VIETNAM NATIONAL UNIVERSITY

UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF INFORMATION SYSTEM

FINAL PROJECT REPORT

SUBJECT: AI IN BUSINESS

TOPIC: An enhanced model for predicting customer churn rates and improving retention using Causal AI in the telecom

Lecturer: Le Hoanh Su

Group:

1. Tiêu Đăng Vinh - K224161842
2. Lê Nguyễn Minh Tài - K224101336
3. Lê Phạm Quỳnh Anh - K224161803
4. Vũ Quốc Khánh - K224161821
5. Lê Minh Tiến - K224161837

TP. HCM, November 20th 2024

Instructor: Le Hoanh Su

Group's members:

No.	Full name	Student ID	Point / 10 (Individual Contribution)	Signature
1	Tieu Dang Vinh	K224161842	10/10	Vinh
2	Le Nguyen Minh Tai	K224101336	10/10	[illegible]
3	Le Pham Quynh Anh	K224161803	10/10	Anh
4	Le Minh Tien	K224161837	10/10	[illegible]
5	Vu Quoc Khanh	K224161821	10/10	Khanh

Acknowledgement

Our team would like to express our sincere gratitude to Mr. Le Hoanh Su- lecturer from the Information Systems Department at the University of Economics and Law for your dedicated guidance. Through your lessons, our group have been exposed to a wealth of useful knowledge and skills related to Artificial intelligence in business analytics to present our final project report. Although we dedicated ourselves to acquiring the relevant knowledge and skills required to execute the project, we still encountered some limitations during the process. As a result, our group welcome and greatly appreciate any feedback from you to refine and enhance our abilities and expertise for subsequent endeavors.

Commitment

Our team assures that the content of project: “An enhanced model for predicting customer churn rates and improving retention using Causal AI in the telecom” is honest and authentic and reference to relevant sources to serve our research purpose. Our group take full responsibility for any instance of deception or deviation that may arise.

Table of contents

List of tables

List of figures

List of Acronyms

AI: Artificial Intelligence

XAI: Explainable Artificial Intelligence

ANN: Artificial Neural Network

DNN: Deep Neural Network

XGBoost: Extreme Gradient Boosting

SCM: Structural Causal Models

CFChurn: Counterfactual Churn

TP: True Positive

FP: False Positive

FN: False Negative

TN: True Negative

DSL: Digital Subscriber Line

PLDA: Probabilistic Linear Discriminant Analysis

CNN: Convolutional Neural Network

RNN: Recurrent Neural Network

PM2.5: Particulate Matter less than 2.5 micrometers

NIST: National Institute of Standards and Technology

ISP: Internet Service Provider

NLP: Natural Language Processing

ML: Machine Learning

DL: Deep Learning

EDA: Exploratory Data Analysis

AUC: Area Under the Curve

ROC: Receiver Operating Characteristic

SMOTE: Synthetic Minority Over-sampling Technique

IV: Instrumental Variable

Abstract

This research presents an improved model for predicting customer churn rates and enhancing retention strategies in the telecommunications industry by applying causal analysis through Causal AI. Customer churn remains a major challenge for telecommunications providers, significantly impacting revenue and market share. Existing predictive methods are often ineffective due to their reliance on correlations, lacking the ability to identify true causal relationships. This research evaluates the results of machine learning models such as XGBoost, ANN, and DNN, combined with Causal AI techniques, to pinpoint the root causes of customer churn. The analysis highlights key factors influencing churn, including contract types, service usage levels, and payment methods. The findings demonstrate the model's accuracy in prediction and its capacity to provide actionable insights, enabling telecom companies to implement timely interventions. By addressing both prediction and causation, this study contributes to improving customer retention strategies, enhancing business sustainability, and promoting the integration of Causal AI in business analytics.

Tóm tắt

Nghiên cứu này trình bày một mô hình cải tiến nhằm dự đoán tỷ lệ rời bỏ khách hàng và cải thiện các chiến lược giữ chân khách hàng trong ngành viễn thông ứng dụng việc phân tích các yếu tố nhân quả bằng mô hình Causal AI. Khách hàng rời bỏ luôn là một thách thức lớn đối với các nhà cung cấp viễn thông ảnh hưởng đến doanh thu và thị phần. Các phương pháp dự đoán hiện có thường chưa hiệu quả do phụ thuộc vào mối tương quan trong khi lại thiếu đi khả năng nhận biết các mối quan hệ nhân quả thực sự. Nghiên cứu này đánh giá kết quả các mô hình học máy như XGBoost, ANN, và DNN, kết hợp cùng với các kỹ thuật Causal AI để xác định các nguyên nhân gốc rễ của rời bỏ khách hàng. Điều này đã làm nổi bật các yếu tố quan trọng ảnh hưởng đến rời bỏ khách hàng bao gồm loại hợp đồng, mức độ sử dụng dịch vụ và phương thức thanh toán. Kết quả cho thấy độ chính xác của mô hình trong dự đoán và khả năng cung cấp các thông tin giúp các công ty viễn thông triển khai các biện pháp can thiệp kịp thời. Bằng cách giải quyết cả vấn đề dự đoán và nhân quả, nghiên cứu này đóng góp vào việc cải thiện các chiến lược giữ chân khách hàng, nâng cao sự phát triển bền vững của doanh nghiệp và thúc đẩy tích hợp Causal AI trong phân tích kinh doanh.

CHAPTER 1: INTRODUCTION TO THE RESEARCH PROBLEM

1.1 Research problem

Customer churn is a pervasive issue in the telecommunications sector, where high competition and rapidly evolving consumer preferences pose significant challenges for companies aiming to maintain customer loyalty. Churn, defined as the discontinuation of service by a customer, has far-reaching implications for telecommunications providers, affecting not only revenue streams but also market share and long-term business sustainability. The ability to predict and mitigate churn is critical for telecom companies striving to retain their customer base in an increasingly saturated market. High churn rates, which can range from 20% to 40% annually, represent significant financial risks for telecom companies, as the cost of acquiring new customers is much higher than retaining existing ones (Saha et al., 2023).

Traditional approaches to predicting churn, such as customer surveys and basic statistical models, often fall short of capturing the complexity of customer behavior in today's data-driven environment. These methods are typically retrospective, relying on historical data without adequately accounting for the multifaceted factors that contribute to customer dissatisfaction and attrition. Additionally, these methods are often limited in their capacity to discern causal relationships between different variables, which restricts the effectiveness of the churn prediction models and the subsequent retention strategies. Many previous studies have identified retention rate's substantial effect on the market, but clients can always churn away from a business, resulting in potential losses for the organization. However, customers usually offer some warning before being churned. Hence, churn prediction systems primarily focus on customer behavior to identify specific customers who are likely to churn out and indicate reasons for the churn. Such factors would aid marketing to develop effective retention strategies, increasing overall customer lifetime value, and assisting in growing the company's market value (Hason Rudd et al., 2022). The need for more sophisticated models that go beyond correlations to uncover causality is becoming increasingly evident in the literature. This research aims to address these limitations by proposing a model that using Causal AI techniques to discover the true causes of churn.

Also, the focus of this study is not only to improve the predictive accuracy of churn models but also to identify actionable insights that can inform retention strategies. The model aims to provide a comprehensive approach to churn management, offering both predictive power and explanatory insights that can help telecom operators implement targeted interventions.

1.2 Scope of the research

This research aims to conduct a thorough analysis of customer churn within the telecommunications sector, where retaining customers is particularly difficult due to intense competition and evolving consumer expectations. The study explores several key areas that are vital for understanding and tackling the churn phenomenon. Central to this analysis is the examination of extensive customer data provided by telecommunications companies, encompassing various attributes such as demographic information, service usage patterns, billing histories, and customer support interactions. Through the analysis of these datasets, the research seeks to identify trends and behavioral patterns that are most closely associated with customer churn.

A core component of the research involves identifying specific churn-related factors that significantly influence customers' decisions to discontinue their services. These factors span a variety of variables, including the length of contracts, the quality of services provided, levels of customer satisfaction, pricing structures, and external pressures from competing market players. Understanding how these variables contribute to customer dissatisfaction is critical for developing strategies that can effectively reduce churn and improve customer retention.

The study will incorporate Causal AI techniques to perform a thorough causal analysis, which goes beyond merely identifying correlations between variables to reveal the underlying cause-and-effect relationships that drive customer churn. This approach offers a deeper insight into the factors that directly influence customer attrition, equipping telecom companies with more precise knowledge for crafting retention strategies that address the root causes of churn. By differentiating between superficial correlations and genuine causal factors, the research aims to provide actionable intelligence that telecom operators can leverage to develop more effective retention policies.

While the research primarily focuses on telecommunications companies operating in highly competitive markets where customer retention poses significant challenges, the methodologies proposed in this study are expected to be broadly applicable. The use of causal analysis through Causal AI can be adapted to other industries facing similar customer retention difficulties, such as banking, retail, and subscription-based services. These sectors, much like telecommunications, experience notable levels of customer churn and stand to benefit from the predictive and explanatory models developed in this research to enhance their retention strategies.

1.3 Research methods

The research methodology for this study is structured in several stages, each designed to facilitate the development, testing, and validation of the proposed model integrating Causal AI techniques.

1.3.1 Data collection

The initial phase of the research involves the collection of comprehensive datasets from telecommunications companies. These datasets will encompass a wide range of variables, including customer demographics, service usage metrics, billing histories, and customer service interactions. Additionally, the datasets will contain churn indicators, enabling the model to accurately predict customer attrition. The diversity and depth of the data are crucial for ensuring the robustness and reliability of the churn prediction model.

1.3.2 Causal AI analysis

Causal AI techniques will be employed to identify the causal relationships between various factors that influence customer churn. Unlike traditional methods that rely solely on correlation, Causal AI provides a more sophisticated analysis by uncovering the underlying cause-and-effect relationships. This deeper understanding of the drivers of churn will offer telecom companies actionable insights, enabling them to focus on addressing the most critical factors that contribute to customer attrition.

1.3.3 Model validation

Once developed, the hybrid model will be rigorously validated using real-world data from telecom customers. To assess its predictive accuracy, performance metrics such as precision, recall, and the F1 score will be employed. These metrics will evaluate the model's effectiveness not only in predicting churn but also in accurately identifying the key drivers behind customer attrition. The validation process is essential for ensuring the practical applicability of the model in real-world scenarios.

1.3.4 Recommendations for action

Based on the outcomes generated by the hybrid model, the research will provide strategic recommendations for telecommunications companies. These recommendations will focus on actionable retention strategies tailored to specific customer segments identified through the model. The recommendations will be grounded in the causal relationships uncovered by the Causal AI analysis, ensuring that the proposed interventions are both targeted and evidence-based, thus enhancing their effectiveness in reducing customer churn.

1.4 Research objectives

The primary objective of this research is to develop and validate a model that leverages the strengths of Causal AI techniques to enhance customer churn prediction in the telecommunications sector. The focus will be on improving the accuracy of churn predictions by identifying the causal factors contributing to customer attrition. By employing Causal AI techniques, this study seeks to move beyond correlation-based analyses and provide insights into the underlying causes of customer churn. Understanding these mechanisms is crucial for creating targeted and effective interventions. Furthermore, the research intends to deliver actionable retention strategies derived from the model's findings. By addressing the specific factors that lead to customer dissatisfaction, telecom companies can implement more precise and impactful retention initiatives. These strategies aim to reduce churn by tackling the root issues that drive customers to leave, ultimately enhancing customer satisfaction and loyalty.

CHAPTER 2: LITERATURE REVIEW

2.1 Theoretical background

2.1.1 Introduction to Causal AI

Causal AI represents a significant advancement in artificial intelligence, focusing on understanding and modeling cause-and-effect relationships rather than mere correlations. This approach enhances decision-making across various sectors, including manufacturing, supply chain management, and even hurricane prediction, where traditional methods often fail to capture essential causal dynamics (Durvasula, 2024) (Latos et al., 2024). Causal AI's ability to conduct what-if simulations allows organizations to explore potential outcomes and optimize strategies, thereby mitigating risks and seizing opportunities (Durvasula, 2024). Furthermore, it addresses limitations in explainable AI (XAI) by providing insights into causal relationships, which is crucial in high-stakes applications like medical diagnostics and fraud detection(Rawal et al., 2024) (Parkar et al., 2024). The integration of causal knowledge with neurosymbolic methods further enriches AI systems, enabling better decision-making in complex scenarios (Jaimini et al., 2024). Overall, Causal AI's emphasis on causality enhances the interpretability and reliability of AI systems, making it a vital area of research and application in contemporary AI development.

2.1.2 Explainable AI (XAI)

Explainable Artificial Intelligence (EAI) has emerged as a pivotal component in enhancing transparency and trust in AI applications, particularly within healthcare. EAI addresses the inherent complexity and opacity of AI models, which can hinder their adoption in critical decision-making scenarios, such as medical diagnosis(Ayesha & Ahamed, 2024) (Thakur, 2024). By employing various methodologies, including rule-based systems and feature importance analysis, EAI not only improves diagnostic accuracy but also provides insights into the decision-making processes of AI systems(Ayesha & Ahamed, 2024) (Kumar et al., 2024). This transparency is crucial for fostering trust among healthcare professionals and patients, as it allows for better understanding and validation of AI-driven recommendations(Thakur, 2024) (Kostopoulos et al., 2024). Furthermore, the evaluation of EAI systems emphasizes interpretability, robustness, and ethical considerations, ensuring that these technologies are not only effective but also fair and user-friendly("Evaluation of Explainable Artificial Intelligence using TOPSIS Method", 2024) (Kostopoulos et al., 2024). As the integration of AI in healthcare continues to evolve, EAI stands to significantly enhance patient care and clinical outcomes by bridging the gap between advanced technology and human expertise(Ayesha & Ahamed, 2024) (Kumar et al., 2024).

2.1.3. Artificial Neural Networks (ANN)

A. ANN architecture

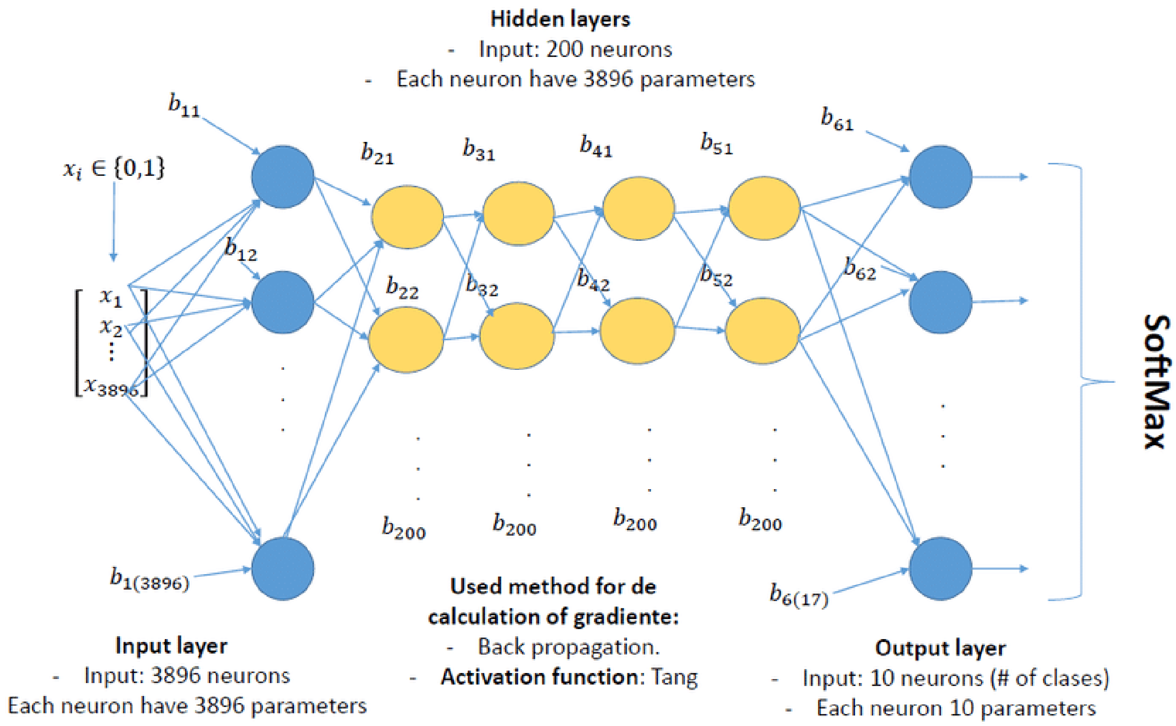

Figure 2.1: The display of An Artificial Neural Network

An Artificial Neural Network (ANN) represents a computational paradigm inspired by the neural structures observed in biological systems. This model typically comprises an input layer, one or more hidden layers, and an output layer. Each neuron in the network is associated with a set of parameters (weights) that transform input data through multiple layers, enabling the network to "learn" by iteratively adjusting these weights to minimize prediction error. The learning process often relies on backpropagation, a gradient-based optimization technique, which allows the network to fine-tune parameters and improve accuracy progressively. To introduce non-linearity and increase learning capacity, activation functions are applied at each neuron, facilitating the discovery of complex relationships in the data. For multi-class classification tasks, the output layer typically uses the SoftMax function to generate a probability distribution, which aids in assigning data points to the most likely classes.

According to Ceballos et al. (2019), this ANN architecture is specifically structured for a 10-class classification problem, comprising an input layer with binary features, multiple hidden layers, and a SoftMax-activated output layer. The architecture is as follows:

Input Layer:

- Neurons and Structure: The input layer consists of 3896 neurons, each corresponding to an individual feature in the dataset. The binary nature of the input (xi ∈{0,1}x i ∈{0,1}) suggests that each neuron represents the presence or absence of specific attributes.
- Parameterization: Each neuron in the input layer is associated with 3896 parameters, potentially representing distinct weight values connected to neurons in the succeeding hidden layers.

Hidden Layers:

- Layer Composition: This network includes multiple hidden layers, each containing 200 neurons. The substantial number of neurons per layer suggests that the architecture is designed to capture intricate patterns within the data.
- Parameters and Connectivity: Each neuron within these hidden layers has 3896 parameters, indicating a high degree of connectivity and a significant capacity for representing complex relationships among input features.
- Learning and Activation Mechanism: The model employs backpropagation for gradient computation, which allows for systematic weight adjustment based on error minimization, enhancing the model’s predictive accuracy over time. The activation function used is the "Tang" function, presumably referring to the hyperbolic tangent (tanh) function. This activation function introduces non-linearity by mapping values

within the range of -1 to 1, thereby improving the network's ability to learn from diverse input distributions.

Output Layer:

- Neuron Configuration: The output layer consists of 10 neurons, each representing one class in a 10-class classification task.
- Parameterization: Each neuron within the output layer has 10 parameters, which are likely connected to each neuron in the last hidden layer.
- SoftMax Activation: The SoftMax function is applied at the output layer to convert the final layer's scores into probabilities across all classes. This approach facilitates multi-class classification by generating confidence scores, which enable the model to assign the most probable class to each input instance.

The complexity of this ANN (Ceballos et al.,2019), marked by a high neuron count and numerous parameters across each layer, implies its suitability for applications involving extensive datasets and sophisticated feature interactions. The inclusion of multiple hidden layers with backpropagation and the tanh activation function enables this model to learn complex, non-linear representations, while the SoftMax layer at the output allows for effective probability-based classification across multiple categories.

B. Learning process in ANNs

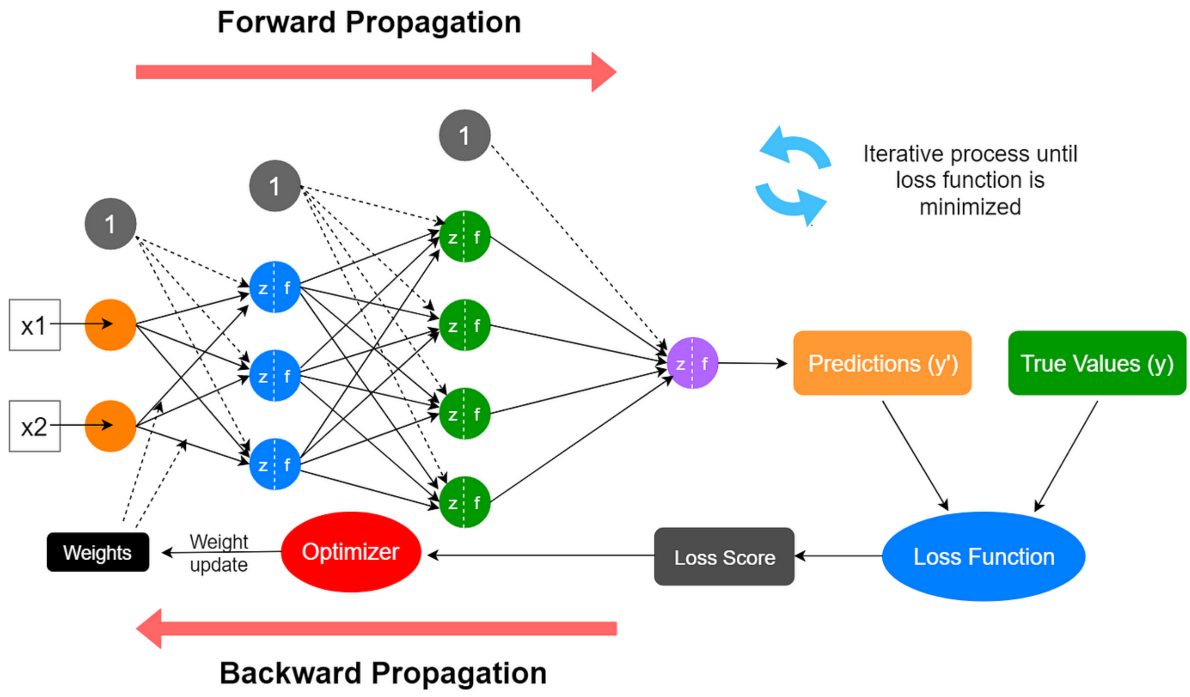

Figure 2.2: The display of Forward Propagation and Backward Propagation of ANN model

According to Data Science 365, the learning process of an Artificial Neural Network (ANN) consists of several key phases, systematically iterated to optimize the model's predictive performance. This process, visualized in the provided image, involves forward propagation, loss calculation, backward propagation, and optimization:

- Forward Propagation: In the forward propagation phase, input data (e.g. features $x1$ and $x2$) is introduced to the network and subsequently propagated through its layers. Each input is associated with specific weights and bias values, which are multiplied with the input data as they progress through each neuron in the network. Each neuron calculates a weighted sum of its inputs and applies an activation function (represented as $z = f(x)$ to introduce non-linearity, a crucial property that allows the network to model complex patterns. The activated outputs are then forwarded to subsequent layers until they reach the output layer. At the output layer, the network produces predictions (y') based on the processed inputs, which will be compared against the actual target values.

