code you provided imports a function called train_test_split from the sklearn.model_selection library in Python. This function is commonly used for splitting data into training and testing sets for machine learning tasks.

```
from sklearn.model_selection import train_test_split
X = scaled_features
Y = df1['Churn_Yes']
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size = 0.3,random_state=46)
```

Figure 4.12: Feature Selection

- from sklearn.model_selection import train_test_split: This line imports the train_test_split function from the model_selection submodule of the scikit-learn library.
- X = scaled_features: This line assumes you have already prepared your data for machine learning. Here, scaled_features likely represents your features (independent variables) that have been preprocessed and scaled (e.g., using standard scaling).
- Y = df1['Churn_Yes']: This line assumes you have a pandas dataframe named df1. It extracts the column named "Churn_Yes" from this dataframe and assigns it to the variable Y. This column likely contains the target variable (dependent variable) indicating whether a customer churned (e.g., "Yes") or not.
- X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=46): This is the most important line where the actual splitting happens. Here's what it does:
- train_test_split(X, Y, ...): This calls the train_test_split function with the following arguments:
- X: This is the data you want to split, which in this case is your features (scaled_features).
- Y: These are the labels or target variables (df1['Churn_Yes']).
- test_size=0.3: This argument specifies the proportion of the data to be included in the test set. Here, you've chosen a 30% split, meaning 30% of the data will be

allocated to the test set and the remaining 70% will be used for training. You can adjust this value based on your needs (common splits are 80/20 or 70/30).

- random_state=46: This argument sets the random seed for splitting the data. This ensures reproducibility if you run the code multiple times. If you set this to a specific value (like 46 in this case), you'll get the same split every time you run the code. This can be useful for debugging or comparing different models. However, for final model evaluation, it's generally recommended to not fix the random state to mimic real-world scenarios where data isn't guaranteed to follow a specific pattern.

The output:
After running this line, we'll have four new variables:

- X_train: This contains the training data for your features.
- X_test: This contains the testing data for your features.
- Y_train: This contains the training data for your target variable (churn labels).
- Y_test: This contains the testing data for your target variable.

By splitting the data into training and testing sets, you can train your machine learning model on the training data (X_train, Y_train) and then evaluate its performance on unseen data from the testing set (X_test, Y_test). This helps to assess how well the model generalizes to new data and avoids overfitting to the training data.

4.2.6 Machine Learning Model Evaluations

Once a model is trained on historical data, its performance is assessed on unseen data (testing set) using various metrics. These metrics evaluate the model's ability to correctly classify churning and non-churning customers.

- Accuracy: Measures the overall percentage of correct predictions (churn or no churn).
- Precision: Indicates the proportion of predicted churn cases that are truly churning customers.
- Recall: Measures the proportion of actual churning customers that the model correctly identifies.

- F1-score: Balances precision and recall, providing a more comprehensive view of model performance.
- AUC-ROC: For churn prediction models, the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) measures the model's ability to distinguish between churning and non-churning customers.

Interpretability: Understanding the factors driving churn predictions is crucial for crafting effective retention strategies. Techniques like feature importance scores highlight which features in the model have the strongest influence on churn prediction.

Model Selection: Choosing the best ML algorithm for churn prediction involves evaluating different options like Logistic Regression, Random Forests, KNN, Decision Tree, Support Vector Machine,AdaBoost Classifier and Gradient Boosting Machines.The selection process considers factors like data characteristics, model complexity, and computational resources.

Overall, machine learning models, when evaluated and interpreted effectively, can provide valuable predictions for customer churn in subscription services. These predictions empower businesses to implement proactive retention strategies, ultimately leading to increased customer lifetime value and reduced churn.

CHAPTER NO - 5

ALGORITHM DESCRIPTION

Analyzing machine learning algorithms is essential for understanding their behavior, efficiency, and suitability for different tasks. This analysis typi cally involves assessing factors such as computational complexity, training and prediction times, scalability, model complexity, and generalization performance.

5.1 Logistics Regression

Logistic regression is a fundamental statistical method widely used in machine learning for classification tasks. It's particularly well-suited for problems where the outcome variable (dependent variable) is categorical, typically binary (yes/no, churn/not churn, subscribed/unsubscribed).

It doesn't directly predict the class label (churn/not churn) but rather models the probability of an event occurring, given a set of features (independent variables) that describe a data point. For example, it might estimate the probability of a customer churning based on their monthly bill amount, call duration, and number of support tickets.

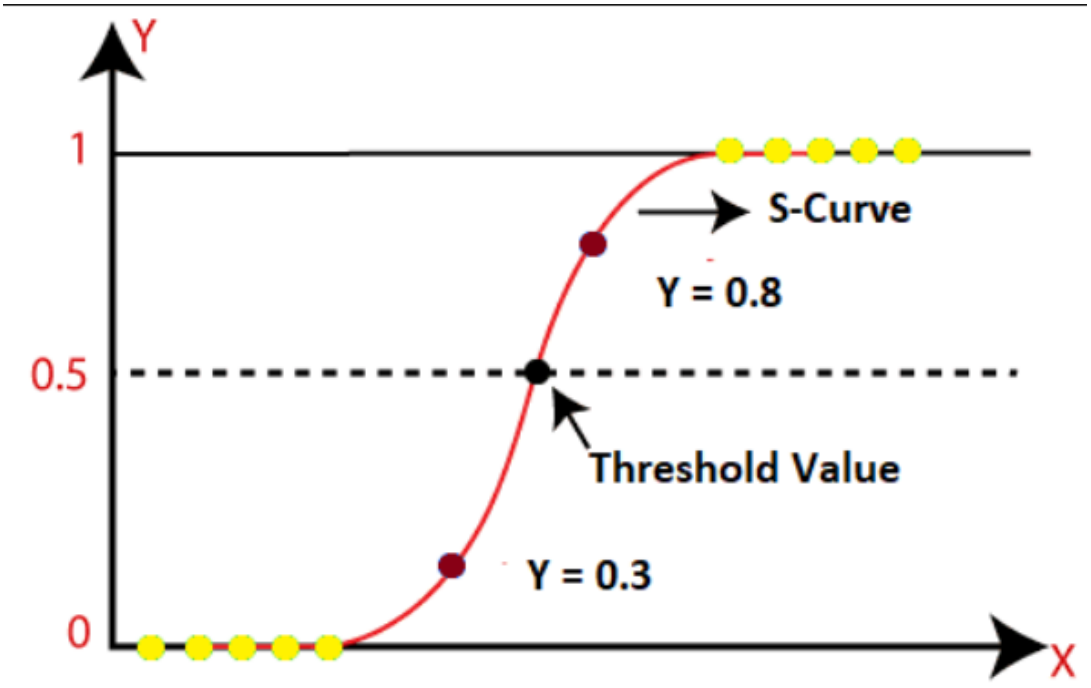

Figure 5.1: Sigmoid function

The Math Behind it is that logistic regression employs a mathematical function called the sigmoid function (also known as the logistic function) to map the linear combination of features (obtained through a linear regression model) to a probability value between 0 and 1.
A value closer to 1 signifies a higher probability of the event (customer churn), while a value closer to 0 indicates a lower probability.

5.1.1 Advantages of Logistic Regression:

- **Interpretability:** One of the key strengths of logistic regression is its interpretability. The coefficients associated with each feature in the model provide insights into how those features influence the predicted probability. This is crucial for understanding the factors driving churn and tailoring retention strategies accordingly.

- **Simplicity:** Logistic regression is a relatively simple algorithm compared to other machine learning models. This makes it easier to understand, implement, and interpret results.

- **Computational Efficiency:** Training a logistic regression model is computationally efficient, making it suitable for large datasets.

5.1.2 Disadvantages of Logistic Regression:

- **Limited to Binary Classification:** The standard logistic regression model is designed for binary classification problems. For multi-class problems (more than two categories), extensions like multinomial logistic regression can be employed, but they might be less interpretable.

- **Non-linear Relationships:** Logistic regression assumes a linear relationship between features and the log odds of the event. If the relationships are highly non-linear, the model might not capture them accurately. In such cases, other machine learning algorithms like decision trees or neural networks might be more suitable.

Logistic regression is a popular choice for customer churn prediction in subscription services. By analyzing customer data like demographics, billing information, and service

usage patterns, it can estimate the probability of churn for each customer. This allows companies to prioritize retention efforts for customers at high risk of churning.

In conclusion, logistic regression is a powerful tool for classification tasks, particularly valuable for its interpretability and efficiency. While it has limitations in handling complex non-linear relationships, its strengths often make it a good starting point for customer churn prediction and other binary classification problems.

5.2 Support Vector Machine(SVM)

Support Vector Machines (SVMs) are a powerful type of supervised machine learning algorithm known for their effectiveness in various classification tasks, including customer churn prediction.
Here's a breakdown of how SVMs work:

- SVMs aim to find an optimal hyperplane in high-dimensional space that separates data points belonging to different classes with the maximum margin.
- A hyperplane is a generalization of a line in two dimensions. For example, in 2D space, a hyperplane would be a line, while in 3D space, it would be a plane.
- The margin refers to the distance between the hyperplane and the closest data points from each class, called support vectors.

5.2.1 Key Concepts:

- **Feature Space:** Data points are often represented in a high-dimensional space using features (characteristics) relevant to the classification task.
- **Classification:** The goal is to find a hyperplane that effectively separates the data points into their respective classes (e.g., churning vs. non-churning customers).
- **Support Vectors:** These are the data points closest to the hyperplane, essentially defining the margin. They are crucial for the model's decision boundary.
- **Maximizing Margin:** A wider margin between the hyperplane and the support vectors translates to a more robust classification model, less susceptible to errors on unseen data.

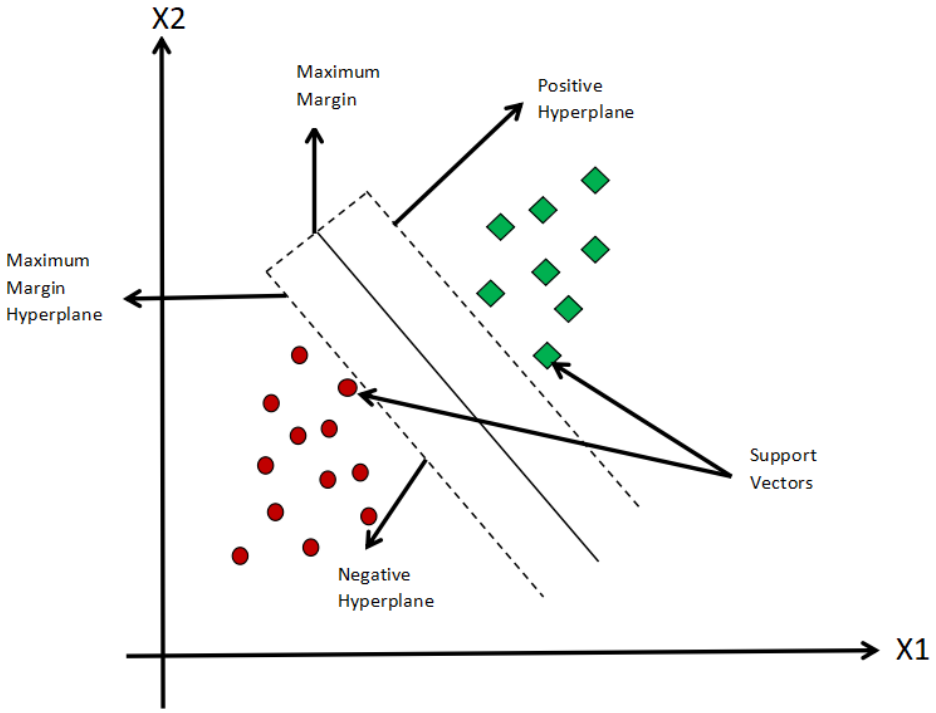

Figure 5.2: SVM Algorithm

5.2.2 Advantages of SVMs:

- **Effective in high-dimensional spaces:** SVMs can handle complex datasets with many features.
- **Good performance with limited data:** SVMs can be efficient even with relatively small datasets compared to other algorithms.
- **Interpretability:** SVMs provide insights into the features that most influence the classification, aiding in understanding the model's decision-making process.