- Loss Calculation: Following forward propagation, a loss function is employed to quantify the discrepancy between the network’s predictions (y') and the true target values (y). This discrepancy is represented as a “loss score,” which provides a measure of the model’s prediction error. A lower loss score indicates that the model’s predictions are closer to the actual values, while a higher score reflects greater inaccuracy.

- Backward Propagation: Using the loss score, the ANN initiates backward propagation, a process wherein the network’s weights are systematically adjusted to reduce prediction error. Backpropagation involves computing the gradients (partial derivatives) of the loss function with respect to each weight in the network, which reveals the contribution of each weight to the total error. An optimizer, such as gradient descent, uses these gradients to update the weights by moving them in a direction that minimizes the loss function. This weight adjustment process is critical for enabling the network to learn from its mistakes and improve its predictive accuracy over time.

- Iteration and Optimization: The entire process—comprising forward propagation, loss calculation, backward propagation, and weight updating—repeats iteratively across multiple training cycles or epochs. The iterative optimization enables the network to continually adjust its weights to minimize the loss function, thereby refining its predictions. This process continues until the loss function is minimized to a satisfactory threshold, indicating that the network has achieved a high degree of accuracy in its predictions. At this stage, the ANN is considered to have learned the underlying patterns in the input data effectively.

C. Applications and limitations of ANNs

Artificial Neural Networks (ANNs) have been applied in various fields due to their ability to model complex, nonlinear patterns and relationships. However, each application comes with notable limitations that impact the model's effectiveness and practicality. Below are some of the applications and corresponding limitations in each field:

Agriculture and Food Quality Management

ANNs have proven effective in applications such as soil analysis, crop management, and quality assessment in agricultural products. ANN configurations like Multilayer Perceptrons (MLPs) and Radial Basis Function (RBF) networks are commonly used for classification, prediction, and modeling tasks in this domain (Huang, 2009). However, ANNs in agriculture face significant challenges related to transparency due to the model’s “black box” nature. This makes it difficult to interpret and understand the decision-making process, particularly important in quality analysis for food products. Additionally, ANNs often require substantial computational resources and are prone to overfitting, which limits their performance when applied to real-world data (Huang, 2009).

Biological Engineering

In biological engineering, ANNs are employed to analyze complex biological patterns, improve efficiency in production processes, and control quality. Thanks to their ability to learn nonlinear patterns, ANNs aid in optimizing processes and enhancing prediction accuracy in biological applications (Huang, 2009). As in agriculture, the “black box” nature of ANNs is a major limitation in biological engineering, where researchers and engineers need a deeper understanding of the model's workings. Additionally, the high computational demand and long training times make ANNs challenging to implement in processes requiring quick processing speeds (Huang, 2009).

Anesthesiology

ANNs are widely applied in anesthesiology to support monitoring the depth of anesthesia, predicting events and risks, and assisting with ultrasound guidance. With their ability to analyze extensive, complex physiological data streams, ANNs improve diagnostic accuracy and operational efficiency in anesthesiology (Hashimoto et al., 2020). However, in this field, the “black box” nature of ANNs poses a significant challenge in interpreting results, an essential factor in healthcare where transparency and explainability are crucial. Additionally, ANNs are susceptible to data bias and require large datasets, which limits their applicability across diverse patient populations and medical scenarios (Hashimoto et al., 2020).

Geotechnical Engineering

In geotechnical engineering, ANNs are applied to tasks like predicting pile capacity, modeling soil behavior, assessing slope stability, and characterizing sites. ANNs have shown higher predictive accuracy than traditional methods, allowing engineers to make reliable assessments of soil settlement and load-bearing capacity (Shahin et al., 2001). However, the “black box” nature of ANNs is also a limitation in geotechnical engineering, where engineers need to understand how predictions are made, especially when decisions directly impact structural safety. Moreover, ANNs in geotechnics require large, high-quality datasets for training, and model performance may suffer if data is insufficient or non-representative (Shahin et al., 2001).

Environmental and Climate Modeling

ANNs have been extensively applied in environmental science and climate modeling, especially for predicting weather patterns, climate changes, and environmental hazards. Their ability to model complex nonlinear relationships is essential in understanding and forecasting climatic conditions based on various atmospheric variables. For instance, ANNs have been employed in areas such as air quality prediction, rainfall estimation, and predicting temperature changes, which assist in better understanding and mitigating climate-related risks. One of the primary challenges of using ANNs in environmental modeling is their reliance on large datasets, which may not always be available or consistent in quality, especially in remote or under-monitored areas. Furthermore, the "black box" nature of ANNs makes it difficult to interpret results, which is crucial in environmental applications where transparency and model verification are essential for policy-making. Also, the potential sensitivity of ANNs to noise in environmental data can impact the accuracy and reliability of predictions.

2.1.4. Deep Neural Networks (DNN)

A. Explanation of DNN architecture

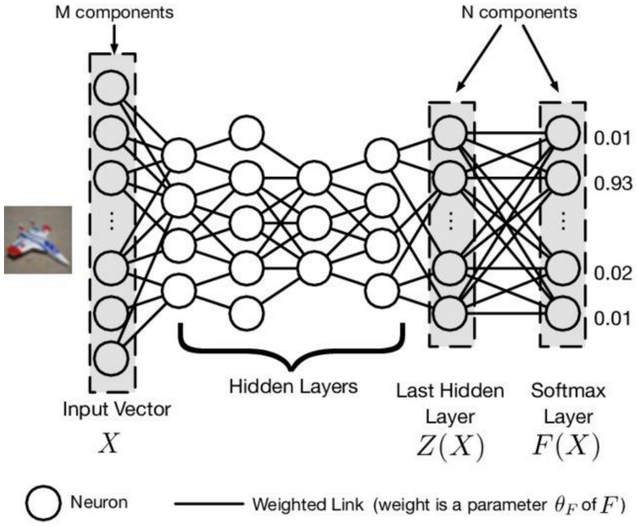

Figure 2.3: The display of a Deep Neural Network

According to the illustration provided and referencing the architecture described in ResearchGate (Papernot et al., 2016), this Deep Neural Network (DNN) architecture represents a structured approach for processing and classifying data through a series of interconnected layers, from an input vector to a final SoftMax layer. The architecture includes several primary components as follows:

Input Layer: The input layer comprises M components, each corresponding to an individual feature or attribute of the input data. In image processing, for example, these components may represent pixel intensities. The input vector X serves as the initial data input for the network, which is then propagated through multiple layers of neurons to extract and learn essential features.

Hidden Layers: The DNN includes several hidden layers, each composed of neurons that capture increasingly complex patterns in the data. These layers are critical in enabling the model to learn hierarchical representations, allowing it to understand intricate, non-linear relationships within the input. Each neuron within a hidden layer is connected to neurons in adjacent layers via weighted links. These weights are denoted by θF\theta_FθF, representing the parameters of the function FFF that the network seeks to optimize. During training, these weights are iteratively updated through backpropagation, enabling the network to reduce error and improve its performance on classification tasks.

Last Hidden Layer (Z(X)): The final hidden layer, labeled as Z(X), aggregates the learned features from the previous layers, forming a high-level representation of the input data. This layer encapsulates essential information, which is subsequently passed to the output layer for classification. It serves as a preparatory stage, synthesizing features necessary for accurate decision-making in the next layer.

SoftMax Layer (F(X)): The output layer employs a SoftMax function, which converts the model's output scores into probability distributions across various classes. This transformation allows for multi-class classification, where each output value represents the network's confidence in a particular class. Probabilities are assigned values between 0 and 1 (e.g., 0.01, 0.93, 0.02, and 0.01), indicating the likelihood that the input belongs to each respective class. The class with the highest probability is typically chosen as the network's predicted output.

Weighted Links: Connections between neurons are represented by weighted links that determine the strength of each neuron's influence on others. During training, the network adjusts these weights to minimize the loss function, thereby enhancing classification accuracy. This architecture is also potentially optimized through distillation, as noted by Papernot et al. (2016), a technique aimed at bolstering the network's resilience to adversarial attacks. Distillation helps in refining the weights, providing an additional layer of robustness against small, intentionally crafted perturbations in the input. This DNN architecture, comprising multiple hidden layers and a SoftMax layer, is structured to enable effective feature extraction and robust classification, particularly for tasks that demand high-level abstraction and resilience to adversarial interference (Papernot et al., 2016).

B. Differences between DNNs and traditional ANNs

Table 2.1: Description of differences between DNNs and traditional ANNs

Criteria	**Artificial Neural Networks (ANN)**	**Deep Neural Networks (DNN)**
Architecture	ANN typically consists of an input layer, a single hidden layer, and an output layer. It has a relatively simple structure, making it suitable for less complex tasks.	DNN features multiple hidden layers, forming a "deep" architecture. This allows it to handle more complex data transformations and extract deeper patterns.
Complexity	The design of ANN is less complex, often employing fewer neurons and layers, which reduces computational requirements	DNNs are inherently more complex due to their depth, often utilizing advanced architectures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for specific tasks.
Learning Capacity	ANN is effective for tasks with structured data and simple relationships. Its capacity to generalize is limited due to fewer layers and neurons.	DNN has superior learning capacity, making it capable of handling unstructured data like images, text, and audio. The depth enables feature extraction from raw input data without manual intervention.

Training Requirements	ANN requires less computational power and memory, making it faster to train with smaller datasets. Backpropagation is simpler and faster due to fewer layers.	DNN demands significant computational resources and large datasets for effective training. The increased depth requires more time and sophisticated optimization techniques for backpropagation.
Performance	ANN performs well on simpler problems and tasks where the data patterns are not complex. It is often used in predictive modeling and simple classification tasks.	DNN outperforms ANN in tasks involving complex data representations, such as image recognition, natural language processing (NLP), and autonomous driving. Its layered structure allows for deeper analysis and feature abstraction.

C. Application of DNNs

In real-time voice conversion, DNN-based methods have shown great promise by enabling high-quality speech transformation. Arakawa et al. (2019) introduced a DNN-based real-time voice conversion (VC) system that operates with low latency, designed to alter speech characteristics while retaining linguistic information, making it suitable for augmented communication applications. Their proposed system consists of three stages: feature extraction, conversion, and synthesis, achieving real-time conversion with an overall latency of 50 ms. To improve robustness, they applied data augmentation techniques such as pitch shift, time stretch, and time shift, which enhanced speech quality and adaptability to variations in pitch and timing. Additionally, they developed a mask-shaped device to block background noise, offering resilience in practical settings. Experiments confirmed that these innovations significantly improve both speech quality and noise resistance, demonstrating the system’s effectiveness in dynamic environments.

In robust speaker recognition, DNN-based autoencoders have shown substantial improvements by enhancing speech through noise reduction and dereverberation. Novotny et al. (2018) trained an autoencoder on the Fisher dataset, incorporating noise and reverberation to simulate diverse real-world environments. Their method used this preprocessing to improve speaker recognition in challenging acoustic conditions, including the NIST SRE 2010 and PRISM datasets. The autoencoder effectively mapped noisy speech to cleaner audio, boosting recognition accuracy. They demonstrated that combining autoencoder preprocessing with PLDA multi-condition training offers the most robust performance, particularly in noisy and reverberated settings, significantly outperforming baseline models in speaker recognition accuracy.

2.1.5. XGBoost

A. Introduction to XGBoost and its features

XGBoost, short for Extreme Gradient Boosting, is a sophisticated implementation of the gradient boosting algorithm tailored for efficiency and speed. It is designed to provide scalable and accurate predictions for supervised learning tasks, including both classification and regression. The core idea of XGBoost revolves around creating an ensemble of decision trees where each new tree attempts to correct the errors made by its predecessors. By leveraging gradient descent optimization, XGBoost minimizes a predefined loss function iteratively, thereby refining the predictive power of the model at each step. Unlike traditional gradient boosting methods, XGBoost incorporates numerous system and algorithmic enhancements, such as regularization, parallel processing, and efficient memory usage, which together make it one of the most powerful tools in the field of machine learning.

The XGBoost algorithm operates through a process of sequential learning, where multiple decision trees are built in stages to progressively reduce the model's residual error. Initially, the dataset is divided into smaller subsets (denoted as D1,D2,…,Dn), and an initial prediction model is trained using a simple decision tree. This initial tree forms a baseline prediction for the target variable. Subsequently, the algorithm computes the residuals, which represent the difference between the actual and predicted values. These residuals are crucial because they quantify the errors made by the initial model, and serve as the target for the next tree to minimize. Each subsequent tree is trained specifically to predict these residuals, thus correcting the errors of the preceding model. This iterative process continues, with each tree refining the model by focusing on the residuals, which effectively acts as a feedback mechanism. The use of gradient descent optimization at each stage allows the algorithm to adjust the model parameters in a direction that reduces the overall loss function, thereby improving predictive accuracy. The final model output is the aggregated sum of the predictions from all individual trees.

In XGBoost, decision trees are constructed in a manner distinct from traditional tree-based models. Each decision tree is built to predict the residuals or errors generated by the previous ensemble of trees, rather than directly attempting to model the target variable. This approach allows the algorithm to iteratively refine its predictions by focusing on the most challenging aspects of the data where previous models performed poorly. The construction of each tree is guided by a loss function, typically a combination of a differentiable convex loss function (e.g., mean squared error for regression tasks) and a regularization term that penalizes overly complex trees. This combination helps strike a balance between fitting the data well and avoiding overfitting.

The trees in XGBoost are typically shallow and wide, meaning they have a limited depth but a broader range of splits, which contributes to faster computation and better generalization. After constructing the trees, the algorithm combines their predictions through an additive process, where each tree’s output is summed to form the final prediction. The use of additive learning, where new trees are added sequentially and each tree corrects the errors of the ensemble, ensures that the model converges to a highly accurate solution.

B. Applications of XGBoost in machine learning

In air quality prediction, XGBoost has proven to be a powerful tool, particularly for forecasting PM2.5 concentrations, where it outperforms traditional numerical models. For instance, the modified XGBoost model has shown higher accuracy than the WRF-Chem model by effectively integrating real-time pollutant measurements, such as

PM2.5, SO2, and O3, alongside meteorological data, including temperature, humidity, and wind conditions. These enhancements address common limitations in traditional models, like inaccuracies due to biased emission data and boundary conditions. However, the XGBoost model still requires large, high-quality datasets for optimal performance, which can be a limitation in regions with sparse or inconsistent data availability (Ma et al., 2020). Furthermore, while XGBoost improves prediction reliability, it remains sensitive to variations in input data quality, which can impact forecast precision under fluctuating environmental conditions.

In handling imbalanced data classification, XGBoost has emerged as a powerful solution, especially when applied to cases like fraud detection and medical diagnosis, where minority classes are often underrepresented. For example, an optimized XGBoost model (SEB-XGB) demonstrated superior accuracy by combining SVM–SMOTE for over-sampling minority classes and EasyEnsemble for under-sampling majority classes. This model also utilized Bayesian optimization to fine-tune parameters, enhancing its predictive performance on datasets with extreme class imbalance, such as credit card and fraud detection data. Despite its improved classification accuracy, the SEB-XGB model requires large and well-prepared datasets to maximize effectiveness, posing a challenge in scenarios with limited data (Zhang et al., 2022). Additionally, while SEB-XGB enhances the identification of minority classes, its sensitivity to sampling and optimization techniques can affect consistency across different types of imbalanced datasets.

In network traffic classification, XGBoost has proven highly effective, particularly in handling encrypted and complex traffic within home networks. This study applied XGBoost to a large dataset of over 350,000 flows from a major French ISP, encompassing traffic types like HTTP, HTTPS, Skype, and BitTorrent. By leveraging 22 features, including packet size and inter-packet delay, XGBoost achieved a classification accuracy of 99.5%, outperforming traditional methods that rely on protocol-based analysis or port numbers, which often fail with encrypted traffic. Extensive parameter tuning, through grid search, contributed to the model's accuracy, making it superior to other machine learning algorithms, such as C5.0 and K-NN.

However, this high precision comes at a cost: XGBoost requires significant computational resources, leading to large model sizes that may restrict its application in real-time or resource-limited environments.

2.2 Research status

2.2.1 Research overview

The modern telecommunications industry has become a crucial pillar in the digital economy, contributing to socio-economic development by providing essential services such as internet access, mobile telephony, and pay-TV. According to the 2023 report by the Office of Electronic Communications (UKE), the global telecommunications market experienced substantial revenue growth, reaching PLN 43.1 billion, an increase of 6% over the previous year, with investments in infrastructure amounting to PLN 53.5 billion. Despite this growth, the industry faces significant challenges in customer retention, especially in the context of rising competition and the diversity of service packages. An essential indicator to assess the effectiveness of telecommunications companies is the churn rate – the percentage of customers who stop using a product or service within a certain period. This metric has a major impact on company revenue and profitability, as the cost of acquiring new customers is generally 5 to 20 times higher than retaining existing ones. Reducing the churn rate has thus become a strategic priority for telecommunications companies to maintain a competitive edge and optimize profitability (Barsotti, 2024).

In telecommunications, predicting customer churn has long been an essential task to help companies optimize their retention strategies. Various traditional methods have been implemented, with regression models and machine learning models being the most prominent:

- Regression Models: Models such as Logistic Regression and Linear Regression are often applied to analyze churn data, as they can predict based on the linear relationship between influencing factors and the probability of customer churn. Logistic Regression is a popular choice for its interpretability and ability to identify major factors affecting churn. However, due to its simple structure, regression models often lack accuracy when applied to complex datasets (Barsotti, 2024).

- Machine Learning Models: To enhance accuracy, telecommunications companies have applied advanced machine learning algorithms such as Random Forest, Decision Trees, and Support Vector Machines (SVM). These algorithms can classify and predict churn based on nonlinear data features, generating more accurate predictions than linear regression. However, a limitation of machine learning models is their "black-box" nature, making them difficult to interpret, which hinders understanding the underlying causes of churn and complicates the development of specific actions to reduce churn (Office of Electronic Communications [UKE], 2024).

The advancement of data analysis techniques has led to the integration of K-means clustering and Causal AI as prominent technologies in churn prediction within the telecommunications industry. K-means clustering, a widely used method for data segmentation, enables telecommunications companies to classify customers into distinct groups based on behavioral and demographic characteristics. This segmentation approach allows firms to identify customer segments with a high risk of churn, thereby enabling a targeted focus on retention and marketing strategies for the most vulnerable groups (Barsotti, 2024). Meanwhile, Causal AI provides telecommunications companies with tools to not only predict which customers are likely to churn but also to understand the underlying reasons for their departure. By analyzing causal relationships between various factors and customer behavior, Causal AI identifies both direct and indirect drivers of churn. This form of causal analysis utilizes causal graphs and Structural Causal Models (SCM) to offer companies deeper insights into how individual factors influence customer churn, facilitating more informed decision-making. When combined, K-means clustering and Causal AI create a robust system for churn prediction that not only highlights high-risk customer groups but also elucidates the underlying causes of churn. This integration empowers telecommunications firms to design more effective and targeted retention strategies, ultimately optimizing costs and improving customer retention rates (Office of Electronic Communications [UKE], 2024).

2.2.2 Recent researchs related to Kmeans and Causal AI in telecommunications industry

In the telecommunications industry, the development of causal analysis methods (Causal AI) has opened up potential for models that not only predict churn accurately but also provide a deeper understanding of the reasons behind customer behavior. Recently, Verhelst (2019) proposed a detailed causal analysis framework using customer data from Orange Belgium. In this study, factors such as inappropriate tariff plans and "bill shock" were identified as primary drivers of churn. The application of causal Bayesian networks and inference methods allowed for a comprehensive examination of the interactions between variables in customer data, providing interpretive insights that support strategic decisions in marketing and customer service. Additionally, a study by Zhang et al. (2022) introduced a counterfactual approach to churn prediction, focusing on social influence—a major driver behind customer churn behavior. The research group's counterfactual model, CFChurn, was designed based on graph neural networks, separately modeling endogenous churn intentions and exogenous social influences. By incorporating counterfactual information, this model achieves more accurate predictions and provides a nuanced understanding of social factors impacting churn. Tests on large datasets demonstrated that CFChurn not only performs effectively but also generates predictions with clear explanations, particularly beneficial for social network-based customer retention campaigns. Another noteworthy study by Shah et al. (2019) applied deep learning techniques and an artificial neural network (ANN) for causal analysis in churn prediction. Shah et al. leveraged big data to identify influencing factors and used specialized techniques to address imbalanced data—an ongoing challenge in churn prediction. The ANN model in this study not only improved prediction accuracy but also yielded insightful analyses of the reasons behind churn, thereby supporting more effective customer retention strategies. These studies collectively contribute to building churn prediction models that can explain underlying causes, paving the way for a new direction in the application of causal analysis in telecommunications. However, there remain certain limitations and challenges in applying Causal AI practically, which require continued optimization in future research.

While Causal AI models in churn prediction have achieved encouraging results, some limitations persist, affecting their effectiveness and applicability in real-world settings. Firstly, most studies rely on correlations rather than exploring true causal relationships, making it difficult for traditional models to produce highly interpretable conclusions about the exact reasons for churn. According to Verhelst (2019), one of the biggest challenges in churn prediction research is the lack of counterfactual data, which limits understanding of the impact of different factors on churn. Counterfactual data is necessary to determine whether changing a variable would result in a difference in customer churn behavior. Additionally, simplified models like logistic regression and other machine learning methods fail to incorporate causal factors, potentially leading to misinterpretations or less insightful predictions. Zhang et al. (2022) noted that the lack of counterfactual data not only limits modeling accuracy but also affects interpretability. To address this, CFChurn was proposed to transform the counterfactual learning problem into a supervised learning problem with partially labeled counterfactual data. This solution allows CFChurn to achieve higher performance and deliver more accurate and interpretable predictions. By employing graph-based learning algorithms, CFChurn effectively leverages information from social networks to capture social influences, which helps identify specific social factors that cause churn. Lastly, Shah et al. (2019) emphasized that integrating deep learning models with causal analysis may be a promising direction for future research. By applying neural networks, the research team successfully isolated relevant factors and addressed data imbalance. This approach not only improved churn prediction accuracy but also enabled clearer analysis and interpretation of the causes while minimizing noise in the data. The study’s results illustrate the potential of using a combined deep learning and causal analysis model in churn prediction to optimize retention strategies.

These studies collectively contribute to building churn prediction models that can explain underlying causes, paving the way for a new direction in the application of causal analysis in telecommunications. However, there remain certain limitations and challenges in applying Causal AI practically, which require continued optimization in future research. To overcome current limitations and optimize these models, researchers could consider integrating different methods. For example, combining counterfactual data processing techniques from CFChurn with causal inference methods in Verhelst's Bayesian network could enhance model accuracy and provide better explanations of each factor's impact. Additionally, applying graph neural networks and deep learning methods to expand input data scope and explore social relationships and customer behavior may yield more accurate churn predictions. Furthermore, research could extend the counterfactual prediction approach by generating synthetic counterfactual data from observational data. This would not only reduce costs but also address counterfactual data limitations in situations where randomized experiments are not feasible. Technologies such as deep learning-based Causal AI, along with advancements in transfer learning methods, also promise to create more accurate and applicable churn prediction models in the future.

CHAPTER 3: DEVELOPING A MODEL TO PREDICT THE CHURN RATE

3.1 Research Framework

The proposed framework integrates machine learning, deep learning, and causal inference to establish a robust pipeline for solving analytical problems in various domains. It adopts a modular and systematic structure to ensure clarity, reproducibility, and adaptability across diverse datasets and use cases. By combining traditional exploratory analysis techniques with cutting-edge ML/DL algorithms and causal reasoning, the framework provides a comprehensive solution for both predictive and prescriptive analytics.

The process begins with defining the problem statement, which anchors the entire workflow to specific goals. This stage emphasizes clarity in identifying the questions to be addressed and the decisions to be informed by the analysis. Subsequently, data loading is conducted to prepare the dataset for analysis. This stage requires an understanding of the data's structure, format, and content, which are essential for determining appropriate preprocessing steps.

The preprocessing stage ensures the dataset is clean, consistent, and suitable for analysis. Handling missing values, standardizing data types, encoding categorical variables, and feature scaling are fundamental operations in this phase. These steps minimize noise and variability in the dataset, improving the performance and accuracy of downstream algorithms. Exploratory data analysis (EDA) plays a crucial role in uncovering patterns, trends, and relationships in the data. By visualizing and inspecting data structures, analysts can develop insights that inform the selection of features and modeling techniques. EDA also highlights potential biases, outliers, or anomalies, which might need to be addressed before proceeding to modeling.

In the machine learning and deep learning pipeline, the framework transitions from data understanding to model building. Data is partitioned into training and testing subsets to ensure that model performance generalizes to unseen data. Various algorithms, including gradient-boosted trees (e.g., XGBoost), artificial neural networks (ANN), and deep neural networks (DNN), are employed to generate predictive models. These techniques are chosen based on their ability to handle structured and unstructured data, capture complex nonlinear relationships, and provide scalability to large datasets. Model evaluation follows, using performance metrics such as accuracy, precision, recall, F1-score, and area under the curve (AUC), depending on the problem type. Comparative analysis of model results allows the selection of the best-performing algorithm. The results are further validated by comparing them with findings from relevant studies in the literature, ensuring that the insights derived are not only accurate but also aligned with domain-specific knowledge.

The framework also incorporates a causal discovery pipeline to identify causal relationships between variables. Unlike purely predictive approaches, this pipeline focuses on understanding how variables influence one another, guided by data-driven methods and, optionally, domain expertise. The causal relationships uncovered in this stage provide the foundation for the inference pipeline In the inference pipeline, causal models are used to simulate interventions and their impacts on outcomes. For example, fixing a specific variable (W) and observing the resulting changes in other variables (Y) allows for an assessment of causal effects. This approach addresses "what-if" questions, enabling actionable insights that support decision-making in real-world scenarios. The dual-pipeline structure discovery and inference ensures a balance between prediction and explanation. While the machine learning pipeline focuses on accurate forecasting, the causal pipelines aim to uncover and quantify underlying mechanisms, providing a deeper understanding of the data and its implications.

Finally, the results of the framework are evaluated holistically, ensuring alignment with existing research and practical considerations. This rigorous comparison enhances the reliability and applicability of the findings, making them more suitable for adoption in real-world applications. By integrating exploratory analysis, predictive modeling, and causal reasoning, the framework provides a robust, end-to-end solution for addressing complex analytical challenges in research and industry.

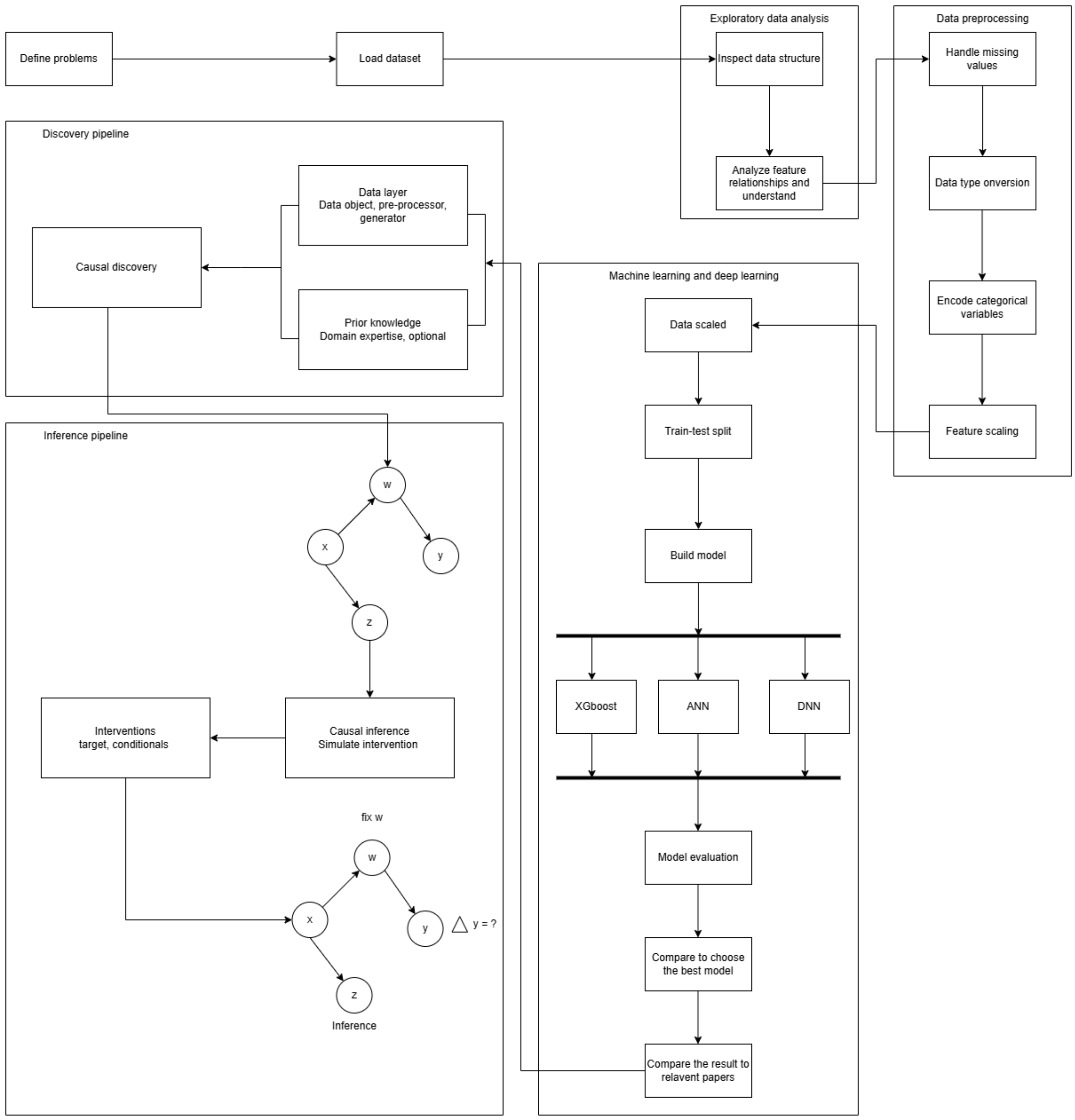

Figure 3.1: Research framework

3.2 Dataset information

3.2.1 Introduction to the dataset

Telco Customer Churn is a simulated dataset containing information about telecommunications customers, primarily intended for churn prediction and analysis. It captures various features influencing customer retention and includes a target variable, "Churn," which identifies whether a customer has left the company. This dataset comprises 7,043 rows and 21 columns, each row representing a unique customer. It encompasses details about the services customers have subscribed to, account information, and demographic attributes. The primary objective of analyzing this data is to develop a predictive model that can identify customers at risk of leaving, enabling targeted retention strategies.

- customerID: Identification of Customers
- gender: Whether the customer is a male or a female
- SeniorCitizen: Whether the customer is old or not (1|0)
- Partner: Whether the customer has a spouse or partner or not (Yes|No)
- Dependents: Whether the customer has children or other people they financially support (Yes|No)
- tenure: Number of months the customer has stayed with the company
- PhoneService: Whether the customer has a phone service or not (Yes|No)
- MultipleLines: Whether the customer has multiple lines or not (Yes|No|No phone service)
- InternetService: Customer's internet service provider (DSL|Fiber optic|No)
- OnlineSecurity: Whether the customer has online security or not (Yes|No|No internet service)
- OnlineBackup: Whether the customer has online backup or not (Yes|No|No internet service)
- DeviceProjection: Whether the customer has device protection or not (Yes|No|No internet service)
- TechSupport: Whether the customer has tech support or not (Yes|No|No internet service)
- StreamingTV: Whether the customer has streaming TV or not (Yes|No|No internet service)
- StreamingMovies: Whether the customer has streaming movies or not (Yes|No|No internet service)

● Contract: The contract term of the customer (Month-to-month|One year|Two year)

● PaperlessBilling: Whether the customer has paperless billing or not (Yes|No)

● PaymentMethod: The customer's payment method (Electronic check|Mailed check|Bank transfer (automatic)|Credit card)

● MonthlyCharges: The amount charged to the customer monthly

● TotalCharges: The total amount charged to the customer

● Churn: Whether the customer churned or not (Yes|No)

3.2.2 Data Exploration and Preprocessing

3.2.2.1. Data Exploration

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

Figure 3.2: The basic description of attributes in the dataset

The data.info() function provides detailed information about the structure of the DataFrame, including the number of columns, rows, data types, and the number of non-null values in each column. In this case, the DataFrame has a total of 21 columns and 7043 rows, with all columns containing 7043 non-null values, ensuring that there is no missing data.

Specifically, there are 18 columns of the object type, including the TotalCharges column; 2 columns of the int64 type, one of which is the SeniorCitizen column; and 1 column of the float64 type. Although there are no null values, the TotalCharges column currently has the object data type, whereas it should be of the float64 type to accurately reflect numerical data. On the other hand, the SeniorCitizen column is of the int64 type, but it may need to be converted to the object type if it is considered categorical data. Identifying and adjusting these data types is important to ensure accuracy and efficiency in the data analysis process.

data is complete, with no missing values, and there is no need to perform any data imputation steps.

	0
customerID	0
gender	0
SeniorCitizen	0
Partner	0
Dependents	0
tenure	0
PhoneService	0
MultipleLines	0
InternetService	0
OnlineSecurity	0
OnlineBackup	0
DeviceProtection	0
TechSupport	0
StreamingTV	0
StreamingMovies	0
Contract	0
PaperlessBilling	0
PaymentMethod	0
MonthlyCharges	0
TotalCharges	0
Churn	0

dtype: int64

Figure 3.3: The number of null values in each attribute

All columns in the DataFrame have a total of 0 missing values, indicating that the data is complete, with no missing values, and there is no need to perform any data imputation steps.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   SeniorCitizen     7043 non-null   object 
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            7043 non-null   int64  
 6   PhoneService      7043 non-null   object 
 7   MultipleLines     7043 non-null   object 
 8   InternetService   7043 non-null   object 
 9   OnlineSecurity    7043 non-null   object 
 10  OnlineBackup      7043 non-null   object 
 11  DeviceProtection  7043 non-null   object 
 12  TechSupport       7043 non-null   object 
 13  StreamingTV       7043 non-null   object 
 14  StreamingMovies   7043 non-null   object 
 15  Contract          7043 non-null   object 
 16  PaperlessBilling  7043 non-null   object 
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7032 non-null   float64
 20  Churn             7043 non-null   object 
dtypes: float64(2), int64(1), object(18)
memory usage: 1.1+ MB
```

Figure 3.4: The basic description of attributes after changing the data type of attributes having incorrect types

As mentioned, the TotalCharges and SeniorCitizen columns have been converted to the correct data types, namely float64 and object, respectively. This adjustment ensures that the data accurately reflects the real-world meaning of each column and supports more accurate and efficient data analysis and processing. However, after the data type conversion, the TotalCharges column now contains only 7032 values, which is fewer than the total number of rows in the DataFrame (7043). This indicates that 11 values are missing and need to be addressed in order to ensure the data is complete.

	tenure	MonthlyCharges	TotalCharges
count	7043.00	7043.00	7032.00
mean	32.37	64.76	2283.30
std	24.56	30.09	2266.77
min	0.00	18.25	18.80
25%	9.00	35.50	401.45
50%	29.00	70.35	1397.47
75%	55.00	89.85	3794.74
max	72.00	118.75	8684.80

Figure 3.5: The basic statistics of numeric attributes

The summary table provides detailed information about three columns: tenure, MonthlyCharges, and TotalCharges. Both tenure and MonthlyCharges have a full set of 7043 values, while TotalCharges contains only 7032 values, indicating that 11 values are missing. The average tenure of customers is 32.37 months, with a standard deviation of 24.56, and a maximum of up to 72 months. The average monthly charge is 64.76, ranging from 18.25 to 118.75, with a standard deviation of 30.09. The average total charge for customers is 2283.30, but it has a wide distribution with a standard deviation of 2266.77, ranging from 18.80 to 8684.80.