5.2.3 Disadvantages of SVMs:

- **Computationally expensive:** Training SVMs can be computationally intensive for very large datasets.
- Not ideal for regression problems: SVMs are primarily designed for classification tasks.
- **Tuning hyperparameters:** Choosing the right hyperparameters for the SVM kernel (a function that transforms data into a suitable space) can be crucial for

optimal performance.

5.2.4 Applications of SVMs:

- **Customer churn prediction:** Identifying customers at risk of canceling their subscriptions.
- **Image classification:** Recognizing objects and scenes in images.
- **Text categorization:** Classifying text documents into different categories.
- **Spam detection:** Filtering out spam emails.

In conclusion, SVMs are a versatile machine learning tool for classification tasks. Their ability to handle high-dimensional data, perform well with limited data, and offer some interpretability makes them a valuable choice for various applications.

5.3 Decision Tree Classifier

Decision trees are a type of machine learning algorithm well-suited for classification tasks, making them a popular choice for customer churn prediction in subscription services.It can be used in conjunction with other machine learning algorithms to create ensemble models that leverage the strengths of different approaches.

5.3.1 Building the Tree:

- **Root Node:** The decision tree starts with the entire customer dataset as the root node.
- **Splitting:** The algorithm identifies the most informative feature (e.g., monthly spend, number of service calls) to split the data into two branches. This feature is chosen based on a measure like information gain, which indicates how much the feature reduces uncertainty about churn.
- **Branching:** Each branch represents a possible value of the chosen feature (e.g., high or low monthly spend). Customers with that value are directed to the corresponding branch.

- **Recursive Splitting:** This process of splitting based on the most informative feature continues at each new node, creating a tree-like structure.
- **Leaf Nodes:** The branches eventually reach leaf nodes, representing terminal points where the data points are classified as churners or non-churners. Customers satisfying the conditions along a specific path through the tree reach the corresponding leaf node with its churn prediction.

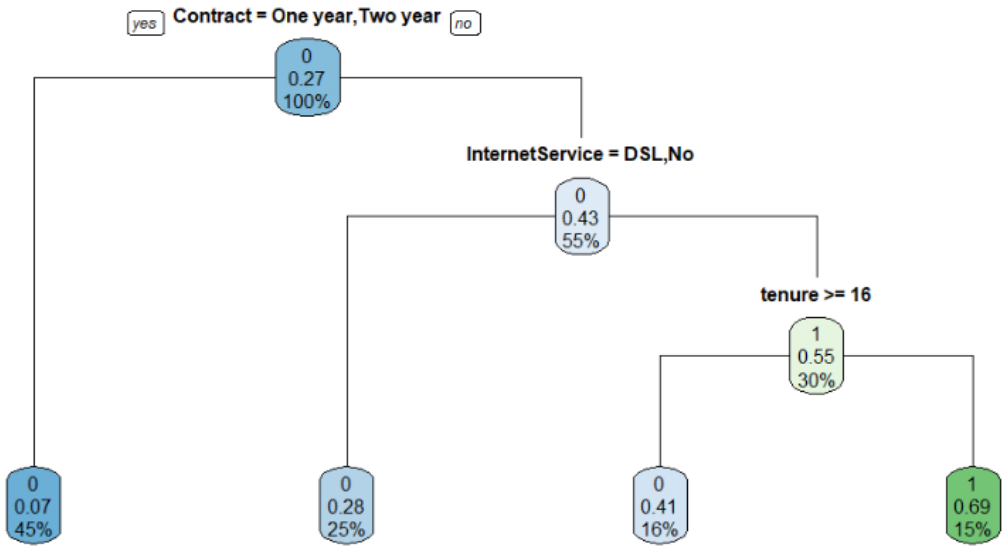

Figure 5.3: Decision tree in subscription services

5.3.2 Advantages for Subscription Services:

- **Interpretability:** Unlike some complex models, decision trees are easily interpretable. By following the tree's branches and splitting criteria, you can understand which customer characteristics are most influential in predicting churn. This allows for targeted retention strategies based on the specific reasons for churn identified by the model.
- **Visualization:** Decision trees can be visualized graphically, making it easier to understand the decision-making process and identify key features impacting churn.
- **Handling Categorical Data:** Decision trees can handle both numerical and categorical data (e.g., customer demographics, service plans) effectively.

5.3.3 Considerations for Subscription Services:

- **Overfitting:** Decision trees can be susceptible to overfitting, where the model performs well on the training data but poorly on unseen data. Techniques like pruning (removing unnecessary branches) or setting minimum data points per split can be used to mitigate this.
- **Feature Selection:** Choosing the most relevant features is crucial for accurate predictions. Feature engineering techniques might be necessary to create meaningful features for the decision tree.
- **Complexity:** Very large datasets can lead to complex decision trees with many branches, potentially hindering interpretability and increasing training time.

5.3.4 Churn Prediction with a Decision Tree:

Subscription service uses a decision tree to predict churn based on two features: monthly call duration (high/low) and number of support tickets (high/low).

- The root node represents all customers.
- The first split might be based on call duration (high or low).
- For customers with high call duration, the model might analyze the number of support tickets (high or low).
- Customers with high call duration and low support tickets might be classified as low churn risk (leaf node).
- On the other hand, customers with high call duration and high support tickets might be classified as high churn risk (another leaf node).
- Similarly, the tree would analyze customers with low call duration and make predictions based on the number of support tickets.
- Customers with low viewing time might be directed towards a branch with a higher churn risk.
- If a customer's preferred genre has limited new releases, the model might predict a higher churn risk.

- Frequent logins could indicate a low churn risk, while infrequent logins might lead to a higher churn risk prediction.

Overall, decision trees can be a valuable tool for customer churn prediction in subscription services. Their interpretability allows for targeted retention strategies, while their ability to handle various data types makes them suitable for analyzing customer data in subscription services.

5.4 K-Nearest Neighbour

K-Nearest Neighbors (KNN) is a machine learning algorithm applicable to customer churn prediction in subscription services. It's a non-parametric, supervised learning technique that classifies data points based on the similarity to their closest neighbors.

5.4.1 KNN Explaination:

- **Training:** The KNN algorithm doesn't explicitly learn a model from the training data. Instead, it stores all the training data points.
- **Prediction:** When a new customer's data (features like usage patterns, billing history) needs to be classified (churner or non-churner), KNN finds the k closest neighbors in the training data based on a chosen distance metric (e.g., Euclidean distance).
- **Classification:** The majority class (churner or non-churner) among the k nearest neighbors determines the predicted class for the new customer.

5.4.2 KNN for Churn Prediction:

In customer churn prediction, KNN can be used to classify new customers as likely to churn or not churn based on their similarity to past churners and non-churners in the training data.

- **Features:** Customer data points represent features like monthly spend, service usage patterns (call duration, internet usage), and customer support interactions.
- **Distance Metric:** A distance metric like Euclidean distance is used to measure the similarity between a new customer and existing customers in the training data.

- **K Value:** The 'k' in KNN refers to the number of nearest neighbors considered for the prediction. Choosing the optimal k value is crucial for model performance.

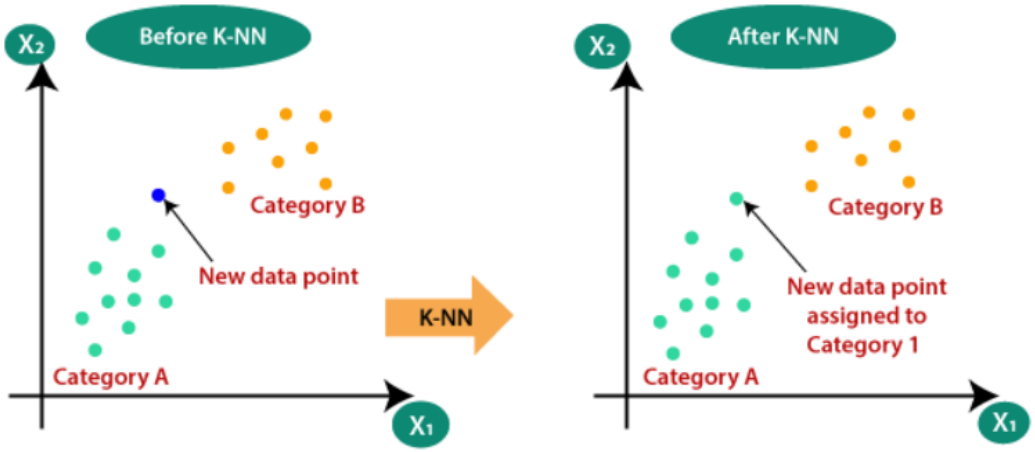

Figure 5.4: Categorization in KNN

5.4.3 Advantages of KNN:

- **Simple and interpretable:** KNN is easy to understand and implement. The classification is based on the nearest neighbors, offering some level of interpretability.
- **No explicit model training:** KNN doesn't require complex model training procedures, making it computationally efficient for smaller datasets.
- **Works well with high-dimensional data:** KNN can handle datasets with many features, which can be common in subscription services.

5.4.4 Disadvantages of KNN:

- **Curse of dimensionality:** KNN can suffer from the "curse of dimensionality" in high-dimensional data. Irrelevant features can skew the distance calculation, affecting accuracy.
- **Sensitive to k value:** Choosing the optimal k value is crucial and can impact performance significantly. Techniques like cross-validation can be used to find the best k.

- **Computational cost for prediction:** While training is fast, predicting for new data points can be computationally expensive as it involves distance calculations with all training data points.

Overall, KNN can be a viable option for customer churn prediction in subscription services, particularly for smaller datasets and situations where interpretability of the results is desired. However, it's important to be aware of its limitations and consider other machine learning algorithms like decision trees, random forests, or gradient boosting machines that might outperform KNN in certain scenarios.

5.5 Random Forest

Random forests are a powerful machine learning technique particularly well-suited for customer churn prediction in subscription services.It combines the predictions of multiple decision trees.It reduce data set overfitting and increase accuracy. Imagine a forest (the model) composed of many individual trees (decision trees). Each tree makes a prediction based on a subset of features randomly selected from the entire data set. The final prediction of the random forest is the majority vote (or average) of the predictions from all the individual trees.