```
Label Encoder Transformation
customerID  :  [5375 3962 2564 ... 3367 5934 2226]  =  ['7590-VHVEG' '5575-GNVDE' '3668-QPYBK' ... '4801-JZAZL' '8361-LTMKD'
 '3186-AJIEK']
gender  :  [0 1]  =  ['Female' 'Male']
SeniorCitizen  :  [0 1]  =  [0 1]
Partner  :  [1 0]  =  ['Yes' 'No']
Dependents  :  [0 1]  =  ['No' 'Yes']
PhoneService  :  [0 1]  =  ['No' 'Yes']
MultipleLines  :  [1 0 2]  =  ['No phone service' 'No' 'Yes']
InternetService  :  [0 1 2]  =  ['DSL' 'Fiber optic' 'No']
OnlineSecurity  :  [0 2 1]  =  ['No' 'Yes' 'No internet service']
OnlineBackup  :  [2 0 1]  =  ['Yes' 'No' 'No internet service']
DeviceProtection  :  [0 2 1]  =  ['No' 'Yes' 'No internet service']
TechSupport  :  [0 2 1]  =  ['No' 'Yes' 'No internet service']
StreamingTV  :  [0 2 1]  =  ['No' 'Yes' 'No internet service']
StreamingMovies  :  [0 2 1]  =  ['No' 'Yes' 'No internet service']
Contract  :  [0 1 2]  =  ['Month-to-month' 'One year' 'Two year']
PaperlessBilling  :  [1 0]  =  ['Yes' 'No']
PaymentMethod  :  [2 3 0 1]  =  ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
 'Credit card (automatic)']
Churn  :  [0 1]  =  ['No' 'Yes']
```

Figure 3.6: The unique values of attributes before and after the labeling process

The Label Encoding process illustrated has converted all categorical data into numerical data, a necessary preprocessing step to make the data effectively usable in machine learning algorithms. Binary columns, such as Partner, Dependents, and Churn, were intuitively encoded with 'Yes' as 1 and 'No' as 0. For columns with special values, such as MultipleLines or those related to internet services (e.g., OnlineSecurity, StreamingTV), values like 'No phone service' or 'No internet service' were encoded separately to ensure they were not confused with standard 'No' values. This reflects careful attention to detail and the significance of each value during the encoding process.

Additionally, multi-category columns, such as Contract and PaymentMethod, were encoded into sequential numerical values, simplifying the handling of complex data. This process not only ensures that the data accurately represents its real-world meaning but also maintains consistency and logic in the conversion. The special handling of values like 'No phone service' or 'No internet service' demonstrates thorough preparation, ensuring these values are not misassigned and do not distort the machine learning model. This is an important step in transforming raw data into a format ready for analysis and modeling.

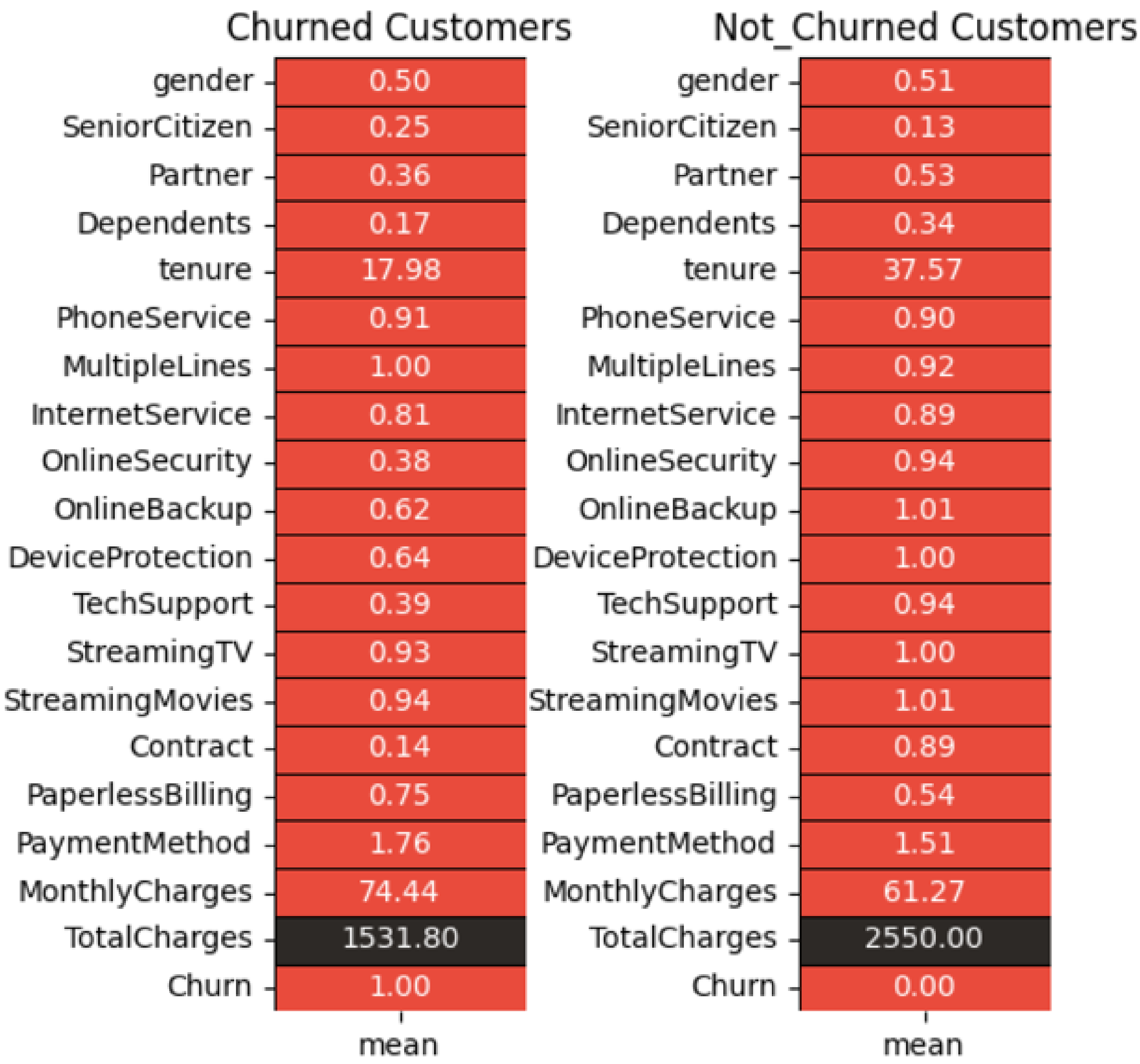

Figure 3.7: Average Value Comparison: Churned vs. Not Churned Customers

The table above compares the average values of variables between two customer groups: Churned Customers (those who have left the service) and Not Churned Customers (those who have remained). The churned customer group has a significantly lower average tenure (17.98 months) compared to the retained customer group (37.57 months), indicating that longer service duration increases customer loyalty. The average monthly charges for the churned group are significantly higher (74.44) compared to the retained group (61.27), while their average total charges are lower (1531.80 vs. 2550.00), reflecting their shorter usage duration.

Additionally, retained customers tend to make greater use of additional services such as OnlineSecurity, TechSupport, and DeviceProtection, whereas churned customers use these services less frequently. This suggests that offering value-added features may help retain customers.

Moreover, the proportion of customers using paperless billing (PaperlessBilling) is higher in the churned group (0.75 vs. 0.54), while the retained group is more likely to opt for long-term contracts, with a significantly higher average (0.89 vs. 0.14). Entertainment services like StreamingTV and StreamingMovies also have higher usage rates in the retained group. From these results, it can be observed that reasonable pricing, longer service durations, and offering value-added services are key factors in reducing churn and enhancing customer satisfaction.

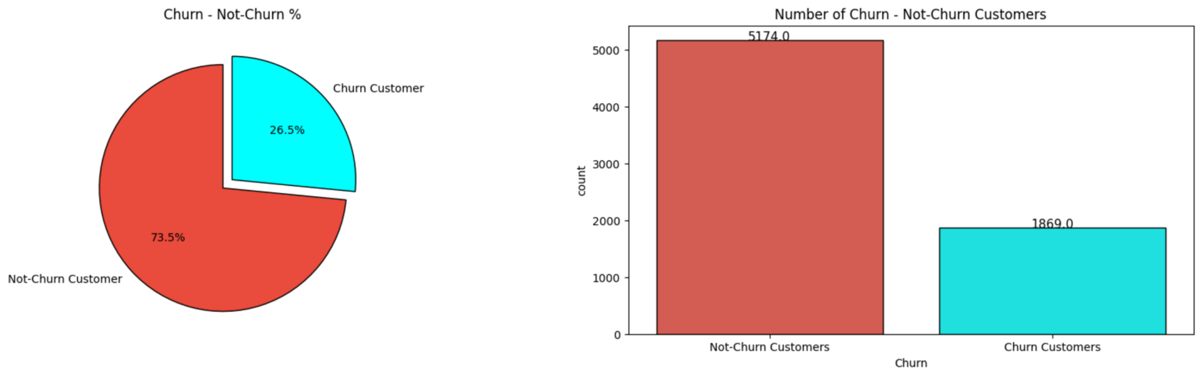

Figure 3.8: The Percentage and Number of Churn Customers compared to Not Churn Customers

The chart above illustrates the proportion and number of customers who churned (left the service) compared to those who retained the service (Not Churn). According to the pie chart, 26.5% of customers have churned, while 73.5% have remained. Although the majority of customers are loyal, a churn rate of more than a quarter of the total customer base is a noteworthy figure. The bar chart clearly highlights the discrepancy in numbers, with 5174 customers retaining the service compared to 1869 customers who have churned.

This gap underscores that, while loyal customers form the majority, losing nearly 2000 customers can have a significant impact on revenue. This necessitates that the business investigate the reasons behind customer churn and develop effective retention strategies, such as improving service quality or offering incentives to enhance the customer experience.

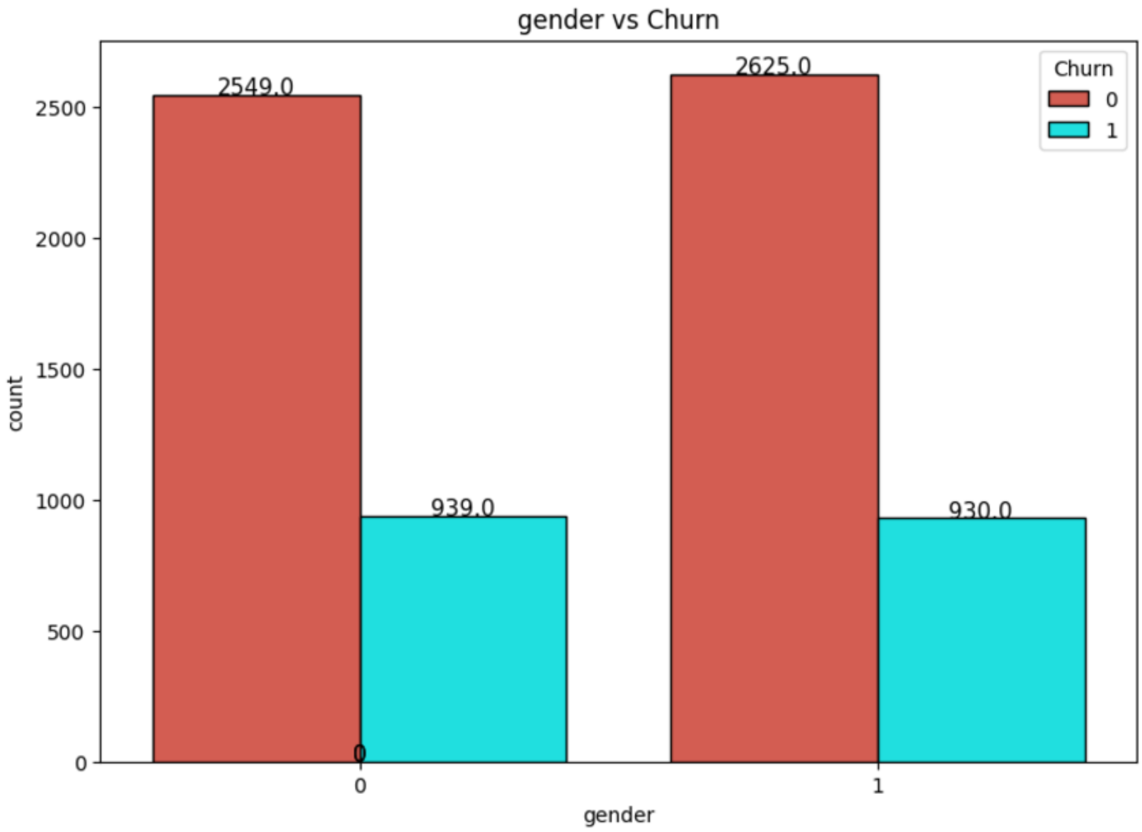

Figure 3.9: The Comparison of Churn and Not Churn Customers by Gender

Looking at the chart, the number of male and female customers who did not churn (Churn = 0) is nearly identical, with 2549 female customers and 2625 male customers. Similarly, the number of customers who churned (Churn = 1) is also very close, with 939 female customers and 930 male customers. This indicates that gender is not a significant factor in determining service churn behavior, as the churn rate between the two genders is almost the same.

Although gender does not show a significant difference in churn rates, understanding other factors (such as cost, services used, or contract type) may be more important in building an effective customer retention strategy. This chart highlights that customer behavior is not influenced by gender but may depend on factors related to the service or the value provided.

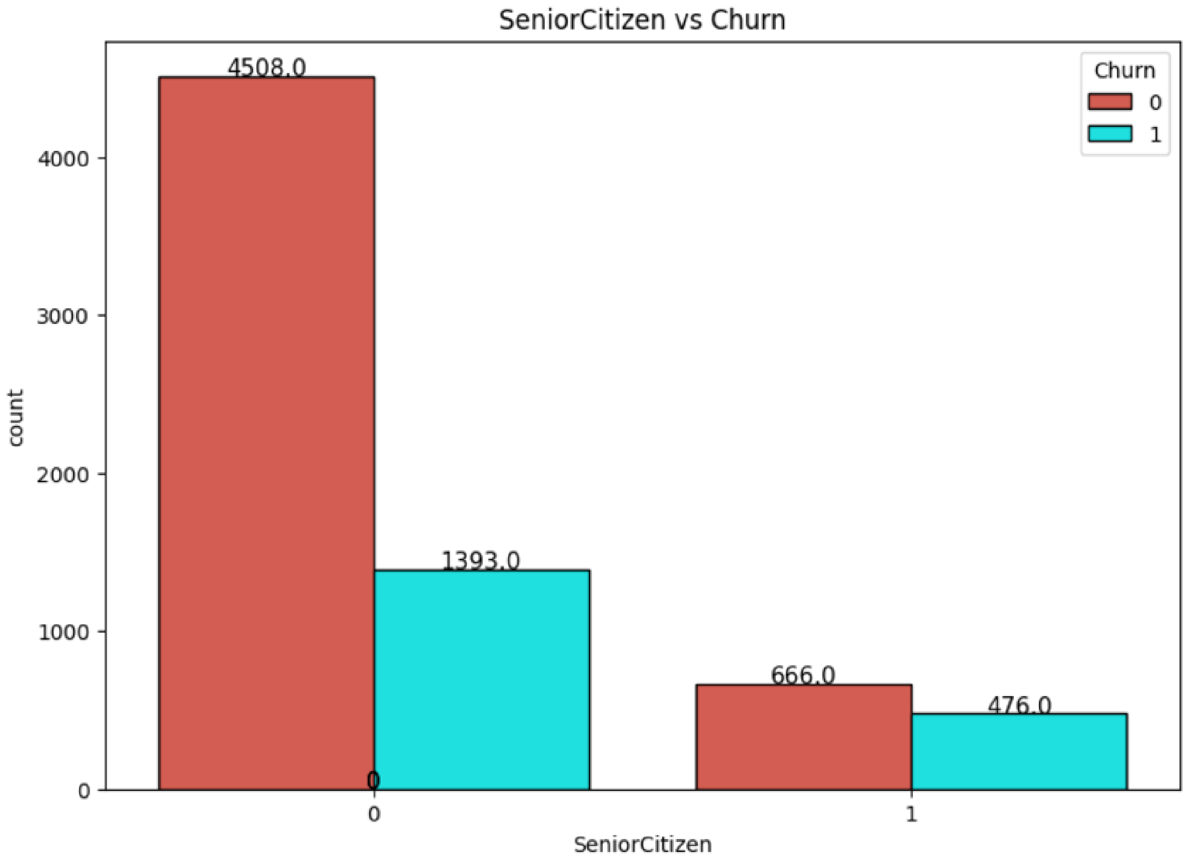

Figure 3.10: The Comparison of Churn and Not Churn Customers by SeniorCitizen

For the non-senior group (SeniorCitizen = 0), the majority of customers did not churn (Churn = 0), with 4508 customers, while the number of customers who churned (Churn = 1) is 1393. This shows that this group has a significantly lower churn rate compared to the retention rate.

For the senior citizen group (SeniorCitizen = 1), the number of customers who did not churn is 666, and the number who churned is 476. The churn rate in this group is much higher compared to the non-senior group. The chart indicates that senior citizens tend to churn at a higher rate than other customer groups. This could be related to factors such as cost, service usage levels, or how well the service meets their needs. To reduce churn, businesses should develop special offers, tailored service packages, or provide better support for the senior customer group, thereby increasing the likelihood of retaining this segment.

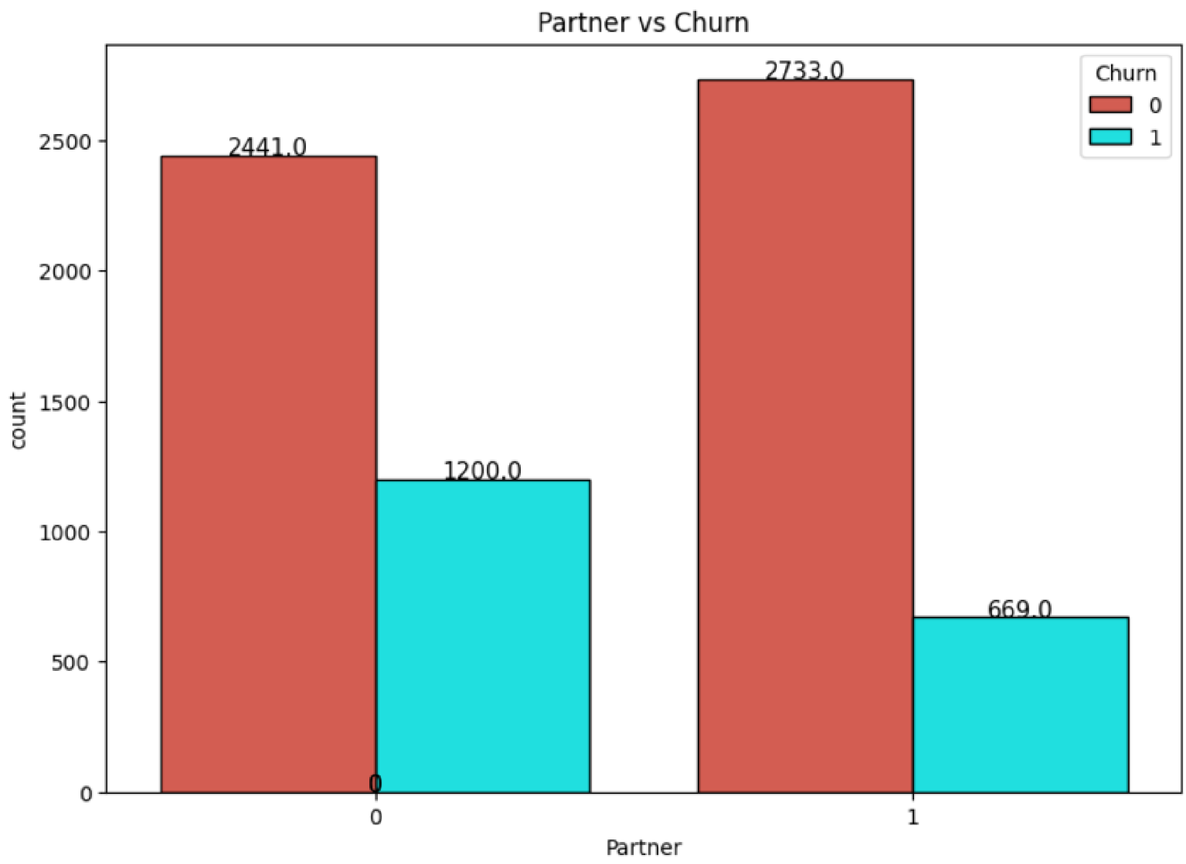

Figure 3.11: The Comparison of Churn and Not Churn Customers by Partner

The group of customers without a partner (Partner = 0) includes 2441 customers who did not churn (Churn = 0) but has as many as 1200 customers who churned (Churn = 1). The churn rate in this group is relatively high.

The group of customers with a partner (Partner = 1) includes 2733 customers who did not churn (Churn = 0) and only 669 customers who churned (Churn = 1). The churn rate in this group is significantly lower compared to the group without a partner.

The chart shows that customers with a partner are less likely to churn compared to those without a partner. This could be related to the stability and long-term service needs of customers with families or partners. To reduce the churn rate, businesses might consider designing promotional programs or tailored service packages for single customers to increase the likelihood of retaining this segment.

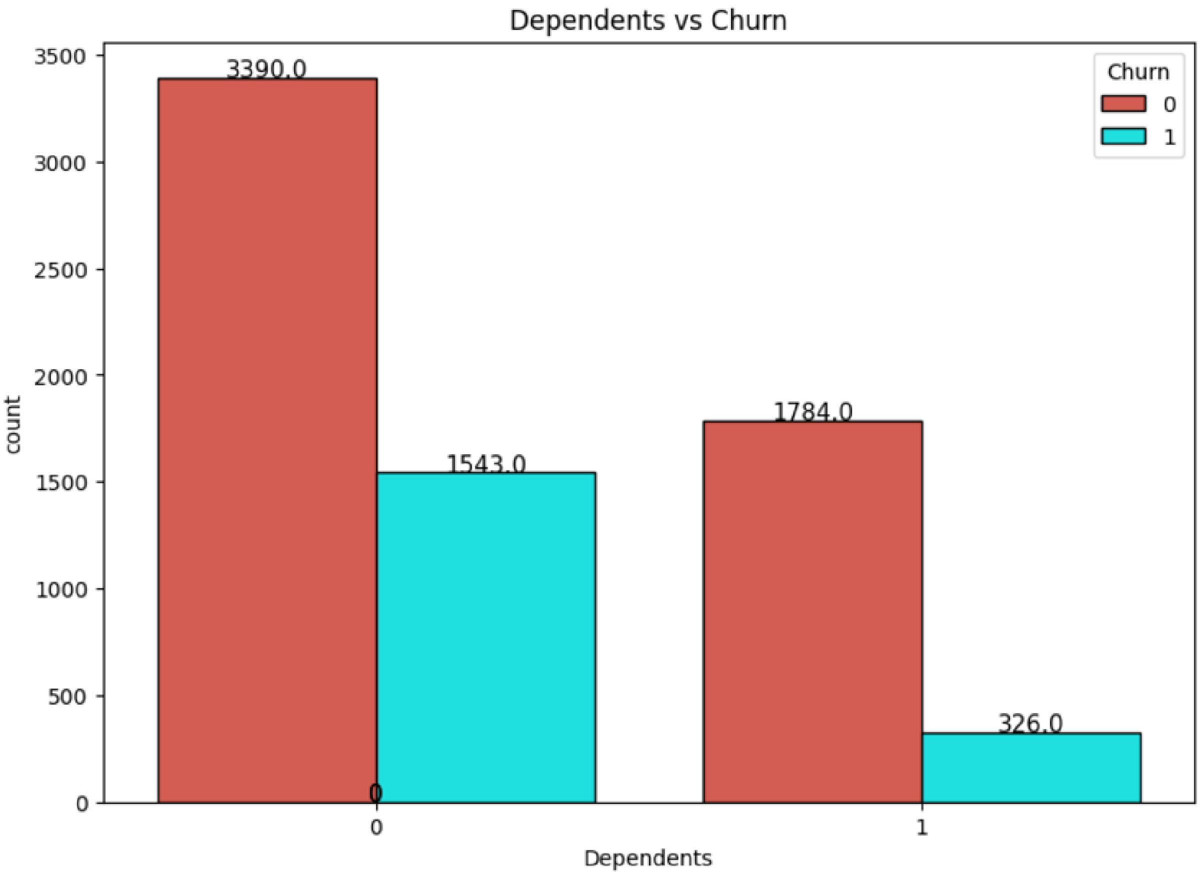

Figure 3.12: The Comparison of Churn and Not Churn Customers by Dependents

For customers without dependents (Dependents = 0), there are 3390 customers who did not churn (Churn = 0), but as many as 1543 customers churned (Churn = 1). The churn rate in this group is quite high

For customers with dependents (Dependents = 1), there are 1784 customers who did not churn (Churn = 0) and only 326 customers who churned (Churn = 1). The churn rate in this group is significantly lower.

The chart shows that customers with dependents are less likely to churn compared to those without dependents. This could reflect the fact that customers with dependents typically have more stable and long-term service needs. To reduce churn, businesses should consider policies or service packages that are more suitable for independent customers (those without dependents), such as offering additional value-added services or incentives to increase the likelihood of retaining this segment.

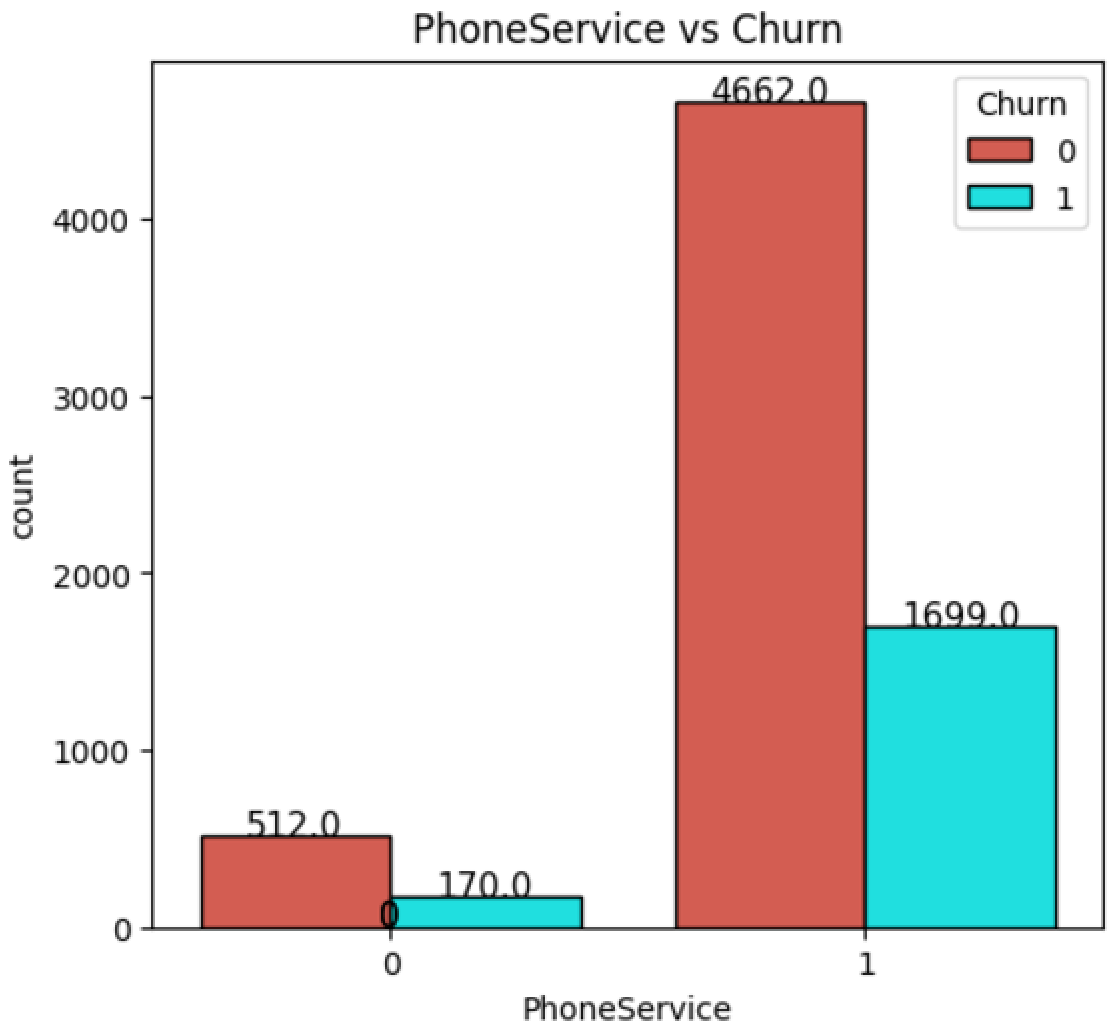

Figure 3.13: The Comparison of Churn and Not Churn Customers by PhoneService

For customers without phone service (PhoneService = 0), there are 512 customers who did not churn (Churn = 0) and 170 customers who churned (Churn = 1). The churn rate in this group is relatively low.

For customers with phone service (PhoneService = 1), there are 4662 customers who did not churn (Churn = 0), but there are also 1699 customers who churned (Churn = 1). The number of churned customers is significantly higher in this group, although the total number of customers with phone service is also higher.

The chart shows that the majority of customers use phone service, and the churn rate in this group is much higher compared to the group without phone service. This could indicate that phone service may be a factor influencing the decision to churn, potentially related to service quality, cost, or how well the service meets their needs. To reduce churn, the business might consider improving phone service packages or offering incentives to enhance the customer experience in this segment.

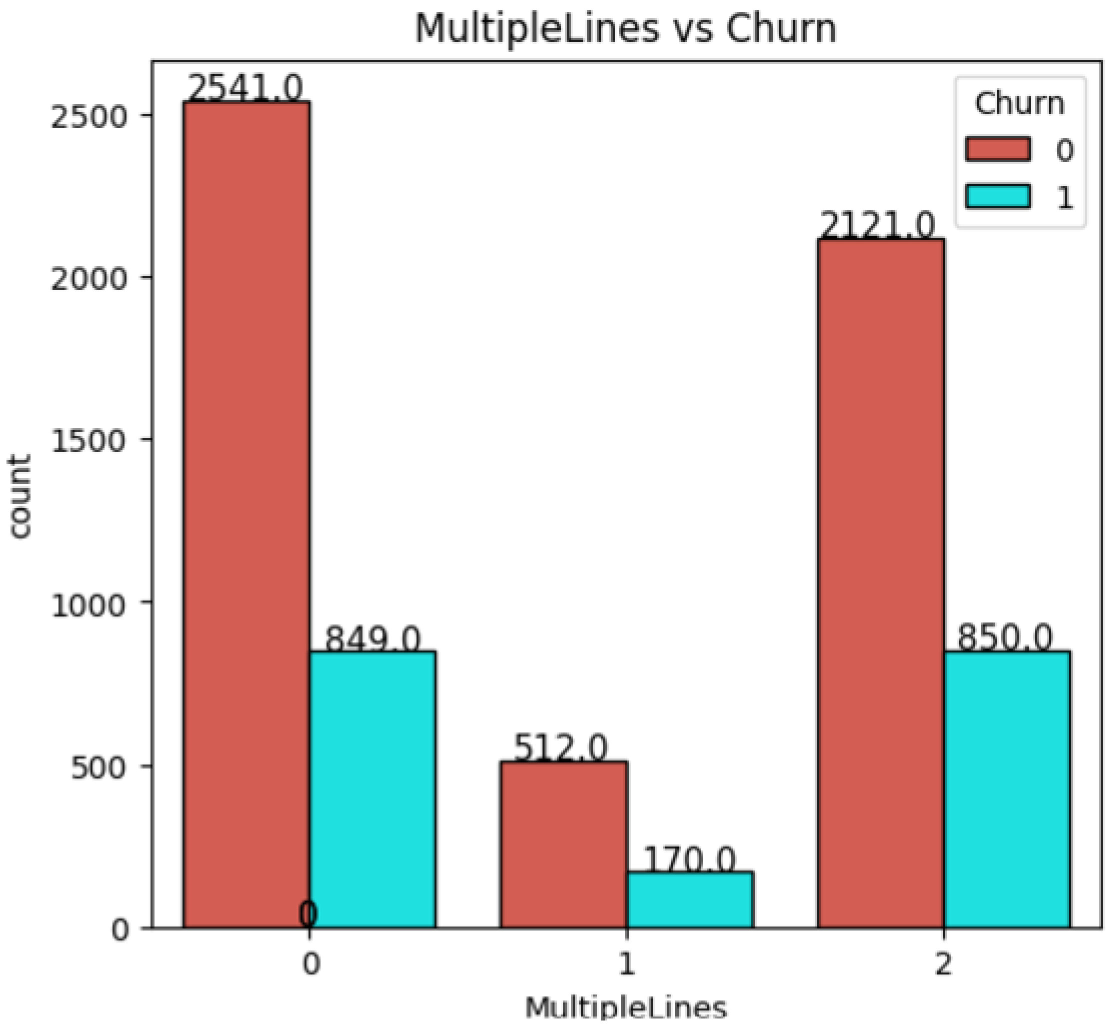

Figure 3.14: The Comparison of Churn and Not Churn Customers by MultipleLines

No phone line usage (MultipleLines = 0):

- 2541 customers did not churn (Churn = 0), and 849 customers churned (Churn = 1).

Single phone line usage (MultipleLines = 1):

- 512 customers did not churn (Churn = 0), and 170 customers churned (Churn = 1).

Multiple phone line usage (MultipleLines = 2):

- 2121 customers did not churn (Churn = 0), and 850 customers churned (Churn = 1).

From the chart, it is evident that the group of customers using multiple phone lines (MultipleLines = 2) has the highest churn rate, with the number of churned customers almost equal to the number of retained customers. Meanwhile, the group that does not use phone lines (MultipleLines = 0) has a higher retention rate, indicating that they are less likely to churn. Using multiple phone lines may be associated with higher needs and costs, leading to a greater likelihood of churn if the service experience does not meet expectations. To reduce the churn rate, businesses can focus on improving service quality and offering incentives for this high-demand customer segment.

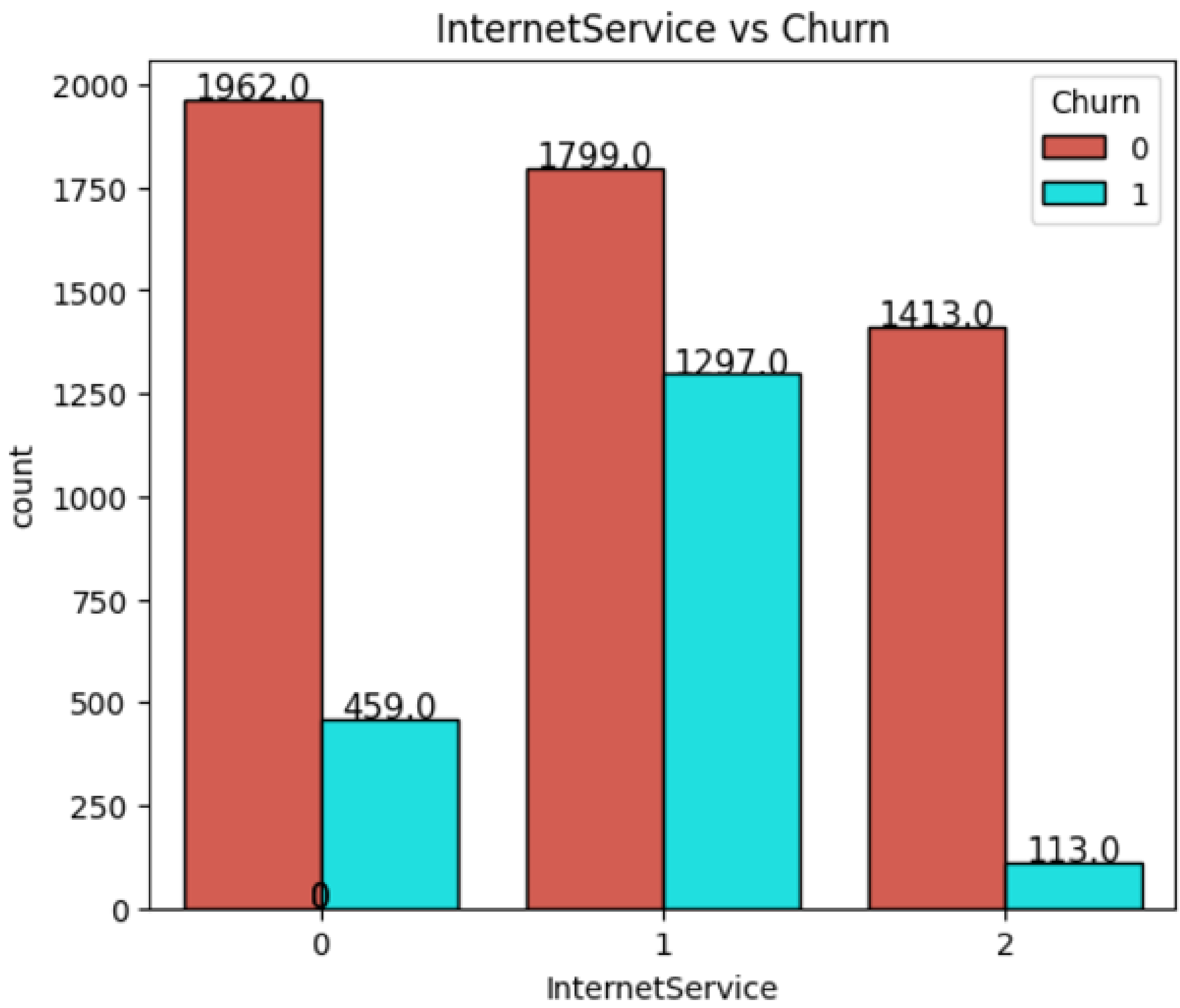

Figure 3.15: The Comparison of Churn and Not Churn Customers by InternetService

No Internet usage (InternetService = 0):

- 1962 customers did not churn (Churn = 0), and 459 customers churned (Churn = 1).
- The churn rate in this group is low, possibly because they are only using other services besides the Internet.

DSL Internet service (InternetService = 1):

- 1799 customers did not churn (Churn = 0), and 1297 customers churned (Churn = 1).
- The churn rate in this group is relatively high, accounting for most of the churned customers.

Fiber optic Internet service (InternetService = 2):

- 1413 customers did not churn (Churn = 0), and 113 customers churned (Churn = 1).
- The fiber optic group has a significantly lower churn rate compared to the DSL group.

The chart shows that the group of customers using DSL Internet service has the highest churn rate, indicating that the quality or suitability of the DSL service may not meet customer needs well. Meanwhile, the group without Internet service and the fiber optic group have lower churn rates. To reduce churn, businesses can focus on improving the quality of DSL service, increasing connection speeds, or offering incentives to this group. At the same time, they should continue to develop and promote fiber optic services to attract and retain customers.

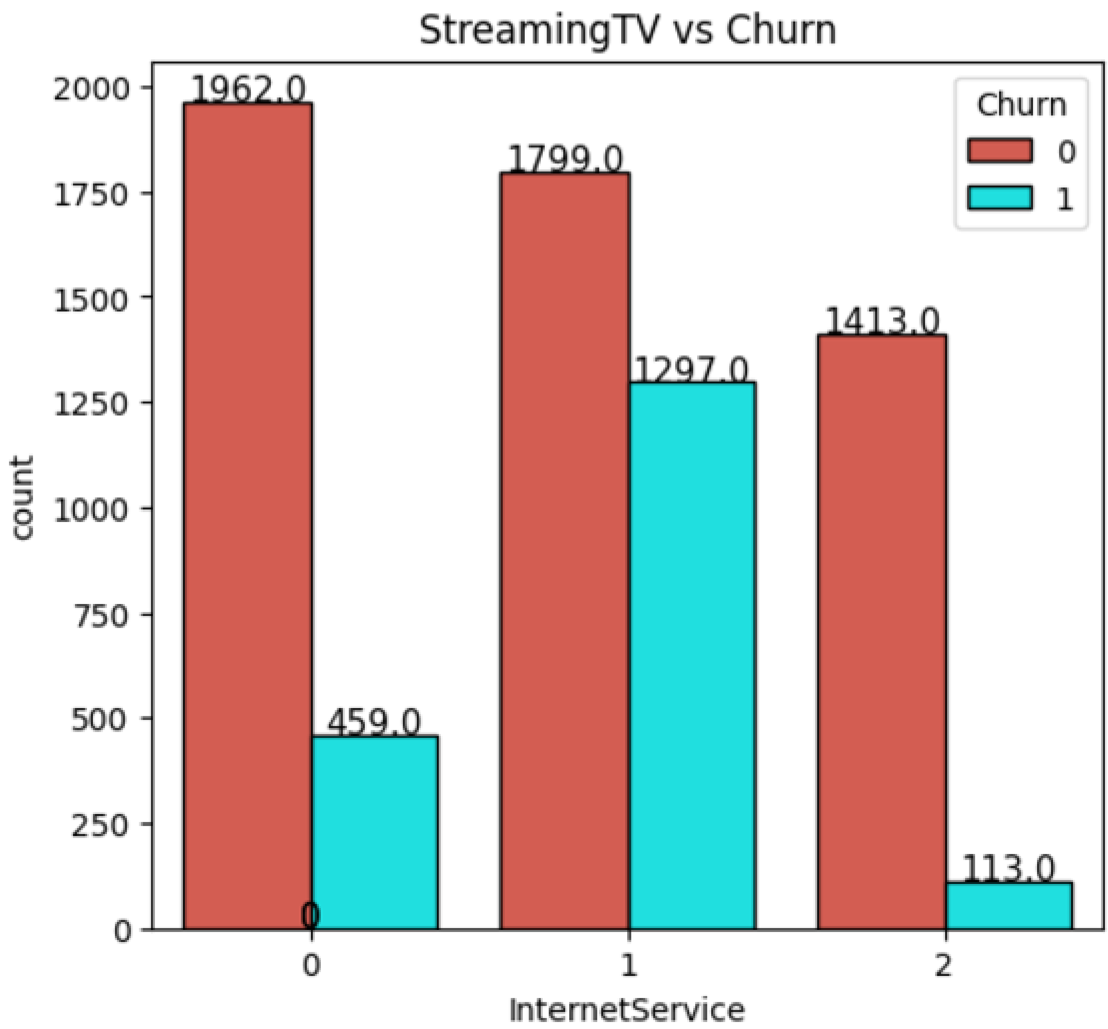

Figure 3.16: The Comparison of Churn and Not Churn Customers by StreamingTV

No StreamingTV usage (StreamingTV = 0):

- 1962 customers did not churn (Churn = 0), and 459 customers churned (Churn = 1).
- The churn rate in this group is low, possibly because they are not dependent on the TV streaming service.

StreamingTV via DSL (StreamingTV = 1):

- 1799 customers did not churn (Churn = 0), and 1297 customers churned (Churn = 1).
- The churn rate is significantly higher compared to the non-streaming group.

StreamingTV via Fiber Optic (StreamingTV = 2):

- 1413 customers did not churn (Churn = 0), and 113 customers churned (Churn = 1).

- The group using Fiber Optic StreamingTV has a significantly lower churn rate compared to the DSL group.

The chart shows that the group of customers using StreamingTV via DSL has the highest churn rate, indicating that the quality or experience of DSL service may not be meeting customer expectations. Meanwhile, the group using StreamingTV via Fiber Optic has a lower churn rate, which may be related to the higher speeds and better experience of this service.

The group not using StreamingTV has the lowest churn rate, possibly because they have lower or no dependency on the service. To reduce churn, the company could focus on improving the DSL service, optimizing streaming quality, and continuing to invest in fiber optic services to enhance the customer experience.

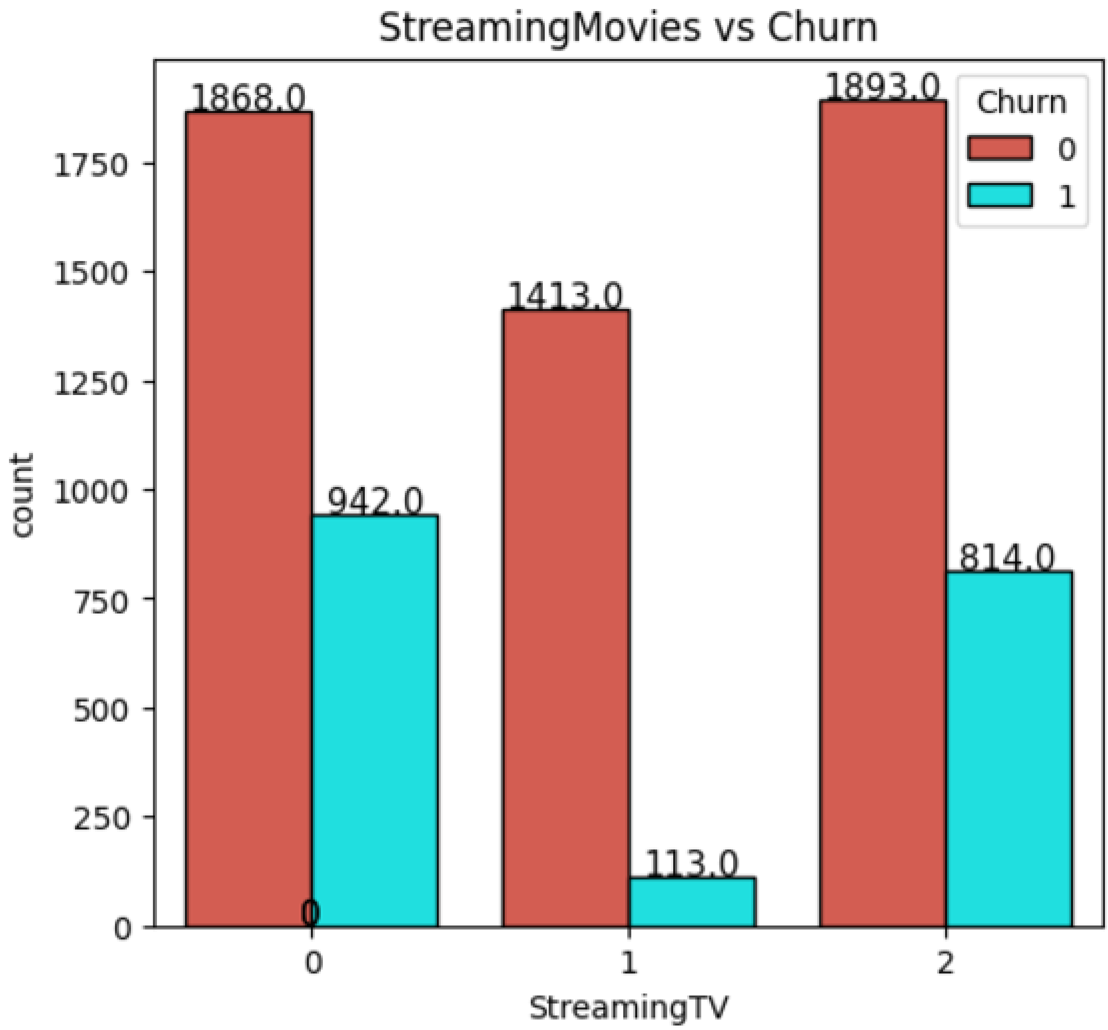

Figure 3.17: The Comparison of Churn and Not Churn Customers by StreamingMovies

No StreamingMovies usage (StreamingMovies = 0):

- 1868 customers did not churn (Churn = 0), and 942 customers churned (Churn = 1).
- The churn rate in this group is relatively high, possibly because these customers are not receiving enough value from other services to maintain their subscription.

StreamingMovies via DSL (StreamingMovies = 1):

- 1413 customers did not churn (Churn = 0), and only 113 customers churned (Churn = 1).
- This group has the lowest churn rate among all three groups.

StreamingMovies via Fiber Optic (StreamingMovies = 2):

- 1893 customers did not churn (Churn = 0), and 814 customers churned (Churn = 1).

- The churn rate in this group is relatively high, despite the total number of customers being fairly large.

The group using StreamingMovies via DSL has the lowest churn rate, indicating that DSL service is well-suited for customers with basic and stable needs. However, both the group using Fiber Optic and the group not using the service have higher churn rates, particularly the group that doesn't use StreamingMovies.

This may suggest that offering additional valuable content or enhancing the streaming experience (especially with fiber optic) could help reduce churn. Additionally, it’s important to develop strategies to improve the experience for the group not using this service, such as introducing promotional packages to encourage them to try out the service and increase its perceived value.

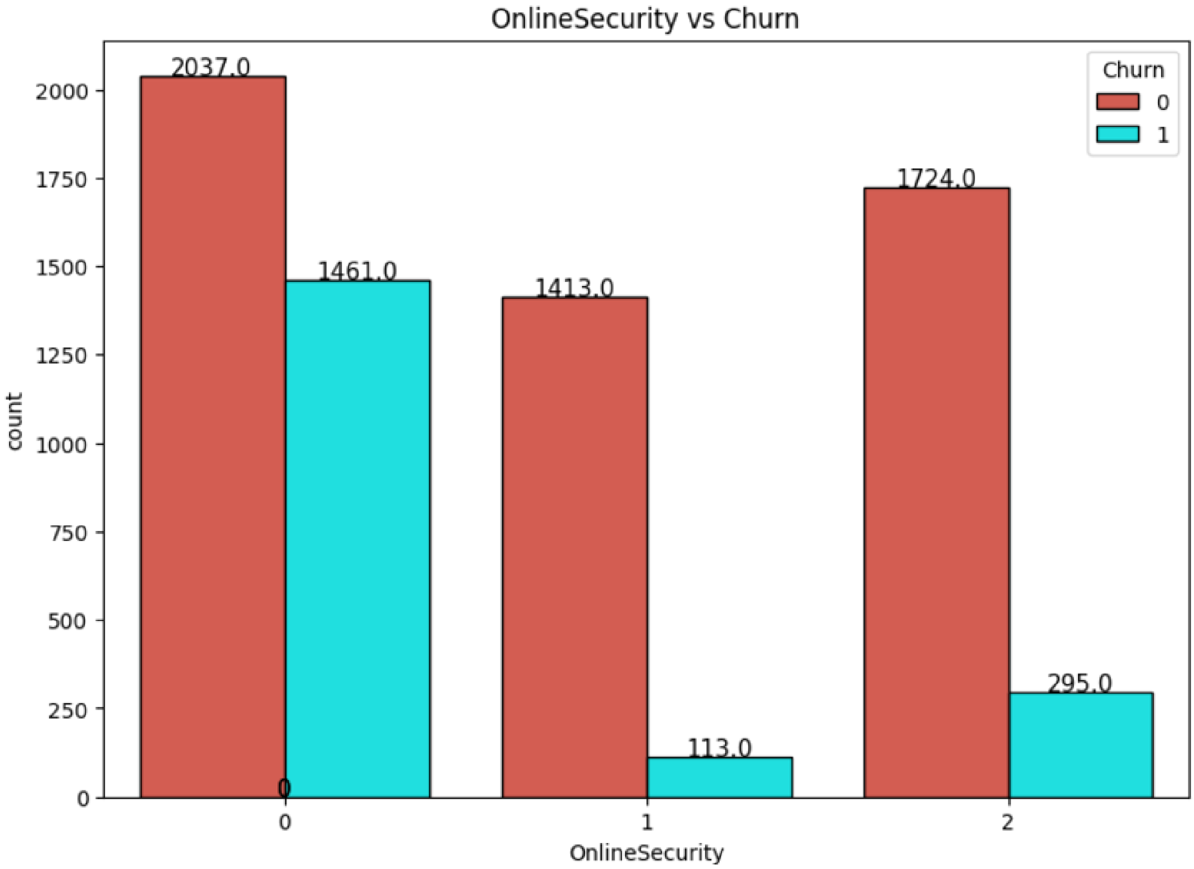

Figure 3.18: The Comparison of Churn and Not Churn Customers by OnlineSecurity

No OnlineSecurity usage (OnlineSecurity = 0):

- 2037 customers did not churn (Churn = 0), and 1461 customers churned (Churn = 1).

- The churn rate in this group is the highest among all three groups, accounting for nearly half of the customers.

OnlineSecurity via DSL (OnlineSecurity = 1):

- 1413 customers did not churn (Churn = 0), and only 113 customers churned (Churn = 1).

- This group has the lowest churn rate, indicating that customers using security via DSL are more stable.

OnlineSecurity via Fiber Optic (OnlineSecurity = 2):

- 1724 customers did not churn (Churn = 0), and 295 customers churned (Churn = 1).

- The churn rate in this group is lower compared to the group that does not use security services.