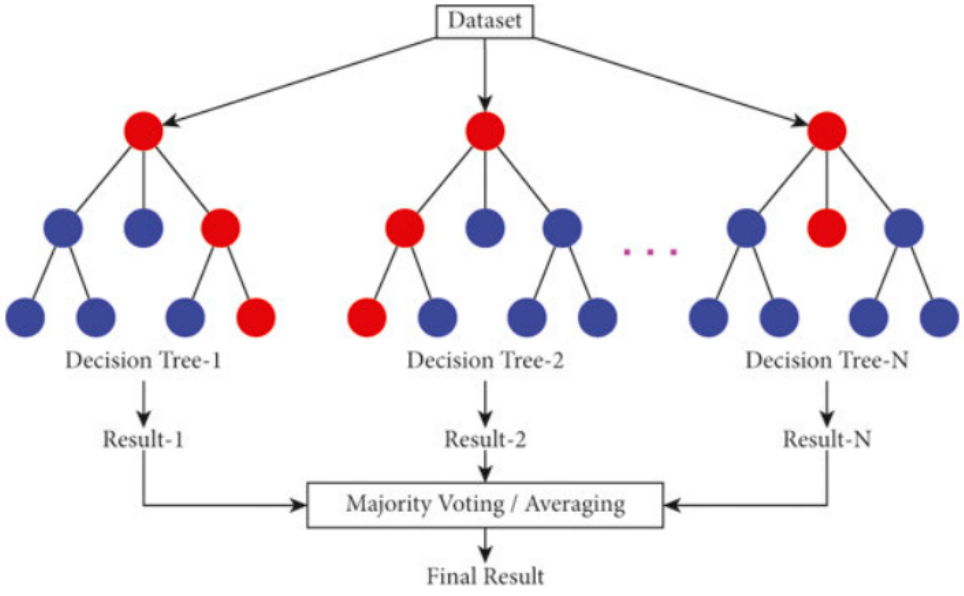

Figure 5.5: Random Forest Tree

5.5.1 Benefits of Random Forests for Churn Prediction:

- **Improved Accuracy:** By aggregating the predictions of multiple trees, random forests can often achieve higher accuracy than a single decision tree.
- **Reduced Overfitting:** Random forests are less prone to overfitting the training data compared to single decision trees. This is because each tree learns from a different subset of features, reducing the model's reliance on specific patterns that might not generalize well to unseen data.
- **Handling Imbalanced Datasets:** In churn prediction, churners might be a minority compared to non-churning customers. Random forests can effectively handle imbalanced datasets by inherently reducing the variance of the model's predictions.
- **Feature Importance:** Random forests provide insights into the relative importance of features in churn prediction. This helps subscription services understand the key factors driving customer churn and prioritize retention strategies accordingly.

5.5.2 Random Forests in Subscription Services:

- **Data Preparation:** Customer data, including demographics, billing information, service usage patterns, and support interactions, is collected and preprocessed for the model.
- **Building the Forest:** The algorithm randomly selects a subset of features (e.g., call duration, number of support tickets) at each node of each decision tree. The tree is then grown based on these features, splitting the data into subgroups based on decision rules (e.g., call duration $< +$ 10 minutes).
- **Prediction:** Each individual decision tree predicts whether a customer is likely to churn or not.
- **Final Prediction:** The final churn prediction for a customer is determined by majority vote (or average) of the predictions from all the trees in the forest.

In a telecommunication company, a random forest model might analyze features like:

- Average monthly call duration
- Frequency of internet usage
- Number of support tickets raised
- Contract type (monthly vs. yearly)

The model can then identify patterns and thresholds within these features that are more likely to indicate customer churn. For instance, the model might reveal that customers with consistently low call duration and infrequent internet usage have a higher churn risk. This information can be used to:

- Design targeted retention offers like bonus data packages for low-usage customers.
- Proactively address potential issues by offering troubleshooting assistance to customers with a history of raising support tickets.

Overall, random forests offer a robust and interpretable approach to customer churn prediction in subscription services. By leveraging the collective intelligence of multiple decision trees, random forests provide valuable insights into customer behavior and churn drivers, empowering businesses to implement effective retention strategies and improve customer lifetime value.

5.6 Gradient Boosting Classifier

Gradient boosting classifiers are a powerful machine learning technique well-suited for customer churn prediction in subscription services. It requires careful parameter tuning to optimize their performance. These parameters control the number of trees in the ensemble, their complexity, and how they are weighted in the final prediction.
While powerful, gradient-boosting models can become complex and require more computational resources compared to simpler models.
It's important to balance model complexity with interpretability. While gradient boosting offers feature importance scores, understanding the specific logic behind each decision tree can be challenging.
Imagine a team of experts making predictions about customer churn, each learning from

the mistakes of the previous one. Gradient boosting classifiers mimic this approach by sequentially building an ensemble of weak learners (typically decision trees) that iteratively improve upon each other.

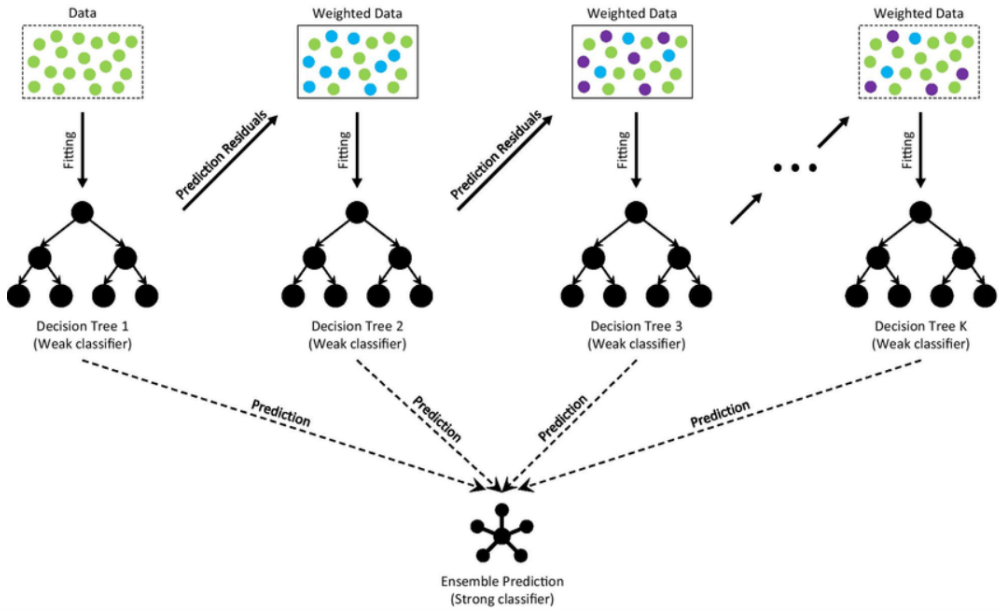

Figure 5.6: Gradient Boosting Decision Tree

5.6.1 Step-by-Step Breakdown:

- **Initial Model:** Start with a simple model, like a shallow decision tree. This model predicts churn probability for each customer based on their features (e.g., call history, service usage).
- **Calculate Errors:** Analyze the initial model's predictions compared to actual churn data. Identify the data points where the model made mistakes (churners it missed or non-churners it predicted to churn).
- **Boosting:** Build a second decision tree focused on correcting the errors of the first model. This new tree prioritizes the data points where the first model struggled.
- **Combine Predictions:** Add the predictions from both models. The final churn probability for a customer is a weighted sum of the individual predictions from each tree in the ensemble.

- **Repeat and Improve:** Repeat steps 2-4 multiple times, building a series of decision trees that progressively improve on the overall accuracy by focusing on the most challenging predictions.

5.6.2 Benefits for Churn Prediction:

- **Handles Complex Relationships:** Gradient boosting can capture complex, non-linear relationships between customer features and churn, unlike simpler models like logistic regression.
- **Feature Importance:** These models provide insights into which features (e.g., high data usage after a discount period) are most influential in predicting churn. This is crucial for designing targeted retention strategies.
- **Robustness:** Ensemble methods like gradient boosting are generally less prone to overfitting compared to single decision trees.

Example: Churn Prediction in Telecom

- A telecom company uses a gradient boosting classifier to predict churn. The model identifies features like low call duration, infrequent internet usage, and past history of service changes as significant predictors of churn.
- The model helps the company understand that customers who make short calls and rarely use internet are more likely to churn. Additionally, customers who frequently switch services might be dissatisfied and at risk of churning.

Based on these insights, the telecom company can:

- Offer bundled plans combining voice and data to cater to low-usage customers.
- Provide targeted promotions encouraging increased internet usage.
- Implement proactive outreach to customers considering service changes, addressing their concerns and offering loyalty incentives.

Overall, gradient boosting classifiers offer a powerful tool for customer churn prediction in subscription services. Their ability to handle complex relationships and provide feature importance makes them well-suited for identifying the key drivers of churn and developing effective retention strategies.

5.7 AdaBoost Classifier

AdaBoost (Adaptive Boosting) is a powerful machine learning algorithm well-suited for customer churn prediction in subscription services. It is an ensemble learning method, meaning it combines multiple weak learners (models) into a single, stronger learner.Weak learners are typically simple models like decision trees with limited predictive power on their own.AdaBoost iteratively trains these weak learners, focusing on data points that the previous learners struggled with.

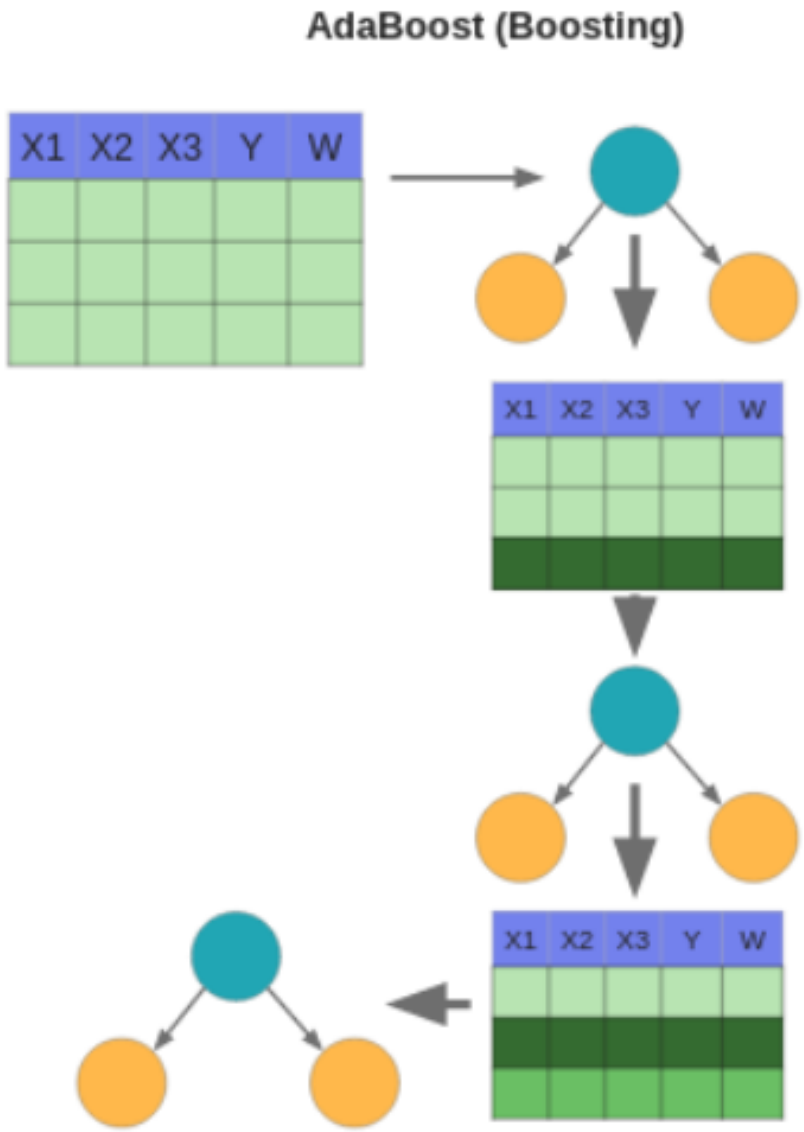

Figure 5.7: AdaBoost Classifier

5.7.1 Boosting Process:

- **Initial Training:** Train the first weak learner on the entire dataset.
- **Weighting:** Assign higher weights to data points that the first learner misclassified. This makes the next learner focus on these "harder" examples.
- **Subsequent Learners:** Train subsequent weak learners on the weighted data, prioritizing the misclassified instances from the previous learning stage.
- **Final Prediction:** Combine the predictions of all weak learners using a weighted voting scheme, giving more weight to learners with better performance on the training data.