The group that does not use online security services has the highest churn rate, suggesting that the lack of security might diminish customer experience or satisfaction. Meanwhile, groups using security services, whether via DSL or fiber optic, have lower churn rates, especially those using DSL.

This highlights that online security services play a crucial role in customer retention. Businesses should focus on introducing and encouraging customers to use these services, while also improving the quality and value of online security packages to enhance customer loyalty.

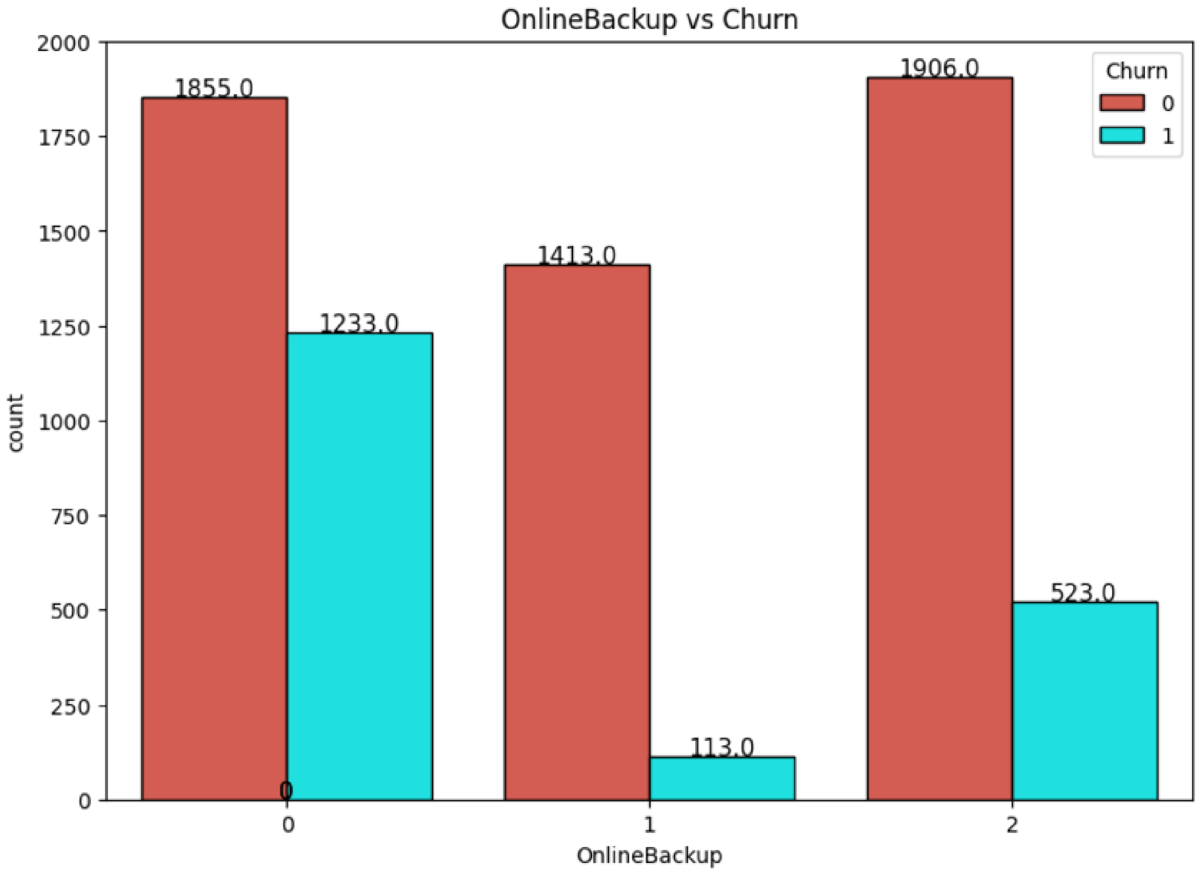

Figure 3.19: The Comparison of Churn and Not Churn Customers by OnlineBackup

No OnlineBackup usage (OnlineBackup = 0):

- 1855 customers did not churn (Churn = 0), and 1233 customers churned (Churn = 1).
- This group has a relatively high churn rate, accounting for a large portion of customers not using the service.

OnlineBackup via DSL (OnlineBackup = 1):

- 1413 customers did not churn (Churn = 0), and only 113 customers churned (Churn = 1).
- This group has the lowest churn rate, suggesting that using online backup services via DSL helps increase customer retention.

OnlineBackup via Fiber Optic (OnlineBackup = 2):

- 1906 customers did not churn (Churn = 0), and 523 customers churned (Churn = 1).
- The churn rate in this group is lower than the group not using the service, but still higher than the group using DSL.

The group not using online backup services has the highest churn rate, suggesting that backup services may play a critical role in improving customer satisfaction. The group using the service via DSL has the lowest churn rate, showing that this is an effective channel for retaining customers.

The group using the service via Fiber Optic also has a significantly lower churn rate compared to the non-usage group, highlighting the positive impact of the service. To reduce churn, businesses could focus on encouraging the use of online backup services while improving the customer experience for both channels (DSL and Fiber Optic).

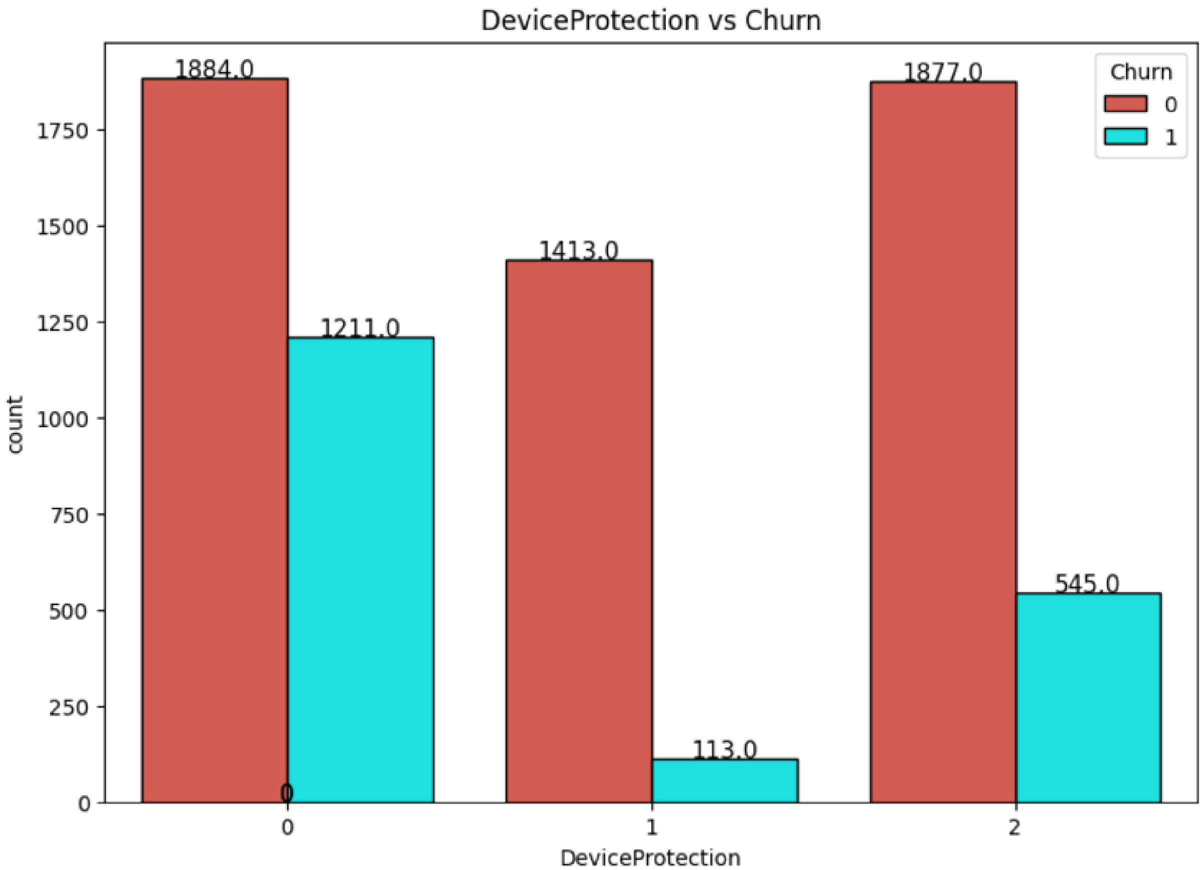

Figure 3.20: The Comparison of Churn and Not Churn Customers by DeviceProtection

No Device Protection (DeviceProtection = 0):

- 1884 customers did not churn (Churn = 0), and 1211 customers churned (Churn = 1).
- The churn rate in this group is relatively high.

Device Protection via DSL (DeviceProtection = 1):

• 1413 customers did not churn (Churn = 0), and only 113 customers churned (Churn = 1).

• This group has the lowest churn rate.

Device Protection via Fiber Optic (DeviceProtection = 2):

• 1877 customers did not churn (Churn = 0), and 545 customers churned (Churn = 1).

• The churn rate in this group is lower than the non-usage group but still higher than the DSL group.

The group not using the service has the highest churn rate, while the group using it via DSL shows a clear advantage in customer retention. This highlights the importance of offering and encouraging the use of device protection services.

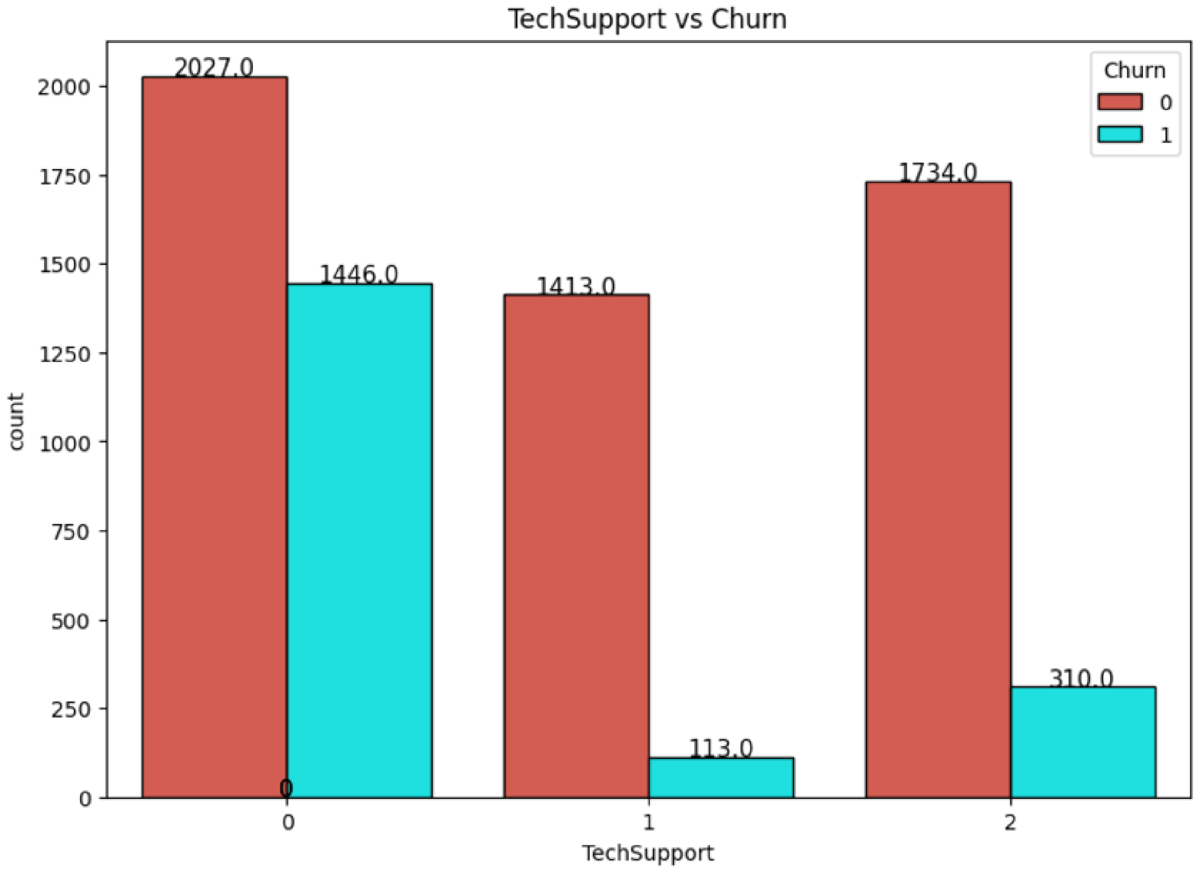

Figure 3.21: The Comparison of Churn and Not Churn Customers by TechSupport

No Tech Support (TechSupport = 0):

• 2027 customers did not churn (Churn = 0), and 1446 customers churned (Churn = 1).

- The churn rate in this group is very high, nearly the same as the group not using Device Protection.

Tech Support via DSL (TechSupport = 1):

- 1413 customers did not churn (Churn = 0), and only 113 customers churned (Churn = 1).
- This group continues to show the lowest churn rate.

Tech Support via Fiber Optic (TechSupport = 2):

- 1734 customers did not churn (Churn = 0), and 310 customers churned (Churn = 1).
- The churn rate in this group is lower than the non-usage group but higher than the DSL group.

Not using tech support services leads to the highest churn rate, suggesting that this service plays a crucial role in customer retention. The group using DSL service shows higher engagement, indicating that providing effective tech support could significantly reduce churn.

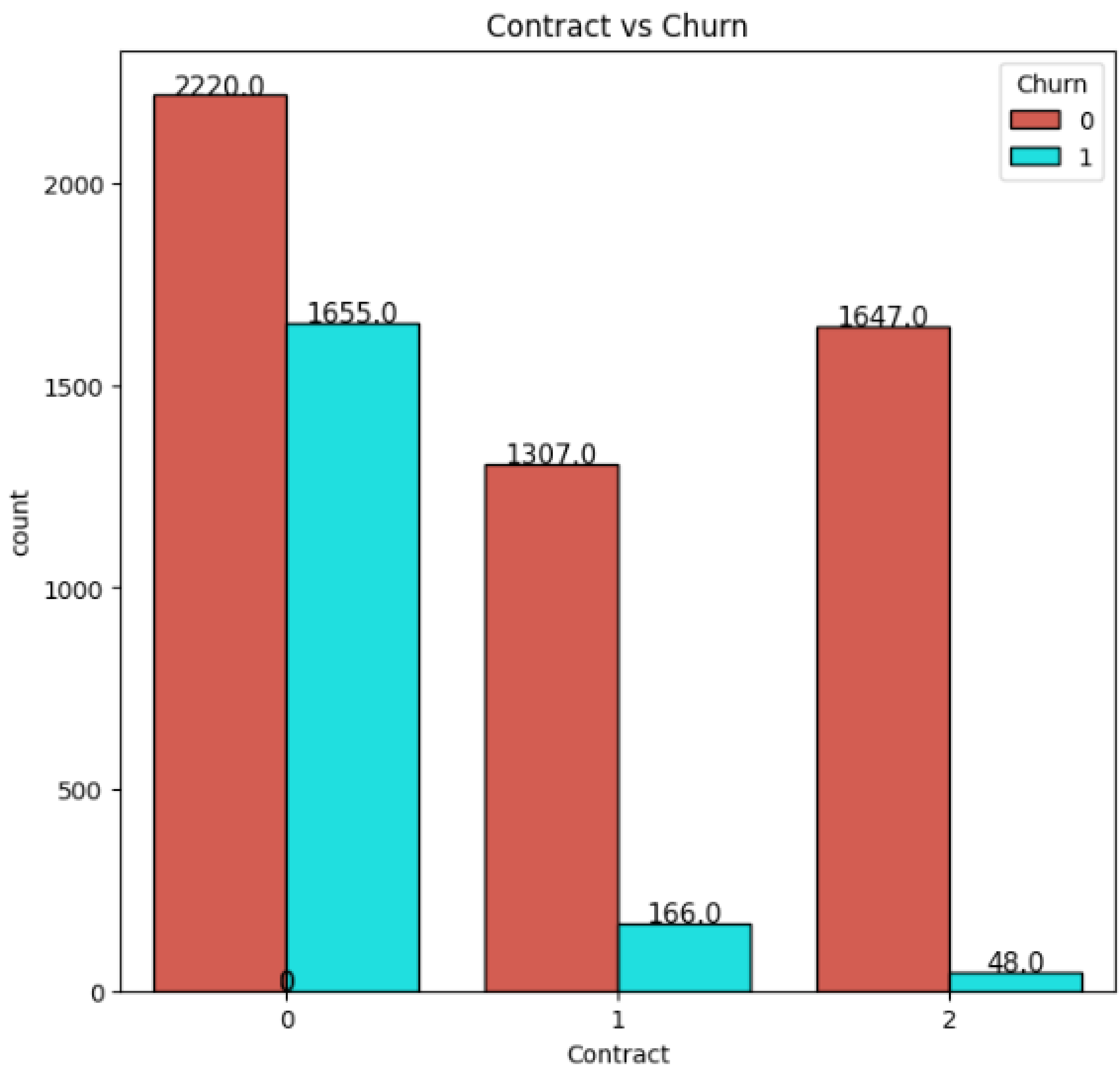

Figure 3.22: The Comparison of Churn and Not Churn Customers by Contract

Monthly Contract (Contract = 0):

- 2220 customers did not churn (Churn = 0), and 1655 customers churned (Churn = 1).
- The churn rate is the highest among the three contract types.

1-Year Contract (Contract = 1):

- 1307 customers did not churn (Churn = 0), and only 166 customers churned (Churn = 1).
- The churn rate is significantly lower compared to the monthly contract.

2-Year Contract (Contract = 2):

- 1647 customers did not churn (Churn = 0), and only 48 customers churned (Churn = 1).
- This group has the lowest churn rate.

Customers with a monthly contract have the highest churn rate, indicating that this type of contract doesn't foster long-term commitment. In contrast, longer-term contracts (1-year and 2-year) significantly reduce the churn rate.

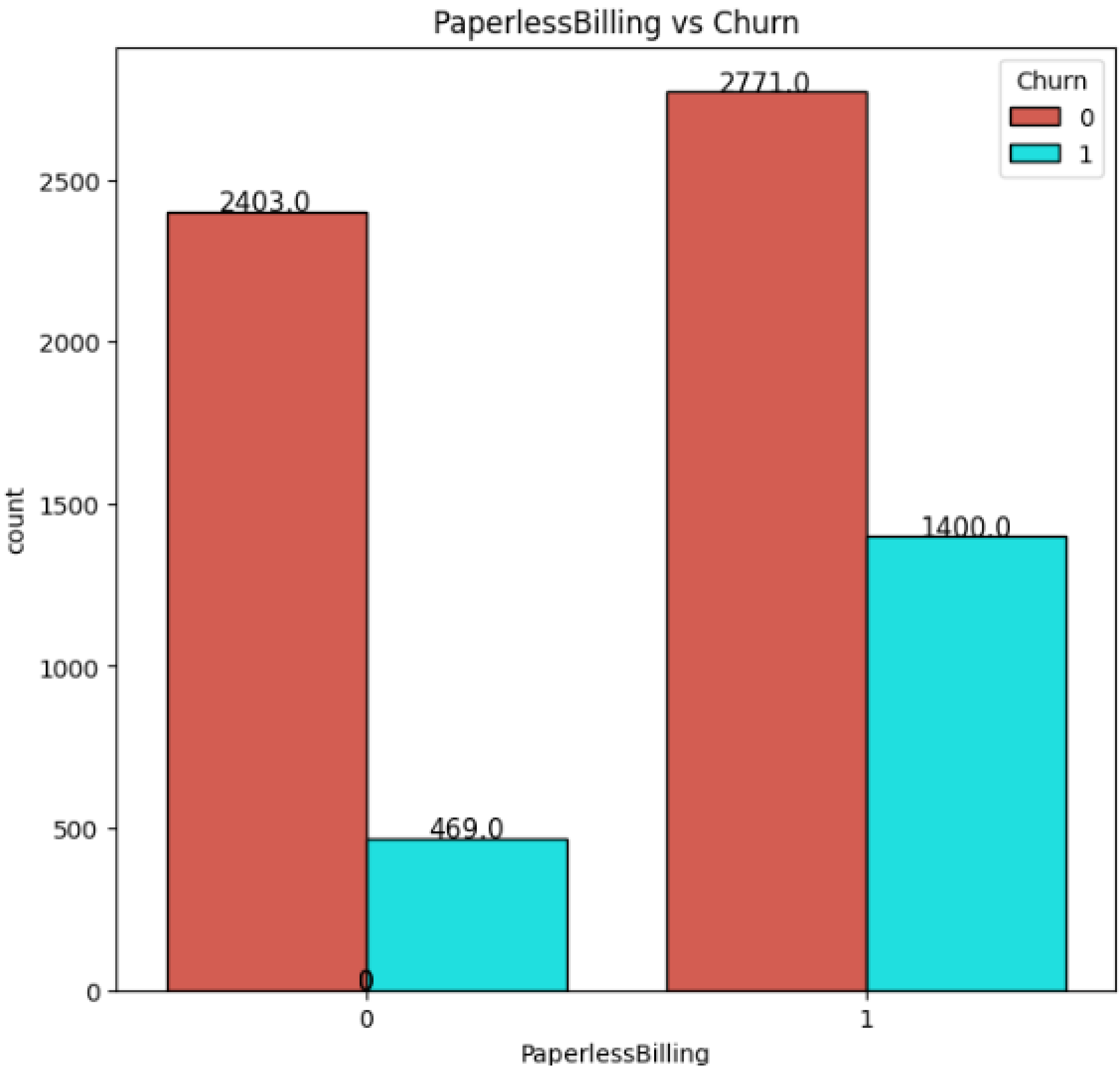

Figure 3.23: The Comparison of Churn and Not Churn Customers by PaperlessBilling

Not Using Paperless Billing (PaperlessBilling = 0):

- 2403 customers did not churn (Churn = 0), and only 469 customers churned (Churn = 1).
- The churn rate in this group is very low.

Using Paperless Billing (PaperlessBilling = 1):

- 2771 customers did not churn (Churn = 0), but 1400 customers churned (Churn = 1).
- The churn rate is significantly higher compared to the group not using paperless billing.

Customers who use paperless billing have a higher churn rate, which may be attributed to a perceived lack of connection or potential issues with the service experience.

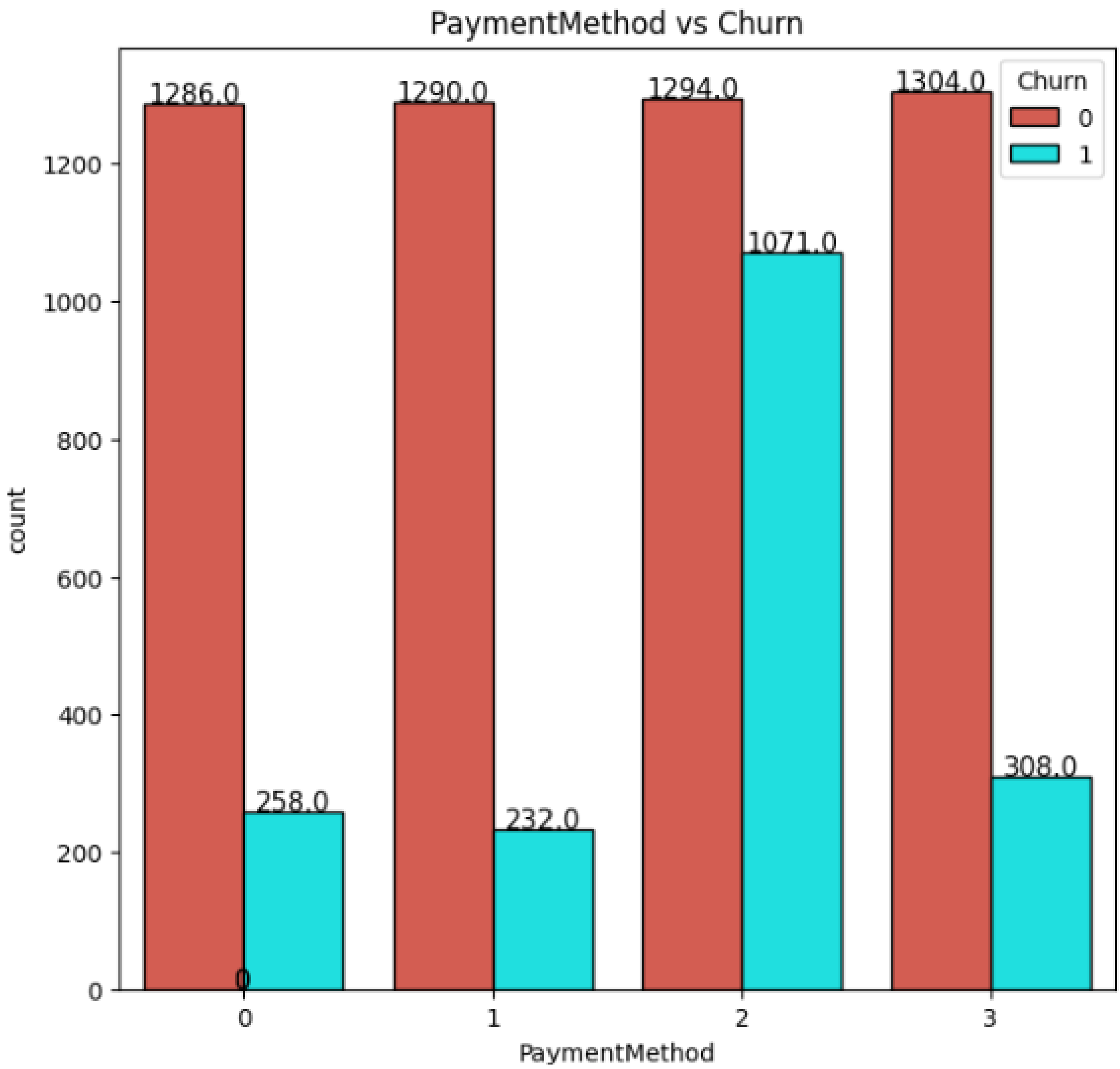

Figure 3.24: The Comparison of Churn and Not Churn Customers by PaymentMethod

Electronic check (PaymentMethod = 0):

- 1286 customers did not churn (Churn = 0), and 258 customers churned (Churn = 1).
- The churn rate is relatively high.

Mailed check (PaymentMethod = 1):

- 1290 customers did not churn (Churn = 0), and 232 customers churned (Churn = 1).
- The churn rate is lower.

Bank transfer (PaymentMethod = 2):

- 1294 customers did not churn (Churn = 0), and 1071 customers churned (Churn = 1).
- This group has the highest churn rate.

Credit card (PaymentMethod = 3):

- 1304 customers did not churn (Churn = 0), and 308 customers churned (Churn = 1).
- The churn rate is relatively low.

The payment method of bank transfer has the highest churn rate, while payment via credit card and checks has a lower churn rate. This suggests that the payment method also plays an important role in customer retention.

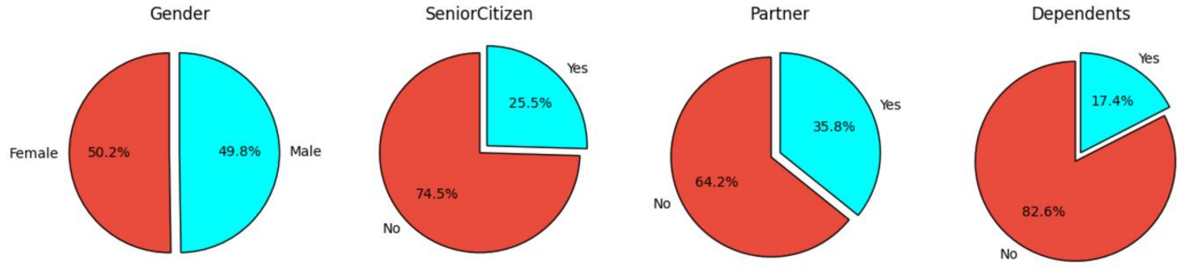

Figure 3.25: The Distribution of Gender, SeniorCitizen, Partner, and Dependents

In terms of gender, male and female customers are almost evenly distributed, with females making up 50.2% and males making up 49.8%. The majority of customers are not senior citizens (74.5%), while seniors account for only 25.5%, indicating that younger customers dominate. Regarding marital status, 64.2% of customers are without a partner, and 82.6% do not have dependents, suggesting that most customers belong to the independent group, not bound by family responsibilities. These characteristics suggest that the business should focus on developing marketing strategies targeting young, independent customers, while also designing service packages or promotional programs specifically for senior customers, those with partners, or those with dependents to attract and retain them.

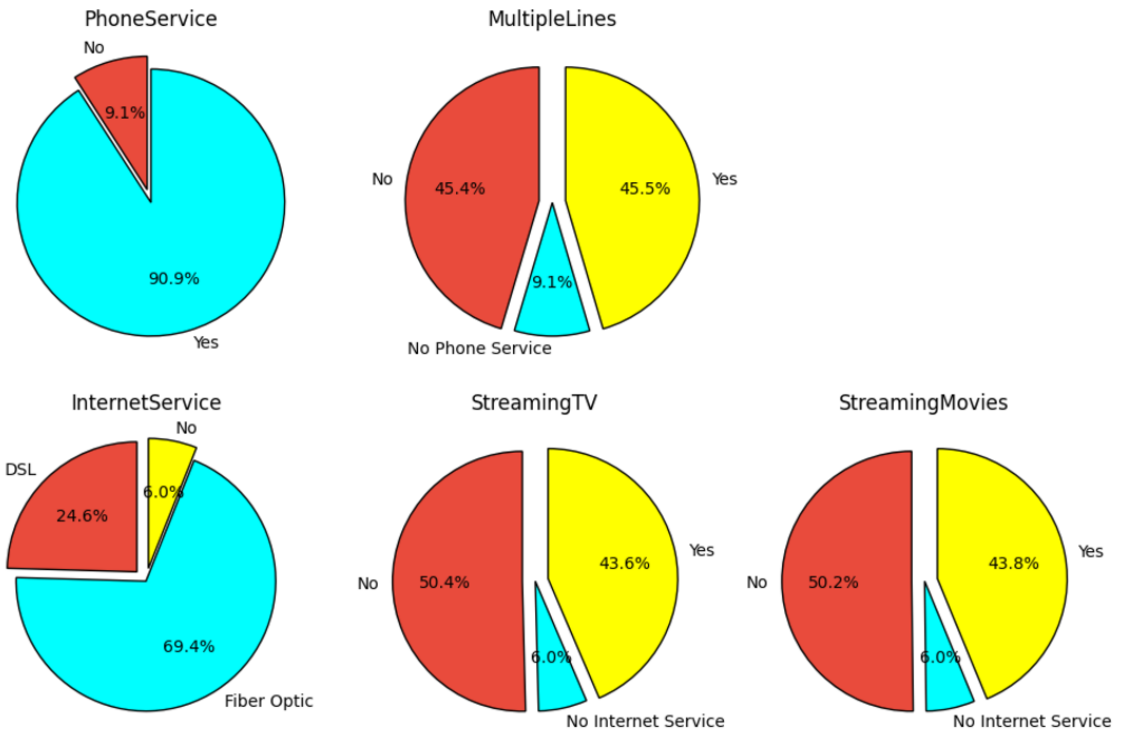

Figure 3.26: The Distribution of PhoneService, MultipleLines, InternetService, StreamingTV, and StreamingMovies

The majority of customers (90.9%) use phone services, with an almost even distribution between customers using a single line (45.4%) and multiple lines (45.5%). Among the Internet services, fiber optic is the most popular choice with 69.4% of customers, compared to 24.6% using DSL and only 6.0% not using the Internet. Value-added services like StreamingTV and StreamingMovies have a similar distribution, with more than 50% of customers not using these services and about 43% of customers using them. This suggests a significant potential in encouraging customers who have not yet used streaming services through targeted marketing strategies or suitable promotional packages, especially for those who already use Internet services but have not fully explored the additional value from supplementary services.