5.7.2 Benefits for Churn Prediction:

- **Improved Accuracy:** AdaBoost can often achieve higher accuracy than individual weak learners, particularly in scenarios with complex churn patterns.
- **Handles Imbalanced Data:** Churners might be a minority in subscription services. AdaBoost's focus on misclassified points can be beneficial in such imbalanced datasets.
- **Interpretability:** While not as interpretable as some other algorithms, AdaBoost often relies on decision trees as weak learners. Analyzing these trees can provide insights into the features most influential in churn predictions.

5.7.3 Implementation in Subscription Services:

- **Data Preparation:** Prepare customer data with features relevant to churn (usage patterns, demographics, billing details).
- **Weak Learner Choice:** Decision trees are common choices, but other options like linear models can also be explored.
- **Model Training:** Implement the AdaBoost algorithm, iteratively training weak learners and combining their predictions.

- **Evaluation:** Assess the model's performance using metrics like accuracy, precision, recall, and F1-score on a separate testing set.
- **Interpretation:** Analyze the decision trees within the AdaBoost model to understand the key drivers of churn predictions.

A mobile network provider might use AdaBoost to predict churn. The model could identify features like:

- **Low data usage:** Customers who rarely use their data plans might be more likely to churn.
- **Infrequent top-ups:** Customers who don't top up their accounts regularly might be considering switching providers.
- **High number of dropped calls:** Customers experiencing frequent call drops might be dissatisfied with the service quality.

By understanding these factors, the mobile network provider can:

- Offer targeted data bundles for low-usage customers.
- Implement automated top-up reminders or introduce loyalty programs rewarding frequent top-ups.
- Invest in network infrastructure improvements to reduce dropped calls and improve customer satisfaction.

In conclusion, AdaBoost can be a valuable tool for subscription services looking to leverage machine learning for customer churn prediction. Its ability to improve accuracy, handle imbalanced data, and offer some level of interpretability makes it a strong contender for building robust churn prediction models.

5.8 Voting Classifier

A voting classifier, also known as an ensemble classifier, is a machine learning technique that combines the predictions of multiple individual classifiers to make a final prediction. It leverages the wisdom of the crowd, aggregating the opinions of multiple models to potentially achieve better predictive performance than any single model on its own.

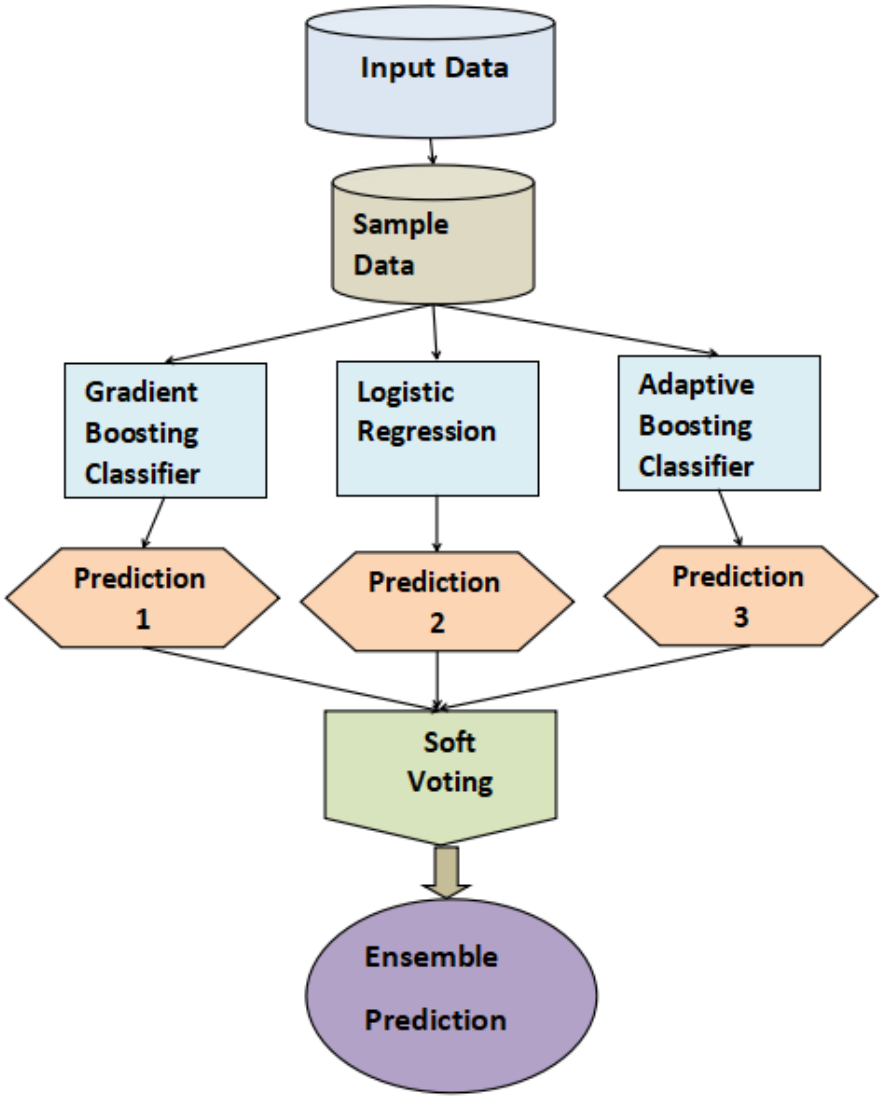

Figure 5.8: Soft voting Classifier

5.8.1 Working Steps

Individual Classifiers: A voting classifier consists of multiple individual base classifiers, each trained on the same dataset but potentially using different algorithms or configurations. These base classifiers could be decision trees, support vector machines, logistic regression models, etc.

Voting Strategy: When it comes time to make predictions, each classifier in the ensemble provides its prediction for the input data point. There are several strategies for combining these individual predictions:

1. **Hard Voting:** In hard voting, each classifier's prediction is considered as a 'vote',

and the majority vote becomes the final prediction. For example, if there are three classifiers and two of them predict class A while one predicts class B, the final prediction will be class A.

2. **Soft Voting:** In soft voting, instead of just considering the majority vote, the classifiers provide probability estimates for each class. The final prediction is determined by averaging these probabilities across all classifiers, and the class with the highest average probability is chosen. Soft voting often works better than hard voting as it considers the confidence levels of the individual classifiers.

3. **Weighted Voting:** In weighted voting, each classifier's prediction is assigned a weight, and the final prediction is a weighted combination of the individual predictions. This allows you to give more importance to certain classifiers that you believe are more accurate or reliable.

Final Prediction: Once the voting strategy is applied, the final prediction is made and returned to the user.

5.8.2 Advantages of Voting Classifier

- **Improved Performance:** Voting classifiers often achieve better predictive performance compared to individual classifiers, especially when the individual classifiers are diverse and make different types of errors.

- **Robustness:** By combining multiple models, voting classifiers can be more robust to outliers, noise, and overfitting.

- **Easy Implementation:** Voting classifiers are relatively simple to implement and can be applied to a wide range of classification tasks without requiring extensive tuning.

5.8.3 Disadvantages of Voting Classifier

- **Complexity:** While voting classifiers are conceptually simple, managing and training multiple individual classifiers can increase the computational complexity and memory requirements of the model.

- **Interpretability:** The final decision made by a voting classifier might be harder to interpret compared to a single classifier, especially if the ensemble consists of many diverse models.

5.8.4 Applications

- **Classification Tasks:** Voting classifiers are widely used in various classification tasks across different domains, including finance, healthcare, marketing, and more.
- **Ensemble Learning:** Voting classifiers are a fundamental technique in ensemble learning, where the goal is to combine multiple models to achieve better predictive performance.

In summary, a voting classifier is a powerful technique for combining the predictions of multiple classifiers to make more accurate and robust predictions. By leveraging the wisdom of the crowd, voting classifiers can often outperform individual classifiers and are commonly used in machine learning for a wide range of applications.

CHAPTER NO - 6

SOFTWARE SPECIFICATIONS

6.1 Jupyter Notebook

Jupyter Notebook is an open-source web application that acts as an interactive environment for creating and sharing documents that combine:

- **Live Code:** You can write code in various programming languages (Python, R, Julia, etc.) within the notebook. The code cells can be executed line by line or as a whole, allowing you to experiment and see the results immediately.
- **Rich Text and Equations:** You can interweave code with formatted text explanations, comments, and even mathematical equations using LaTeX syntax. This makes your notebooks well-documented and easy to understand.
- **Visualizations:** The results of your code can be visualized directly within the notebook. You can generate plots, charts, images, and other graphical outputs to explore and present your findings.

Jupyter Notebook is a popular tool for various purposes, including:

- **Data Science and Machine Learning:** It's widely used for data exploration, analysis, model building, and visualization due to its interactive nature and ease of integrating code, text, and results.
- **Scientific Computing:** Jupyter Notebook is a valuable tool for scientists and researchers to perform calculations, analyze data, and document their work in a clear and reproducible manner.
- **Education and Training:** The interactive environment makes it ideal for learning programming languages, data analysis techniques, and various scientific concepts.

Here are some key benefits of Jupyter Notebook:

- **Open-source and Free:** Anyone can access and use Jupyter Notebook without any licensing fees.

- **Cross-platform:** It runs on various operating systems (Windows, macOS, Linux) making it widely accessible.
- **Interactive and Exploratory:** The ability to run code and see results instantly fosters exploration and experimentation.
- **Reproducible Research:** Notebooks can capture the entire workflow, from data loading and cleaning to analysis and visualization, promoting reproducibility.
- **Community and Sharing:** A large community supports Jupyter Notebook with extensive libraries, tutorials, and the ability to share notebooks easily.

In essence, Jupyter Notebook provides a user-friendly and versatile platform for data scientists, programmers, educators, and anyone who wants to combine code, narrative text, and visualizations for computational tasks.

6.2 MATPLOTLIB

Matplotlib is a powerful Python library for creating static, animated, and interactive visualizations. It's a fundamental tool for data scientists and analysts who need to effectively communicate insights from data.