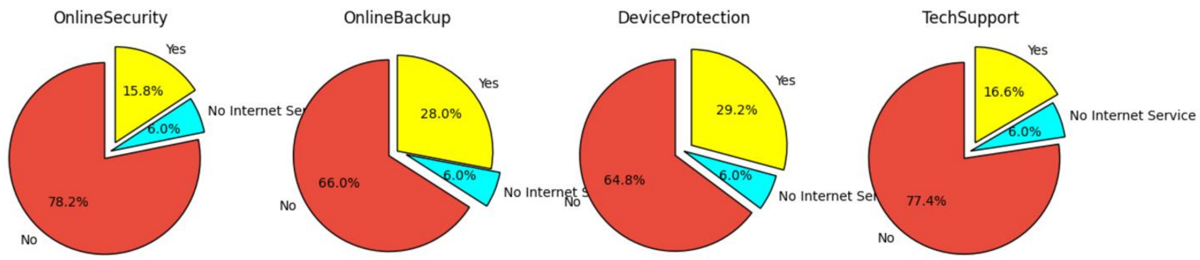

Figure 3.27: The Distribution of OnlineSecurity, OnlineBackup, DeviceProtection, and TechSupport

The non-usage rate ranges from 64.8% (DeviceProtection) to 78.2% (OnlineSecurity). Only a small portion of customers use these online services, with OnlineBackup and DeviceProtection having the highest usage rates at 28.0% and 29.2%, respectively, while TechSupport and OnlineSecurity have lower usage rates at 16.6% and 15.8%, respectively. About 6.0% of customers do not use the Internet, which limits their access to online services. The low usage rates of these services indicate that the business needs to intensify customer education on the benefits of value-added services, while also implementing promotional strategies or bundled service packages to increase usage and improve customer satisfaction.

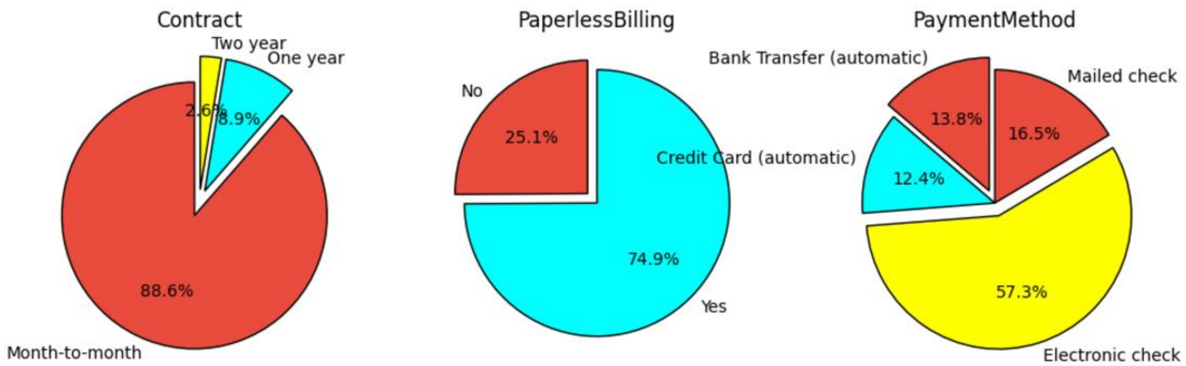

Figure 3.28: The Distribution of Contract, PaperlessBilling, and PaymentMethod

The majority of customers (88.6%) prefer monthly contracts for their flexibility, while only 9.4% choose a 1-year contract and 2.0% opt for a 2-year contract, reflecting a preference for non-long-term commitments. Regarding billing, 74.9% of customers use paperless billing, reflecting the trend toward digitization, but 25.1% still prefer paper bills. As for payment methods, 57.3% of customers use electronic checks, while other methods like credit cards (16.5%), mailed checks (13.8%), and bank transfers

(12.4%) account for smaller shares. This choice shows the demand for convenience, and traditional payment habits are still prevalent among certain customer groups. The business should focus on strategies to encourage customers to switch to long-term contracts and modern payment methods while enhancing the experience to better meet the diverse needs of customers.

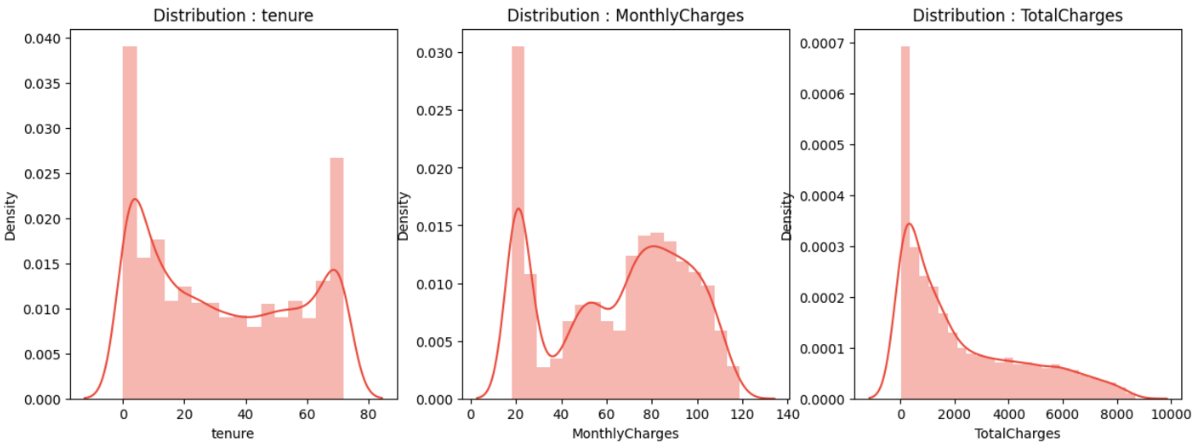

Figure 3.29: The Distribution of tenure, MonthlyCharges, and TotalCharges

The chart above shows the distribution of three variables: tenure, MonthlyCharges, and TotalCharges in the dataset. For tenure (the duration customers have used the service), the distribution shows a clear left skew, with a high density concentrated around shorter durations (close to 0 months). This suggests that most customers only use the service for a short time, possibly due to early churn or because they are new customers. Next, the distribution of MonthlyCharges (monthly charges) is multimodal, with values concentrated around 20-30, and other peaks at higher levels, ranging from 50-80. This reflects a price tiering, corresponding to different service packages. Finally, TotalCharges (total charges) has a strong right skew, with the majority of data concentrated at lower values. This distribution makes sense because total charges often depend on the tenure, meaning customers with shorter tenures will have lower TotalCharges.

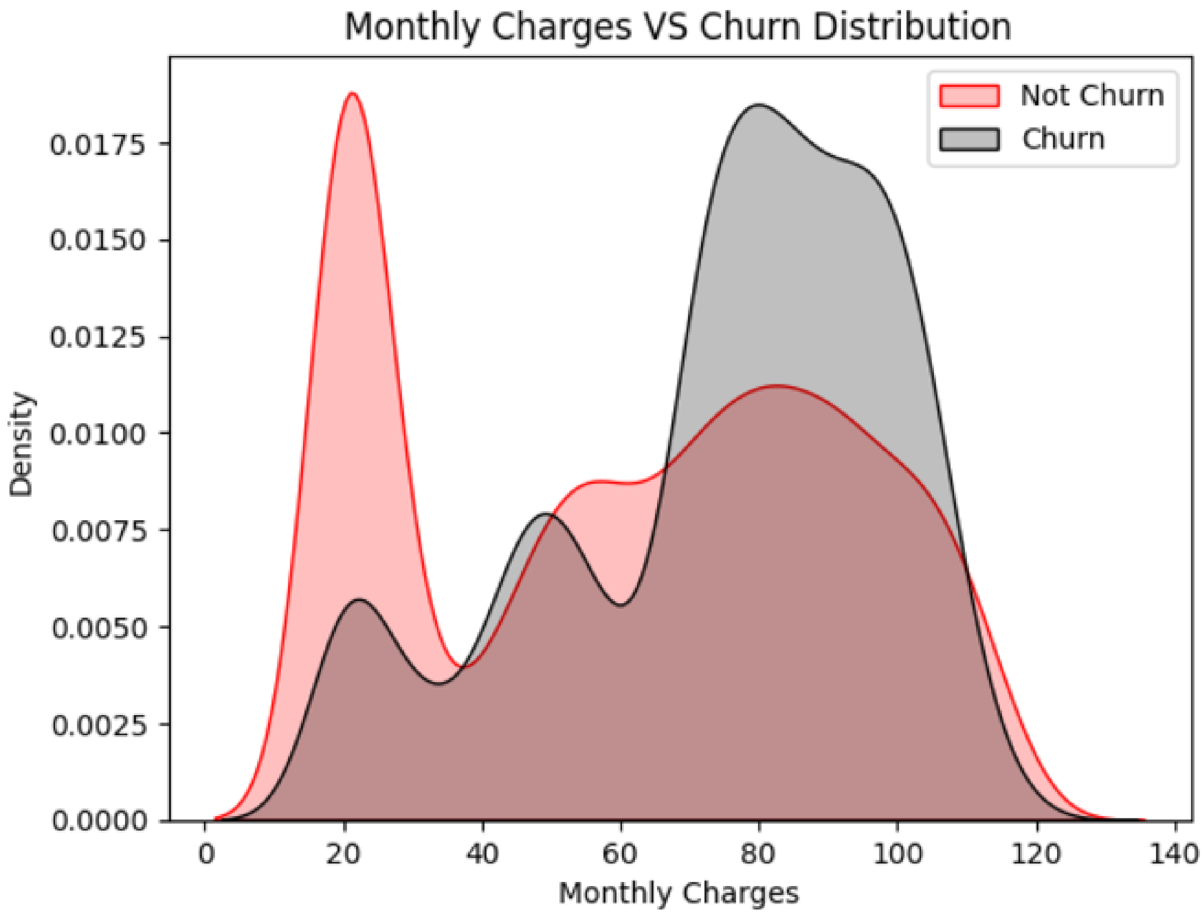

Figure 3.30: The Comparison of Churn and Not Churn Customers by MonthlyCharges

The chart above illustrates the distribution of Monthly Charges (monthly charges) between two customer groups in the telecommunications industry: Churn and Not Churn. The Not Churn group has the highest density in the lower cost range, from 20 to 30, indicating that these customers tend to stay subscribed when their monthly charges are low. In contrast, the Churn group shows high density in the mid-cost range, from 70 to 100, suggesting that higher monthly charges may be a contributing factor to the increased churn rate. This analysis shows that Monthly Charges is an important factor and can be used to build a churn prediction model, as the clear difference in distribution between the two groups can help identify customers at risk of churning.

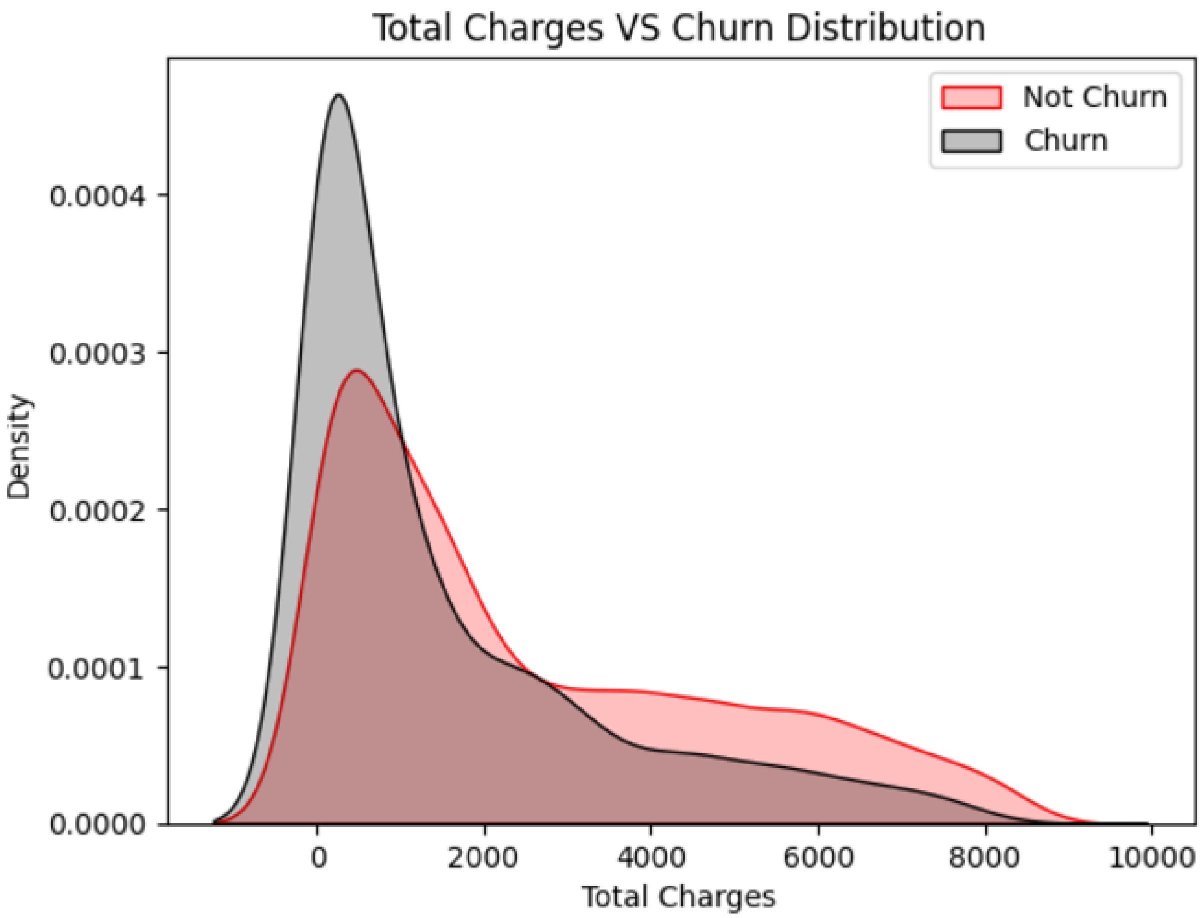

Figure 3.31: The Comparison of Churn and Not Churn Customers by TotalCharges

The chart above illustrates the distribution of Total Charges between two customer groups in the telecommunications industry: Churn and Not Churn. The Churn group has a high concentration of density in the low total charges range, from 0 to 2000, while the Not Churn group has a distribution that extends further towards higher values. This suggests that customers with low total charges are often new users who tend to churn more. In contrast, customers with higher total charges are usually long-term users who have been loyal to the service. This difference indicates that Total Charges is an important factor in analyzing and predicting churn, as it reflects the level of customer loyalty over time.

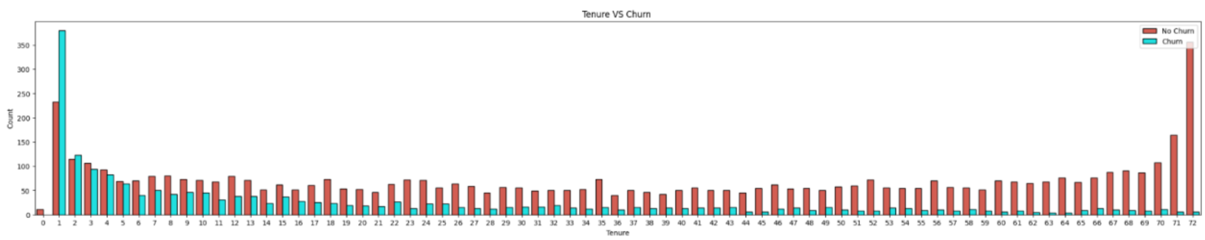

Figure 3.32: The Comparison of Churn and Not Churn Customers by tenure

This chart shows the relationship between customer tenure (the duration of service usage) and the likelihood of churn. The x-axis represents the number of months customers have been with the service, ranging from 0 to 72 months, while the y-axis represents the corresponding number of customers. Two different colors are used to represent the customer status: red for customers who did not churn (No Churn) and blue for customers who churned (Churn).

Observations show that customers with short tenures (less than 3 months) have a higher churn rate compared to other groups, particularly in the first month. The churn rate gradually decreases after the first few months, and the majority of customers with long tenures (around 70-72 months) tend to stay with the service. This suggests that customers are more likely to churn in the early months, while those who have been with the service for a longer time are less likely to leave.

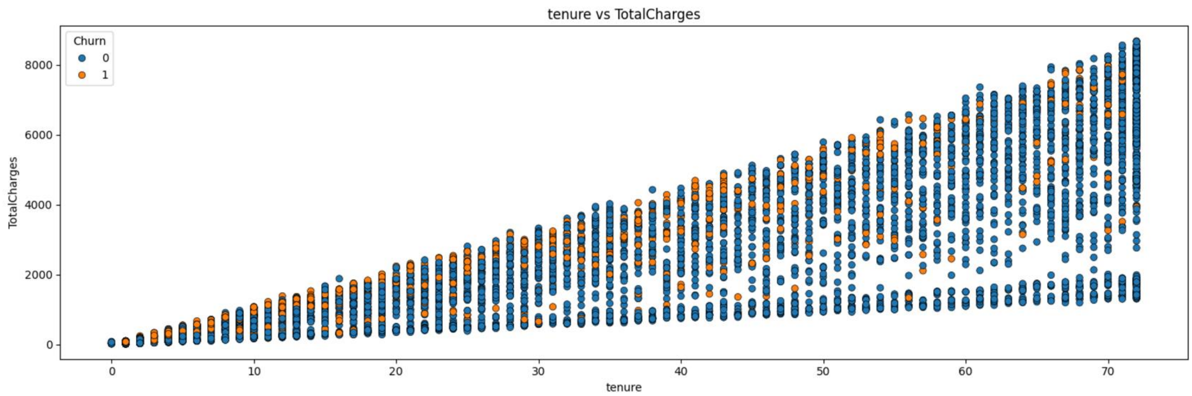

Figure 3.33: The Scatter Plot of tenrure and TotalCharges

This chart shows the relationship between customer tenure (the duration of service usage) and the total amount spent by customers (TotalCharges), while distinguishing between two groups of customers: those who churn and those who do not.

- The x-axis represents the tenure (service usage duration) ranging from 0 to 72 months.
- The y-axis represents the total amount spent by customers (TotalCharges).

- The color indicates the customer status: blue for customers who did not churn and orange for customers who churned.

Overall, total charges increase with the duration of service usage, with the data points forming a distinct upward trend. Customers who churn (orange) appear across most tenure levels, but the churn rate seems to be higher among customers with shorter tenures. Long-term customers, with higher total charges, tend to stay with the service (blue).

This suggests that long-term customers are more likely to be loyal, while newer or short-term customers are more likely to churn.

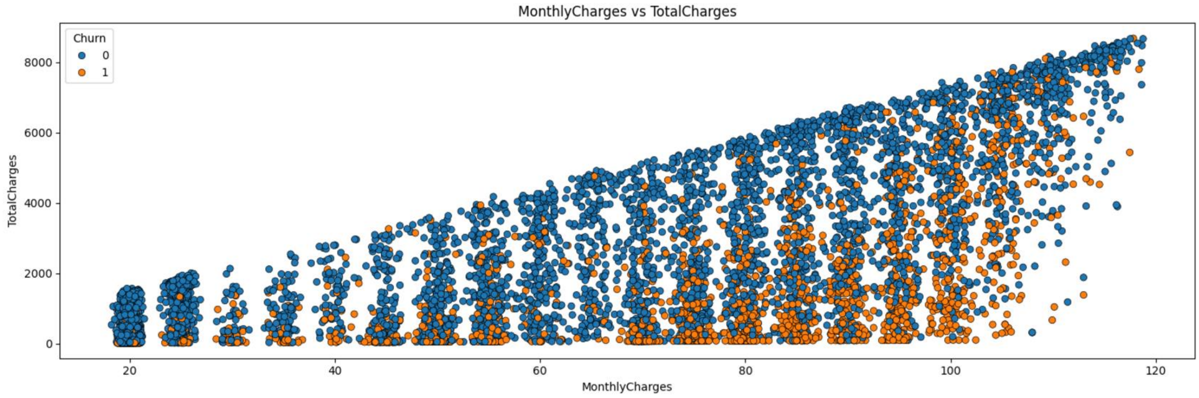

Figure 3.34: The Scatter Plot of MonthlyCharges and TotalCharges

This chart shows the relationship between monthly charges (MonthlyCharges) and total charges paid by customers (TotalCharges), while distinguishing between two groups of customers: those who churn and those who do not.

- The x-axis represents the monthly charges paid by customers (MonthlyCharges), ranging from 20 to around 120.

- The y-axis represents the total amount paid by customers over the entire duration of their service usage (TotalCharges).

- The color indicates the customer status: blue represents customers who did not churn, and orange represents customers who churned.

Observing the chart, we can see that the total charges paid by customers (TotalCharges) increase with monthly charges (MonthlyCharges), forming an upward trend. Customers with lower monthly charges tend to have lower accumulated total charges, while customers with higher monthly charges have higher total charges.

Additionally, the churn rate (orange) appears to be higher at higher monthly charge levels. This suggests that high monthly charges may be a factor contributing to customer churn, while customers paying lower monthly charges tend to be more loyal.

3.2.2.2. Data preprocessing