Core functionalities of Matplotlib:

- **Plot creation:** Matplotlib offers a wide range of plot types, including line charts, scatter plots, bar charts, histograms, heatmaps, and more.
- **Customization:** You can extensively customize the visual appearance of your plots, including colors, fonts, labels, markers, legends, and axes.
- **Integration:** Matplotlib integrates well with other scientific Python libraries like NumPy and Pandas, facilitating data manipulation and analysis before visualization.
- **Embedding:** Plots generated by Matplotlib can be embedded into various graphical user interfaces (GUIs) or Jupyter notebooks for interactive exploration.

Key advantages of Matplotlib:

- **Ease of use:** Matplotlib provides a user-friendly API for creating basic plots with minimal code.
- **Flexibility:** It offers a high degree of control over plot customization, allowing for creation of publication-quality visualizations.
- **Open-source:** Being free and open-source software makes Matplotlib widely accessible and fosters a large community of users and developers.
- **Extensive documentation and tutorials:** Abundant resources are available online to learn and explore the capabilities of Matplotlib.

Use cases of Matplotlib:

- **Exploratory data analysis (EDA):** Visualizing data distributions, relationships between variables, and identifying patterns or anomalies.
- **Data communication:** Creating clear and informative visualizations to communicate findings to technical and non-technical audiences.
- **Scientific publications:** Generating high-quality plots for research papers and presentations.
- **Web applications:** Embedding interactive visualizations into web dashboards for real-time data exploration.

In summary, Matplotlib is a versatile and user-friendly library that serves as the foundation for data visualization in Python. It empowers users to create informative and visually appealing plots to communicate data insights effectively.

6.3 Numpy

NumPy (Numerical Python) is a fundamental library for scientific computing in Python. It provides a powerful and efficient way to work with arrays, which are collections of elements of the same data type.
Here are some key features of NumPy:

- **Multidimensional Arrays:** NumPy introduces the ndarray object, a versatile data structure for creating and manipulating multidimensional arrays. This enables efficient storage and processing of large datasets commonly encountered in scientific computing and data analysis tasks.

- **Mathematical Operations:** NumPy offers a rich collection of mathematical functions that operate on entire arrays element-wise. This allows for vectorized operations, significantly faster than traditional Python loops for element-by-element calculations.

- **Linear Algebra Operations:** NumPy provides functions for linear algebra operations like matrix multiplication, inversion, solving systems of linear equations, and eigenvalue decomposition. These are essential tools for various scientific and engineering applications.

- **Integration with Other Libraries:** NumPy serves as a foundation for many other scientific Python libraries like pandas (data analysis) and scikit-learn (machine learning). These libraries leverage NumPy's array capabilities for efficient data manipulation and algorithm implementation.

Overall, NumPy empowers Python with the capabilities to handle numerical computations effectively. It simplifies working with large datasets and provides high-performance mathematical and linear algebra operations, making it an essential tool for various scientific disciplines, data science, and machine learning applications.

6.4 Pandas

Pandas is a powerful and open-source software library written specifically for Python. It provides high-performance, easy-to-use data structures and data analysis tools designed for working with tabular data (like spreadsheets) and time series data.
Here's a breakdown of its key functionalities:

- **Data Structures:**

  a)Series: A one-dimensional array capable of holding any data type (integers, strings, Python objects). Think of it as a single column from a spreadsheet.

b)DataFrame: A two-dimensional labeled data structure with columns that can hold different data types. Essentially, it's like a spreadsheet with rows and columns.

- **Data Manipulation:** Pandas offers a rich set of functions for loading data from various file formats (CSV, Excel, JSON), cleaning and transforming data (handling missing values, filtering, sorting), and performing mathematical operations on data (calculations, aggregations).
- **Time Series Analysis:** Pandas provides specialized tools for working with time series data, including functionality for indexing data by dates, resampling data at different frequencies, and performing time-based operations.
- **Data Visualization:** While not its primary focus, Pandas integrates well with libraries like Matplotlib and Seaborn for creating informative visualizations of your data (histograms, scatter plots, boxplots).

Here are some of the key benefits of using Pandas for data analysis in Python:

- **Ease of Use:** Pandas offers a user-friendly syntax that makes data manipulation intuitive, even for those less familiar with coding.
- **Efficiency:** It's built on top of NumPy, a powerful numerical computing library, ensuring efficient handling of large datasets.
- **Versatility:** Pandas can handle various data types and structures, making it suitable for a wide range of data analysis tasks.
- **Integration:** It seamlessly integrates with other popular Python data science libraries like Scikit-learn for machine learning and Matplotlib for data visualization.

Overall, Pandas has become an essential tool for data scientists and analysts working with Python. Its ability to simplify data manipulation, analysis, and visualization makes it a cornerstone library for data-driven tasks across various domains.

6.5 Seaborn

Seaborn is a widely used Python library for creating statistical graphics and data visualizations. Built on top of the Matplotlib library, it provides a high-level interface for

creating attractive and informative plots with minimal code.
Here are some key characteristics of Seaborn:

- **Statistical Focus:** Seaborn excels at creating visualizations commonly used in statistical analysis, such as histograms, scatter plots, box plots, violin plots, and heatmaps. These plots are designed to reveal patterns, trends, and relationships within your data.
- **Enhanced Aesthetics:** It offers a set of pre-defined color palettes and styles that make visualizations aesthetically pleasing and publication-ready. You can easily customize these styles to match your preferences or branding.
- **Integration with Pandas:** Seaborn is designed to work seamlessly with pandas DataFrames, a popular data structure in Python for data analysis. This allows you to easily create visualizations directly from your pandas data.
- **High-Level API:** Seaborn provides a concise and user-friendly API that simplifies the creation of complex statistical plots. You can focus on the data and insights you want to communicate, rather than the underlying technical details of plot creation.

Here are some common use cases for Seaborn:

- **Exploratory Data Analysis (EDA):** Exploring the distribution of variables, identifying relationships between features, and uncovering potential outliers.
- **Statistical Modeling:** Visualizing model predictions and residuals to assess model performance and identify areas for improvement.
- **Data Storytelling:** Communicating insights and findings from data analysis clearly and compellingly.

Overall, Seaborn is a valuable tool for anyone working with data in Python, particularly those in data science, statistics, and machine learning. It empowers you to create informative and visually appealing visualizations that enhance your understanding of the data and effectively communicate your findings to others.

6.6 Scikit-learn

Scikit-learn, also commonly abbreviated as sklearn, is a free and open-source machine learning library for the Python programming language. It's a fundamental tool widely used for various machine learning tasks, particularly within the scientific and data analysis communities.

Here are some key characteristics of scikit-learn:

- **Comprehensive functionality:** It provides a broad range of machine learning algorithms, encompassing:

  a) Supervised learning: For tasks like classification (predicting categories) and regression (predicting continuous values). Examples include Logistic Regression, Random Forests, Support Vector Machines, and Gradient Boosting.

  b) Unsupervised learning: For tasks like clustering (grouping similar data points) and dimensionality reduction (reducing the number of features). Examples include K-means clustering and Principal Component Analysis (PCA).

- **Ease of use:** Scikit-learn offers a consistent and user-friendly API (Application Programming Interface) that simplifies the process of working with machine learning models. It follows a common workflow for training, evaluating, and deploying models.

- **Interoperability:** Scikit-learn is designed to work seamlessly with other popular Python scientific libraries like NumPy (numerical computing) and SciPy (scientific computing). This integration allows for efficient data manipulation and advanced mathematical operations.

- **Open-source development:** Scikit-learn is actively developed and maintained by a large community of contributors. This ensures continuous improvement, bug fixes, and new features.

Overall, scikit-learn is a powerful and versatile machine-learning library for Python. Its extensive functionality, user-friendly interface, and community support make it a popular choice for various data science applications, including customer churn prediction as discussed previously.

6.7 Python

Python is a powerful, versatile, high-level, translated, object-arranged, unusual-state programming language with dynamic semantics. Its unusual state worked in information structures, combined with dynamic composing and dynamic authoritative, making it attractive for Rapid Application Development, just as a scripting or paste language to interface existing segments together.
Here's a breakdown of its key characteristics:

- **High-Level:** Python code is known for its readability and simplicity. It uses clear syntax that resembles natural language, making it easier to learn and write compared to more complex languages.
- **General-Purpose:** Python can be applied to a wide range of programming tasks across various domains. From web development and data science to machine learning and automation, Python's extensive libraries and frameworks empower you to tackle diverse programming challenges.
- **Interpreted Language:** Unlike compiled languages that require translation into machine code before execution, Python code is interpreted line by line at runtime. This simplifies development and debugging as you can see the results immediately.
- **Object-Oriented:** Python supports object-oriented programming (OOP) paradigms. This allows you to structure your code using objects, which encapsulate data (attributes) and related operations (methods). OOP promotes code modularity, reusability, and maintainability.
- **Dynamically Typed:** Python is dynamically typed, meaning you don't need to explicitly declare the data type of variables. This offers flexibility during development, but type hints can be added for improved readability and potential static type checking.
- **Open-Source and Community-Driven:** Python is an open-source language with a large and active community. This translates to a wealth of freely available libraries, frameworks, and resources, fostering collaboration and innovation within the Python ecosystem.

Here are some popular applications of Python:

- **Web Development:** Frameworks like Django and Flask enable building web applications with Python.
- **Data Science and Machine Learning:** Libraries like NumPy, Pandas, TensorFlow, and PyTorch empower data analysis, machine learning model development, and scientific computing.
- **Scripting and Automation:** Python's versatility makes it suitable for automating repetitive tasks and system administration.

In summary, Python's readability, versatility, and extensive ecosystem make it a popular choice for programmers of all levels. From beginners to experienced developers, Python offers a powerful and user-friendly language for tackling diverse programming challenges.

CHAPTER NO - 7

RESULT ANALYSIS AND VALIDATION

In machine learning, evaluating the classifier's performance is crucial, particularly in selecting the optimal algorithm for a given problem. Previous research on churn prediction predominantly employs metrics such as accuracy, precision, recall, and F-measure, all derived from the confusion matrix. Our study follows suit, assessing algorithm performance and efficiency using these metrics.However, when dealing with imbalanced data, accuracy may not provide an accurate reflection of the algorithm's performance. Therefore, we prioritize precision, recall, and F1-score, specifically focusing on the target class, which in our case is Churn. Among these metrics, we emphasize F1-score as it strikes a balance between precision and recall.

In our application, precision denotes the rate of correctly classified churn instances, while recall measures the model's ability to predict actual churners. Given our focus on churn prediction, recall holds greater importance than precision. Evaluation is conducted on unseen instances from the test set, ensuring that the algorithm's performance is assessed on data it has not been trained on.

7.1 Performance Based Evaluation Matrix

7.1.1 Confusion Matrix

A confusion matrix is a fundamental concept in machine learning, providing insight into the classifier's performance by detailing actual and predicted classifications. It comprises True Positives (TP) and True Negatives (TN), which denote correctly classified test instances, and False Negatives (FN) and False Positives (FP), representing incorrectly classified instances.

- **TP (True Positive):** These are the customers the model correctly predicts will stay subscribed (active/not churn), and they actually do not churn. This represents a successful prediction of customer retention.
- **TN (True Negative):** These are the customers the model correctly predicts will churn (cancel their subscription), and they actually do churn. This represents a successful prediction of customer churn.

- **FP (False Positive):** These are the customers the model incorrectly predicts will churn (cancel their subscription), but they actually do not churn. This is a type of error where the model mistakenly identifies a loyal customer as at risk.
- **FN (False Negative):** These are the customers the model incorrectly predicts will stay subscribed (active/not churn), but they actually do churn. This is a more critical error, as the model misses an opportunity to intervene and retain a valuable customer.

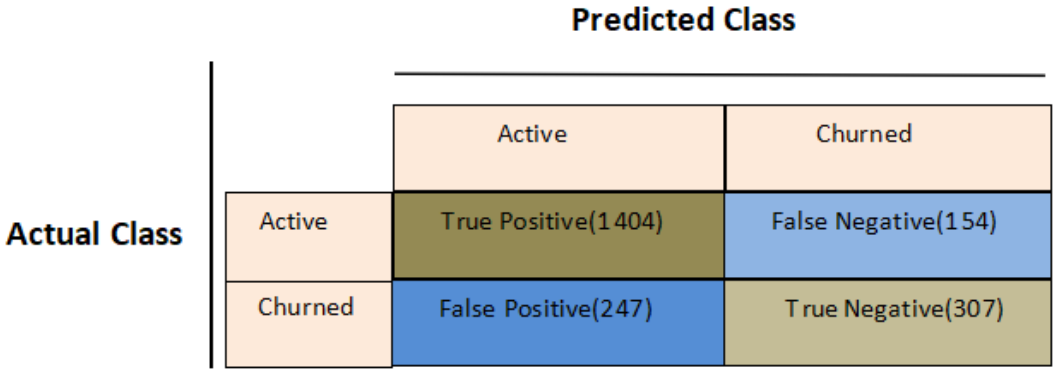

Figure 7.1: confusion matrix

7.1.2 Recall

In a prediction model, recall assesses how well the model can accurately detect all the pertinent instances within a dataset. It specifically gauges the ratio of true positive predictions, which are the correctly identified positive instances, to all the actual positive instances in the dataset.It is the proportion of Active (or Churn) customer for the correctly identified.Calculating with the help of equation (2)

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7.1}$$

7.1.3 Precision

In the prediction model, precision is a performance metric, It quantifies the proportion of true positive predictions (correctly identified positive instances) out of all instances

predicted as positive by the model.It is the proportion of the predicted cases.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{7.2}$$

7.1.4 Accuracy

A metric called accuracy indicates the classifier's general efficacy. It is a measure that displays the percentage of cases that are correctly classified overall. Accuracy is described

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FN+FP} \tag{7.3}$$

7.1.5 F-Measure

In a prediction model, the F-measure, also known as the F1 score, is a combined metric that balances both precision and recall. It provides a single numerical value that evaluates the model's performance by considering both the rate of correctly predicted positive instances (precision) and the model's ability to capture all positive instances (recall).The F-measure is the harmonic mean of precision and recall and is calculated using the following formula:

$$F-Measure = \frac{2 \times (\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \tag{7.4}$$

7.1.6 Support

In the context of a confusion matrix used for customer churn prediction, "support" refers to the total number of data points for a particular class. There are two main ways "support" can be interpreted depending on the specific class you're considering:
1. Support for the Churn Class: This refers to the total number of customers in the dataset who actually churned (cancelled their subscription).In the confusion matrix, this value would typically be displayed in the bottom row, under the "Churn" class label.
2. Support for the Non-Churn Class: This refers to the total number of customers in the dataset who did not churn (remained subscribed). In the confusion matrix, this value would be displayed in the far right column, under the "Non-Churn" class label.
Analyzing the support values alongside other metrics can provide insights into the model's performance for different classes.

- A high support value for the churn class and a low number of True Positives might indicate the model struggles to identify churners effectively.

- A high support value for the non-churn class and a high number of False Positives might suggest the model is overly cautious, incorrectly flagging too many loyal customers as at-risk.

Support in a confusion matrix provides the context for interpreting other evaluation metrics. It helps us understand the distribution of classes within the data and how well the model performs in identifying both churners and non-churners.
Different types of data possess unique characteristics that can present challenges for data mining algorithms aiming to extract meaningful patterns.

Table 7.1: Performance Metrics of Various Algorithms

Algorithms Used	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Decision Tree	0.82	0.80	0.81	0.72
KNN	0.84	0.88	0.86	0.79
Ada Boost	0.85	0.90	0.87	0.80
Gradient Boost	0.84	0.91	0.87	0.80
Logistic Regression	0.85	0.90	0.88	0.81
Random Forest	0.83	0.92	0.87	0.80
SVM	0.83	0.91	0.87	0.80
Voting Classifier	0.85	0.91	0.88	0.81

In Table (7.1), we calculate these values of precision, recall, F1-score, and accuracy of different algorithms by using the confusion matrix parameters.

7.1.7 ROC-Curve

A ROC Curve (Receiver Operating Characteristic Curve) is a visual tool used to evaluate the performance of binary classification models. In customer churn prediction for subscription services, it helps assess how well the model distinguishes between customers who will churn (positive class) and those who will stay subscribed (negative class).
The Axes are mentioned below:

- X-axis (False Positive Rate - FPR): The proportion of customers incorrectly classified as churners (out of total non-churners).

- Y-axis (True Positive Rate - TPR): The proportion of customers correctly classified as staying subscribed (out of total loyal customers).

The ideal Curve for the perfect model would perfectly differentiate churners and non-churners. This translates to a ROC curve that follows the left border and top border of the graph, ultimately reaching the top-left corner (FPR = 0, TPR = 1).
The closer the ROC curve is to the top-left corner, the better the model's performance at distinguishing churners from non-churners. A curve closer to the diagonal line (dashed line) indicates a model no better than random chance.
ROC Curve Benefits:

- It allows for comparing the performance of different classification models for churn prediction. The model with the ROC curve closest to the top-left corner is generally considered superior.

- An ROC curve can help determine an appropriate churn risk threshold. By setting a threshold on the TPR axis (e.g., 0.8 - identifying 80% of churners correctly), you can classify customers exceeding that threshold as high-risk for churn.

- The ROC curve highlights the trade-off between TPR and FPR. Increasing the TPR (catching more churners) might also lead to an increase in FPR (mistakenly identifying non-churners). It helps to visualize this relationship for decision-making.

The ROC Curve is a valuable tool for understanding the customer churn prediction model that differentiates between churning and loyal customers. It helps visualize model performance and choose an appropriate churn risk threshold for proactive retention strategies.

7.1.8 Error Rate in KNN

The error rate in K-Nearest Neighbors (KNN) refers to the proportion of mistakes the model makes when classifying new data points. It represents the percentage of predictions that are incorrect.
Classification Error Rate is the most common metric and is simply the number of misclassified data points divided by the total number of data points:
Error Rate = (Number of Misclassified Points) / (Total Number of Data Points)

Factors Affecting Error Rate of KNN:

- **K Value:** The number of neighbors (K) used for prediction significantly impacts error rate. A low K value can lead to overfitting, where the model memorizes the training data but performs poorly on unseen data. Conversely, a high K value can lead to underfitting, where the model fails to capture the underlying patterns in the data. Finding the optimal K value is crucial for minimizing error rate.

- **Distance Metric:** The distance metric used to calculate the similarity between data points (e.g., Euclidean distance, Manhattan distance) can affect the error rate. Choosing an appropriate metric depends on the characteristics of your data.

- **Data Quality:** The quality and distribution of our training data can significantly impact the error rate. Noisy or imbalanced data can lead to higher error rates.

- **Dimensionality of Data:** For datasets with high dimensionality (many features), the KNN model might struggle to find meaningful nearest neighbors, leading to increased error rates.

Interpreting Error Rate: A lower error rate indicates a more accurate KNN model. However, the desired threshold for a "good" error rate depends on the specific problem and the cost of misclassification.Techniques like cross-validation can be used to get a more robust estimate of the KNN model's error rate on unseen data.
By understanding the factors affecting error rate and employing appropriate techniques, We can optimize your KNN model for better performance in customer churn prediction or other classification tasks

7.2 Algorithm Based Evaluation

In algorithm evaluation, We consider a variety of algorithms such as logistic regression, KNN, SVM, Voting Classifier, Gradient Boost, AdaBoost, Decision Tree, and Random Forest. We import several libraries such as Pandas, NumPy, Matplotlib, and Seaborn to evaluate the accuracy score. We convert categorical values like Gender, Monthly Charges, and Payment Method into numerical and binary formats to assess churn and non-churn conditions in the confusion matrix.

Logistics regression model positive and negative rates refer to the proportion of correctly predicted positive and negative instances, respectively. With an accuracy of 81.01% the instances in the dataset are classified correctly by the logistics regression model.

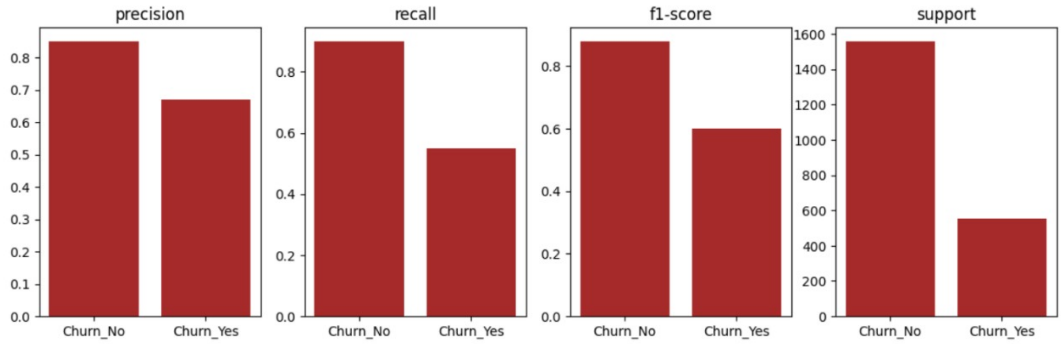

Figure 7.2: Classification Report of Logistic Regression

In Figure (7.2), We represented the bar graph of comparison of precision, recall, f1-score, and support for churn_No and churn_Yes.

In the support vector machine, the positive and negative rates refer to the proportion of correctly predicted positive and negative instances, respectively. With an accurracy 80.20% in the dataset are classified correctly by the support vector classifier.

```
print(classification_report(Y_test, y_pred_svc))
```

	precision	recall	f1-score	support
0	0.83	0.91	0.87	1558
1	0.67	0.49	0.57	554
accuracy			0.80	2112
macro avg	0.75	0.70	0.72	2112
weighted avg	0.79	0.80	0.79	2112

Figure 7.3: Classification Report of Support Vector Machine

In Figure (7.3), We represented the classification report of precision, recall, f1-score, accuracy, and support for churn_No(Active) = 0 and churn_Yes(Churned) = 1.

In the AdaBoost Classifier, the positive and negative rates refer to the proportion of correctly predicted positive and negative instances, respectively. With an accuracy of 80.70% the dataset is classified correctly by the AdaBoost Classifier.

```
print(classification_report(Y_test, y_pred_ab))
```

	precision	recall	f1-score	support
0	0.85	0.90	0.87	1558
1	0.66	0.55	0.60	554
accuracy			0.81	2112
macro avg	0.75	0.72	0.74	2112
weighted avg	0.80	0.81	0.80	2112

Figure 7.4: Classification Report of Adaptive Boosting Classifier

In Figure (7.4), We represented the classification report of precision, recall, f1-score, accuracy, and support for the AdaBoost Classifier.Here churn_No(Active) = 0 and churn_Yes(Churned) = 1.

In the Gradient Boosting Classifier, the positive and negative rates refer to the proportion of correctly predicted positive and negative instances, respectively. With an accuracy of 80.70%, the datasets are classified correctly by the gradient boosting Classifier.

```
print(classification_report(Y_test, y_pred_gbc))
```

	precision	recall	f1-score	support
0	0.84	0.91	0.87	1558
1	0.66	0.51	0.58	554
accuracy			0.80	2112
macro avg	0.75	0.71	0.73	2112
weighted avg	0.79	0.80	0.80	2112

Figure 7.5: Classification Report of Gradient Boosting Classifier

In Figure (7.5), We represented the classification report of precision, recall, f1-score,

accuracy, and support for the Gradient Boosting Classifier.Here Not_Churned(Active) = 0 and Churned_Yes(Churned) = 1.

In the Decision tree Classifier, the positive and negative rates refer to the proportion of correctly predicted positive and negative instances, respectively. With an accuracy of 72.01% the dataset is classified correctly by the decision tree classifier.

In KNN, the n neighbor value is 30, and the positive and negative rates refer to the proportion of correctly predicted positive and negative instances, respectively. With an accuracy of 78.64%, the dataset is classified correctly by the KNN, which is shown in Fig (7.6)

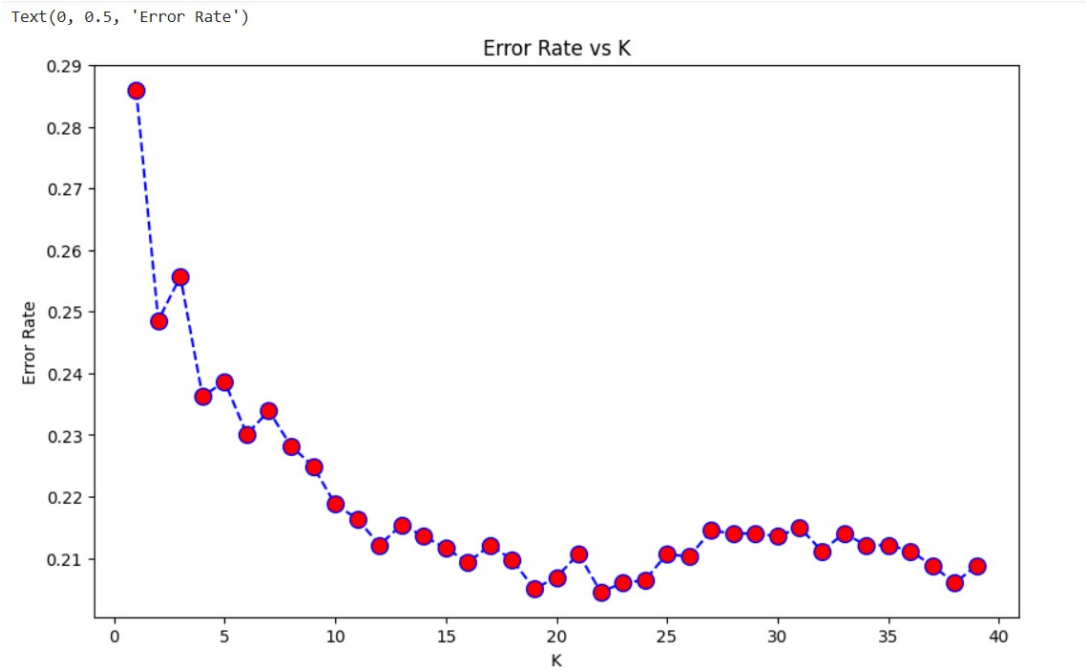

Figure 7.6: Error Rate in KNN

In the random forest model, the positive and negative rates refer to the proportion of correctly predicted positive and negative instances, respectively. With an accuracy of 79.78% the dataset is classified correctly by the random forest model.

In Figure(7.7) We can see the model configuration, the parameter values are set as follows n estimators is assigned the value of 500, OOB score is set to True, n jobs is set to -1 to utilize all available processors, and random state is set to 50 for reproducibility, max features is specified as ”sqrt”, and max-leaf nodes is set to 30. These parameter values are chosen to optimize the performance and efficiency of the model.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import roc_curve
rf = RandomForestClassifier(n_estimators=500 , oob_score = True, n_jobs = -1,
                                    random_state =50, max_features = "sqrt",
                                    max_leaf_nodes = 30)
rf.fit(X_train, Y_train)

y_pred_rf= rf.predict(X_test)
```

Figure 7.7: Parameters in Random Forest

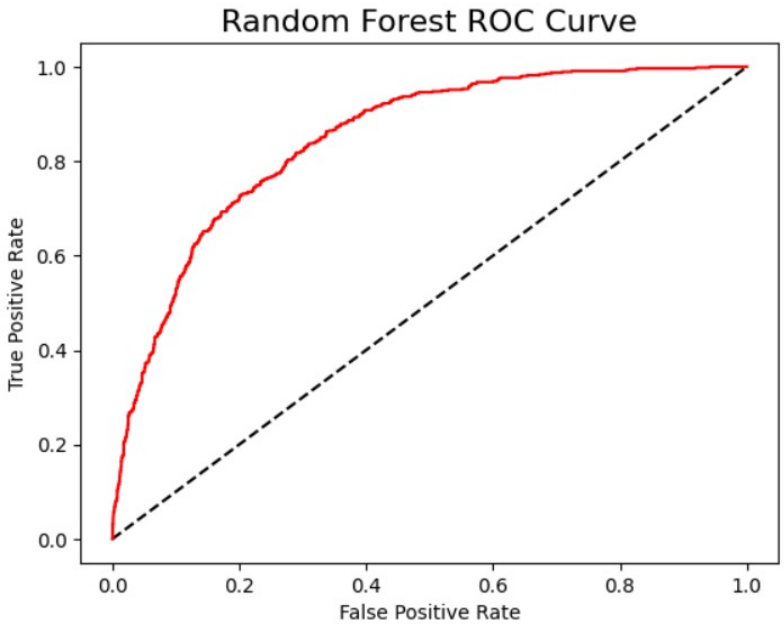

Figure 7.8: ROC Curve in Random Forest

In the figure(7.8) ROC curve, the true positive rate (TPR) is plotted on the y-axis, and the false positive rate (FPR) is plotted on the x-axis. Each point on the curve represents a different threshold setting, and the curve illustrates how the trade-off between true positives and false positives changes as the threshold varies.

In the Figure(7.9) voting classifier, there is three estimators have been used, which are Gradient Boosting Classifier, Logistic Regression, and AdaBoost Classifier. Additionally, a Voting Classifier with soft voting has been employed, as it tends to yield higher accuracy by combining the predictions of multiple base estimators.These scores are calculated using the ”predicted class” and ”actual class” parameters in the confusion matrix.

```
from sklearn.ensemble import VotingClassifier
clf1 = GradientBoostingClassifier()
clf2 = LogisticRegression()
clf3 = AdaBoostClassifier()
eclf1 = VotingClassifier(estimators=[('gbc', clf1), ('lr', clf2), ('abc', clf3)], voting='soft')
eclf1.fit(X_train, Y_train)
predictions = eclf1.predict(X_test)
```

Figure 7.9: Parameters in Voting Classifier

In Figure (7.10), We represented the classification report of precision, recall, f1-score, accuracy, and support for the Voting Classifier.Here 0 is using for Active_Customers and 1 is using for Churned_Customers

```
print(classification_report(Y_test, predictions))
```

	precision	recall	f1-score	support
0	0.85	0.91	0.88	1558
1	0.67	0.54	0.60	554
accuracy			0.81	2112
macro avg	0.76	0.72	0.74	2112
weighted avg	0.80	0.81	0.80	2112

Figure 7.10: Classification report of Voting Classifier

In the figure (7.11) confusion matrix, we can observe that there are two rows and two columns named active and churned. This confusion matrix has four boxes that depict True Positive, True Negative, False Positive, and False Negative.We can use these values to calculate the recall, precision , and accuracy of an algorithm.
Their values are mentioned below:

- True positive contains the value 1253,
- True Negative contains the value 272,
- False Positive contains the value 282
- False Negative contains the value 305.

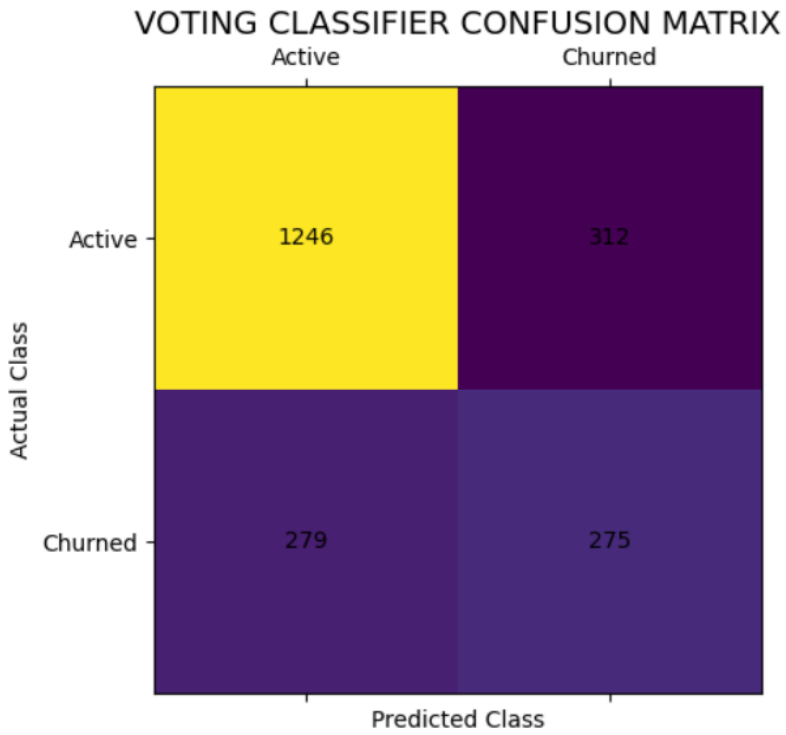

Figure 7.11: Confusion Matrix Of Voting Classifier

In the Voting Classifier, the positive and negative rates refer to the proportion of correctly predicted positive and negative instances, respectively. With an accuracy of 81.06%, the dataset is classified correctly by the voting classifier.

```
print("Final Accuracy Score of voting classifier: ")
print(accuracy_score(Y_test, predictions))
```

```
Final Accuracy Score of voting classifier:
0.8106060606060606
```

Figure 7.12: Accuracy Of Voting Classifier

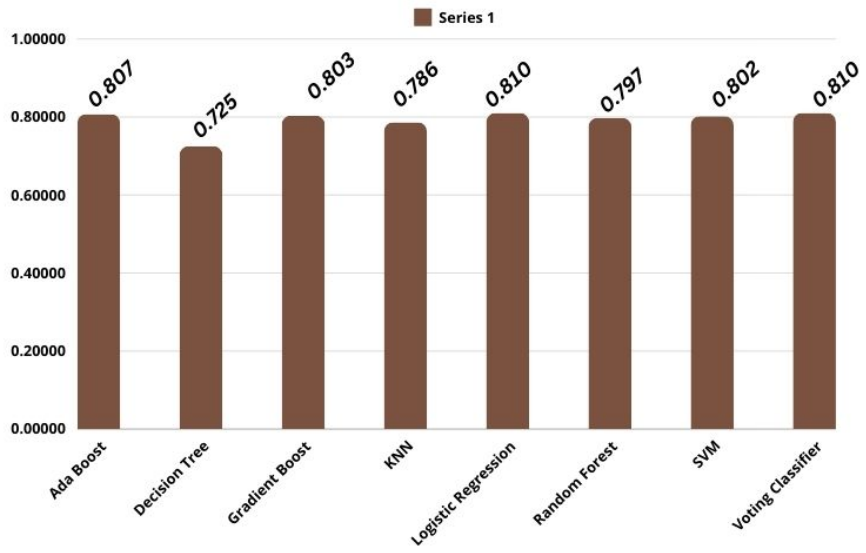

Figure 7.13: Accuracy Of All Algorithms

The final score of the voting classifier and logistics regression algorithm demonstrating the highest accuracy with the values 81.01 and 81.06 respectively, is depicted in Figure (7.13).

CHAPTER NO - 8

CONCLUSION

In conclusion, our analysis of customer churn prediction in subscription services reveals valuable insights when considering recall, precision, confusion matrix, F-measure, and utilizing Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn. We employed various machine learning algorithms including Random Forest, Decision Tree, Support Vector Machine, Logistic Regression, Voting Classifier, and Gradient Classifier. Additionally, we utilized techniques such as one-hot encoding and dummy variables for handling categorical data.

Recall and precision metrics help assess the effectiveness of our models in correctly identifying churned customers while minimizing false positives. A higher recall indicates the ability to capture more true positives, while higher precision suggests fewer false positives.The confusion matrix provides a comprehensive summary of the model's performance, depicting true positives, true negatives, false positives, and false negatives. This aids in understanding the strengths and weaknesses of the classification models.The F-measure, which combines both recall and precision into a single metric, offers a balanced assessment of the model's overall performance. It is particularly useful when there is an imbalance between the classes.By leveraging Python libraries such as Pandas and NumPy, we efficiently preprocessed the data and prepared it for model training. Visualization libraries like Matplotlib and Seaborn aided in data exploration and performance evaluation, enhancing our understanding of the results.Utilizing Scikit-learn, we implemented a variety of machine learning algorithms, each with its strengths and weaknesses. Random Forest, Decision Tree, Support Vector Machine, Logistic Regression, Voting Classifier, and Gradient Classifier were evaluated, providing diverse perspectives on the problem.Finally, techniques such as one-hot encoding and dummy variables were employed to handle categorical data effectively, ensuring compatibility with the machine learning algorithms.

CHAPTER NO - 9

FUTURE SCOPE

The future of customer churn prediction in subscription services holds exciting possibilities that can further enhance customer retention and business growth.

Deep Learning includes leveraging architectures like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) can handle complex sequential data (call history, usage patterns) and potentially improve churn prediction accuracy. Explainable AI (XAI) is developing models that are not only accurate but also interpretable will be crucial. XAI techniques can help understand the "why" behind churn predictions, leading to more targeted and effective retention strategies. Ensemble Learning combined multiple machine learning models can leverage the strengths of each approach to create a more robust and accurate overall prediction system. Moving beyond batch processing, real-time churn prediction can identify at-risk customers as their behavior changes. This allows for immediate intervention with personalized offers or support, potentially preventing churn before it happens. Seamless integration with CRM systems allows for automated workflows based on churn risk scores. This streamlines the process of targeting high-risk customers with appropriate retention efforts. As models become more sophisticated, ensuring fairness and mitigating potential biases within data or algorithms will be crucial. Techniques like fairness metrics and bias detection methods can help create more equitable and trustworthy churn prediction systems.

REFERENCES

1. D. Manzano-Machob, "The architecture of a churn prediction system based on stream mining," in Proc. Artif. Intell. Res. Develop., 16th Int. Conf. Catalan Assoc. Artif. Intell., vol. 256, Oct. 2013, p. 157.

2. S. Babu, D. N. Ananthanarayanan, and V. Ramesh, "A survey on factors impacting churn in telecommunication using data mininig techniques," Int.J. Eng. Res. Technol., vol. 3, no. 3, pp. 1745–1748, Mar. 2014.

3. A. Idris and A. Khan, "Customer churn prediction for telecommunication:Employing various various features selection techniques and tree based ensemble classifiers," in Proc. 15th Int. Multitopic Conf., Dec. 2012,pp. 23–27.

4. A. Amin et al., "Customer churn prediction in the telecommunication sector using a rough set approach," Neurocomputing, vol. 237, pp. 242–254,May 2017.

5. A. Amin et al., "Cross-company customer churn prediction in telecommunication: A comparison of data transformation methods," Int. J. Inf. Manage., vol. 46, pp. 304–319, Jun. 2019.

6. L. Zhao, Q. Gao, X. Dong, A. Dong, and X. Dong, "K-local maximum margin feature extraction algorithm for churn prediction in telecom,"Cluster Comput., vol. 20, no. 2, pp. 1401–1409, Jun. 2017.

7. A. Amin, F. Al-Obeidat, B. Shah, A. Adnan, J. Loo, and S. Anwar,"Customer churn prediction in telecommunication industry using data certainty," J. Bus. Res., vol. 94, pp. 290–301, Jan. 2019.

8. H.-S. Kim and C.-H. Yoon, "Determinants of subscriber churn and customer loyalty in the Korean mobile telephony market," Telecommun. Policy, vol. 28, nos. 9–10, pp. 751–765, Nov. 2004.

9. S. A. Qureshi, A. S. Rehman, A. M. Qamar, A. Kamal, and A. Rehman,"Telecommunication subscribers' churn prediction model using machine learning," in Proc. 8th Int. Conf. Digit. Inf. Manage., Sep. 2013, pp. 131–136

10. V. Lazarov and M. Capota, “Churn prediction,” Bus. Anal. Course,TUM Comput. Sci, Technische Univ. Munchen, Tech. Rep., 2007 ¨

11. Y. Huang, B. Huang, and M.-T. Kechadi, “A rule-based method for customer churn prediction in telecommunication services,” in Proc. Pacific– Asia Conf. Knowl. Discovery Data Mining. Berlin, Germany: Springer, 2011, pp. 411–422.

12. V. L. Migueis, D. van den Poel, A. S. Camanho, and J. F. e Cunha, ´ “Modeling partial customer churn: On the value of first productcategory purchase sequences,” Expert Syst. Appl., vol. 12, no. 12, pp. 11250–11256, Sep. 2012.

13. Nasebah Almufadi, Ali Mustafa Qamar, Rehan Ullah Khan and Mohamed Tahar Ben Othman, ”Deep Learning-based Churn Prediction of Telecom Subscribers”, International Journal of Engineering Research and Technology, vol. 12, no. 12, pp. 2743-2748, 2019, ISSN 0974-3154.

14. Joao B. G. Brito, Guilherme B. Bucco, Rodrigo Heldt, Jo ˜ ao L. Becker, ˜ Cleo S. Silveira, Fernando B. Luce, Michel J. Anzanello, ”A framework to improve churn prediction performance in retail banking”, Financial Innovation, vol.10, no.1, 2024.

15. Nikita Khandelal, Vikas Sakalle, ”Customer Churn Prediction in Telecommunication, Medical Industry Using Machine Learning Classification Models”, 2023 6th International Conference on Contemporary Computing and Informatics (IC3I), vol.6, pp.1727-1734, 2023.

16. A. D. Caigny, K. Coussement, and K. W. D. Bock, “A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees,” Eur. J. Oper. Res., vol. 269, no. 2, pp. 760–772,Sep. 2018.

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

18. Zhan J, Guidibande V, Parsa SPK. Identification of top-k influential communities in big networks. J Big Data. 2016;3(1):16.

19. He Y, He Z, Zhang D. A study on prediction of customer churn in fixed communication network based on data mining. In: Sixth international conference on fuzzy systems and knowledge discovery, vol. 1. 2009. p. 92–4.

20. Makhtar M, Nafis S, Mohamed M, Awang M, Rahman M, Deris M. Churn classification model for local telecommunication company based on rough set theory. J Fundam Appl Sci. 2017;9(6):854–68.

21. Amin A, Anwar S, Adnan A, Nawaz M, Howard N, Qadir J, Hawalah A, Hussain A. Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study. IEEE Access. 2016;4:7940–57.

22. Kumar S, Chandrakala D: A Survey on Customer Churn Prediction using Machine Learning Techniques. Int. J. Comput. Appl. 2016; 154(10): 13–16.

23. 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

24. P. Routh, A. Roy and J. Meyer, "Estimating customer churn under competing risks", Journal of the Operational Research Society, vol. 72, pp. 1138-1155, 2021

25. Huang F, Zhu M, Yuan K, Deng EO. Telco churn prediction with big data. In: ACM SIGMOD international conference on management of data. 2015. p .607–18.

26. P. T. Noi and M. Kappas, "Comparison of random forest k-nearest neighbor and support vector machine classifiers for land cover classification using sentinel-2 imagery", Sensors (Basel Switzerland), vol. 18, 2018.

27. R. I. Dzerzhinsky, M. D. Trifonov and E. V. Ledovskaya, "The support vectors and random forest methods analysis in the forecasting customer churn problem in banking services", Lecture Notes in Networks and Systems, 2021.

28. I. Ahmad, M. Basheri, M. J. Iqbal and A. Rahim, "Performance comparison of support vector machine random forest and extreme learning machine for intrusion detection", IEEE Access, vol. 6, pp. 33789-33795, 2018.

29 M. Pondel, M. Wuczynski, W. Gryncewicz, L. Lysik, M. Hernes, A. Rot, et al., "Deep learning for customer churn prediction in e-commerce decision support", BIS, 2021.

30 Y. T. Naing, M. Raheem and N. K. Batcha, "Feature Selection for Customer Churn Prediction: A Review on the Methods & Techniques applied in the telecom industry," 2022IEEE International Conference on Distributed Computing and Electrical Circuits and electronics (ICDCECE), Ballari, India, 2022, pp. 1-5, doi:10.1109/ICDCECE53908.2022.9793315

31 L. Jovanovic, M. Kljajic, V. Mizdrakovic, V. Marevic, M. Zivkovic and N. Bacanin, "Predicting Credit Card Churn: Application of XGBoost Tuned by Modified Sine Cosine Algorithm," 2023 3rd International Conference on Smart Data Intelligence (ICSMDI), Trichy, India, 2023, pp. 55-62, doi: 10.1109/ICSMDI57622.2023.00018

32 P. K. Dalvi, S. K. Khandge, A. Deomore, A. Bankar and V. A. Kanade, "Analysis of customer churn prediction in telecom industry using decision trees and logistic regression," 2016 Symposium on Colossal Data Analysis and Networking (CDAN), Indore, India, 2016, pp. 1-4, doi: 10.1109/CDAN.2016.7570883.

33 https://github.com/udaybhan10/CustomerChurnPrediction

34 https://www.kaggle.com/code/bhartiprasad17/customer-churn-prediction#-2.-Loading-libraries-and-data

35 https://www.diva-portal.org/smash/get/diva2:1574424/FULLTEXT01.pdf