```
cols = list(df1.columns)
cols.remove('Churn')

x = df1.loc[:,cols]
y = df1.loc[:,'Churn']


imputer = SimpleImputer(strategy='mean')
x = imputer.fit_transform(x)

over = SMOTE(sampling_strategy = 1)

x1,y1 = over.fit_resample(x,y)
print("Class distribution before SMOTE:", Counter(y))
print("Class distribution after SMOTE:", Counter(y1))
```

```
Class distribution before SMOTE: Counter({0: 5174, 1: 1869})
Class distribution after SMOTE: Counter({0: 5174, 1: 5174})
```

Figure 3.35: Upsampling using Smote

The code above performs data preprocessing and balancing for a classification problem. First, it initializes a list of columns from the DataFrame df1 and removes the 'Churn' column since it is the target variable. Then, it separates the input data (features) and the target labels from the DataFrame, assigning them to variables x and y, respectively.

Next, the code uses SimpleImputer to handle missing values in the input data. Specifically, the imputation method of filling missing values with the mean (strategy='mean') is applied to the columns in x, ensuring that there are no missing values in the input data before it is fed into the machine learning model.

Then, the code applies the SMOTE (Synthetic Minority Over-sampling Technique) method to balance the dataset. SMOTE generates synthetic samples for the minority class to increase its sample size, helping balance the classes in the classification problem. In this case, sampling_strategy=1 means the number of samples in the minority class will be oversampled to match the number of samples in the majority class.

Finally, the code prints the class distribution in the dataset before and after applying SMOTE. This step helps verify whether SMOTE has successfully balanced the data, with the ultimate goal of ensuring the classification model can learn effectively and accurately from both classes.

```
x_train, x_test, y_train, y_test = train_test_split(x1, y1, test_size = 0.20, random_state = 2)
# Initialize MinMaxScaler
scaler = MinMaxScaler()

# Apply MinMaxScaler to the training and testing data
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
```

Figure 3.36: Min-max scaler

The code above performs important steps in the data preparation process for a machine learning model. First, the data is split into two parts: one for training the model and one for testing the model. Specifically, the train_test_split function from the sklearn.model_selection library is used to divide 80% of the data for training and 20% for testing, with the random_state=2 parameter to ensure the data split is reproducible each time the code is run.

Next, a MinMaxScaler object from the sklearn.preprocessing library is initialized to standardize the data. The MinMaxScaler transforms the input data values so that they fall within the range of 0 to 1, making machine learning algorithms more efficient, especially those sensitive to the scale and range of the data, such as artificial neural networks (ANN) or SVM.

Finally, the fit_transform method is applied to the training data to compute the scaling parameters (such as the min and max values), and then the scaling is applied to the training dataset. For the test data, the transform method is used to apply the scaling without recalculating the parameters, ensuring consistency between the training and test datasets. This process helps fully and effectively prepare the data before feeding it into the model training.

3.3 Model Training

3.3.1 XGBoost

```
# 1. Huấn luyện mô hình XGBoost
# ---------------------------------------
xgboost_model = xgb.XGBClassifier(use_label_encoder=False, eval_metric='logloss')
xgboost_model.fit(x_train, y_train)

# Dự đoán và đánh giá mô hình XGBoost
xgboost_predictions = xgboost_model.predict(x_test)
print("XGBoost Model Accuracy:", accuracy_score(y_test, xgboost_predictions))
print("Classification Report:\n", classification_report(y_test, xgboost_predictions))
```

Figure 3.37: XGBoost model

The code above performs the training and evaluation of an XGBoost model for a classification problem. First, the XGBoost model is initialized with the parameters use_label_encoder=False to avoid using the default label encoding and eval_metric='logloss' to use the log loss metric during training. After that, the model is trained on the training data x_train and labels y_train using the fit() method.

Once the training is complete, the model is used to predict results on the test set (x_test). The predicted results are stored in the variable xgboost_predictions. Next, the model is evaluated using two important metrics: accuracy and a classification report. Accuracy is calculated by comparing the predicted results with the actual labels y_test, while the classification report provides detailed metrics such as Precision, Recall, and F1-Score for each class in the classification problem.

In summary, this code not only trains the XGBoost model but also evaluates its performance using key metrics, helping to assess the model's accuracy and classification capability on the test data.

3.3.2 ANN

```
# 2. Huấn luyện mô hình ANN
np.random.seed(42)
tf.random.set_seed(42)
random.seed(42)
# ---------------------------------------
ann_model = Sequential([
    Dense(64, input_dim=x_train.shape[1], activation='relu'),
    Dropout(0.5),
    Dense(32, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])

ann_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
ann_model.fit(x_train, y_train, epochs=50, batch_size=32, validation_split=0.2)
```

Figure 3.38: ANN model

The code above performs the training and evaluation of an Artificial Neural Network (ANN) model for a binary classification problem. First, random values are reset using the functions np.random.seed(42), tf.random.set_seed(42), and random.seed(42) to ensure that the training process is reproducible.

The ANN model is built using the Sequential class from the tensorflow.keras.models library. The model consists of three Dense layers with the following characteristics:

- The first layer has 64 units and uses the ReLU activation function (activation='relu'), where the input shape is determined by x_train.shape[1].
- The second layer has 32 units and also uses the ReLU activation function.
- The final layer has a single unit with the sigmoid activation function (activation='sigmoid'), which is suitable for binary classification problems, as the sigmoid function returns a value in the range (0, 1), representing the probability of belonging to the positive class.

Each Dense layer is followed by a Dropout layer with a 50% rate (Dropout(0.5)), which helps mitigate overfitting by randomly dropping a portion of the units during training.

Once the model architecture is defined, the model is compiled with the Adam optimizer and the binary crossentropy loss function (binary_crossentropy) for binary classification tasks. The model is trained using the fit() method, with the training data x_train and labels y_train, for 50 epochs and a batch size of 32. The validation_split=0.2 parameter indicates that 20% of the training data will be used for validation during training.

Finally, the model is evaluated using the evaluate() method with the test data x_test and labels y_test. The results of this evaluation, including the loss and accuracy of the model on the test set, are stored in the variable ann_accuracy and printed to the screen.

In summary, this code builds, trains, and evaluates an ANN model for binary classification, with Dense and Dropout layers to optimize learning performance and reduce overfitting.

3.3.3 DNN

```
# 3. Huấn luyện mô hình DNN
np.random.seed(42)
tf.random.set_seed(42)
random.seed(42)
# ---------------------------------------
dnn_model = Sequential([
    Dense(128, input_dim=x_train.shape[1], activation='relu'),
    Dropout(0.4),
    Dense(64, activation='relu'),
    Dropout(0.4),
    Dense(32, activation='relu'),
    Dropout(0.4),
    Dense(1, activation='sigmoid')
])

dnn_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
dnn_model.fit(x_train, y_train, epochs=100, batch_size=32, validation_split=0.2)

# Đánh giá mô hình DNN
dnn_loss, dnn_accuracy = dnn_model.evaluate(x_test, y_test)
print("DNN Model Accuracy:", dnn_accuracy)
```

Figure 3.39: DNN model

The code above trains and evaluates a Deep Neural Network (DNN) model for a binary classification problem. First, random values are reset using the functions np.random.seed(42), tf.random.set_seed(42), and random.seed(42) to ensure that the training process is reproducible.

The DNN model is built using the Sequential class from the tensorflow.keras.models library, consisting of four Dense layers. The model structure includes:

- The first layer has 128 units and uses the ReLU activation function (activation='relu'). The input size of this layer is determined by the length of the training data x_train.shape[1].
- The second layer has 64 units and uses the ReLU activation function.
- The third layer has 32 units and uses the ReLU activation function.
- The final layer has one unit and uses the sigmoid activation function (activation='sigmoid'), suitable for binary classification as the sigmoid function returns a value in the range (0, 1), representing the probability of belonging to the positive class.
- Each Dense layer is followed by a Dropout layer with a rate of 40% (Dropout(0.4)), which helps reduce overfitting by randomly dropping some of the units during training.

The model is compiled with the Adam optimizer and the binary_crossentropy loss function, which is suitable for binary classification tasks. The model is trained using the fit() method, with the training data x_train and labels y_train. The number of epochs is set to 100, and the batch size is 32. The validation_split=0.2 parameter indicates that 20% of the training data will be used for validation during training.

Finally, the model is evaluated using the evaluate() method with the test data x_test and labels y_test. The results of this method, including the loss and accuracy of the model on the test set, are stored in the variable dnn_accuracy and printed to the screen.

In summary, this code builds, trains, and evaluates a DNN model for binary classification, using Dense and Dropout layers to optimize learning performance and reduce overfitting, while evaluating the model based on its accuracy on the test set.

3.4. Model result

3.4.1 XGBoost

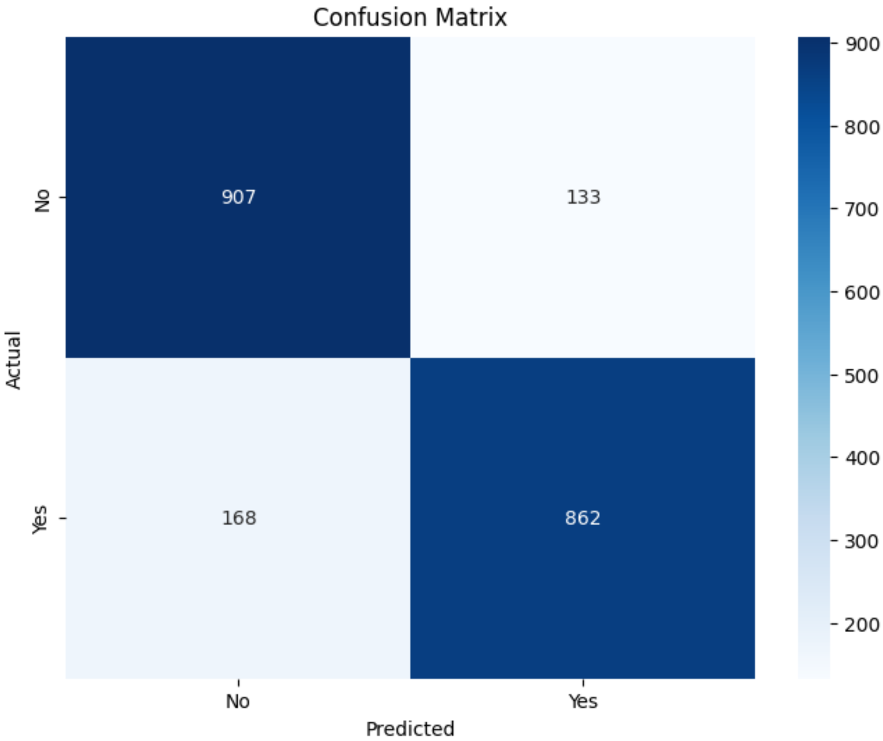

Figure 3.40: XGBoost Confusion Matrix

The confusion matrix of the model provides an overview of its predictive capabilities. Specifically, the model correctly predicted 907 customers who did not churn (True Negative - TN) and correctly identified 862 customers who actually churned (True Positive - TP). These numbers reflect the model's ability to reliably identify both customer groups. However, there are some notable limitations. There are 133 False Positive (FP) cases, meaning that customers were incorrectly predicted to stay while they actually churned. This not only wastes resources but could also result in missing important opportunities to retain customers. Additionally, with 168 False Negative (FN) cases, the model incorrectly predicted that some customers who churned would stay. This is particularly concerning because these customers may not receive timely retention strategies, reducing the overall effectiveness of intervention measures. These limitations suggest the need to continue improving the accuracy and efficiency of the model to better achieve the goal of identifying churned customers.

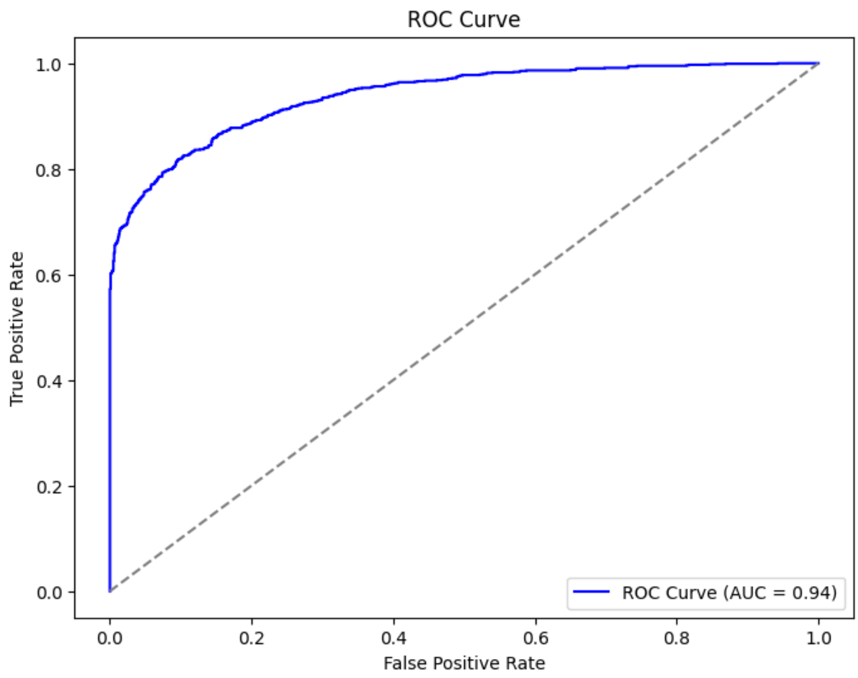

Figure 3.41: XGBoost ROC Curve

The ROC (Receiver Operating Characteristic) curve shows that the XGBoost model performs very effectively in distinguishing between the two customer groups: those who churn and those who do not. The curve has a steep slope at the top-left corner, demonstrating the model's accuracy in classification. Notably, the AUC (Area Under the Curve) score reaches 0.94, which is close to the ideal value of 1. This confirms that the model excels in accurately identifying customers at risk of churning as well as those who will stay, even when facing potentially imbalanced data. With this capability, the model has great potential in supporting customer retention strategies, allowing businesses to focus on the right target audience.

3.4.2 ANN

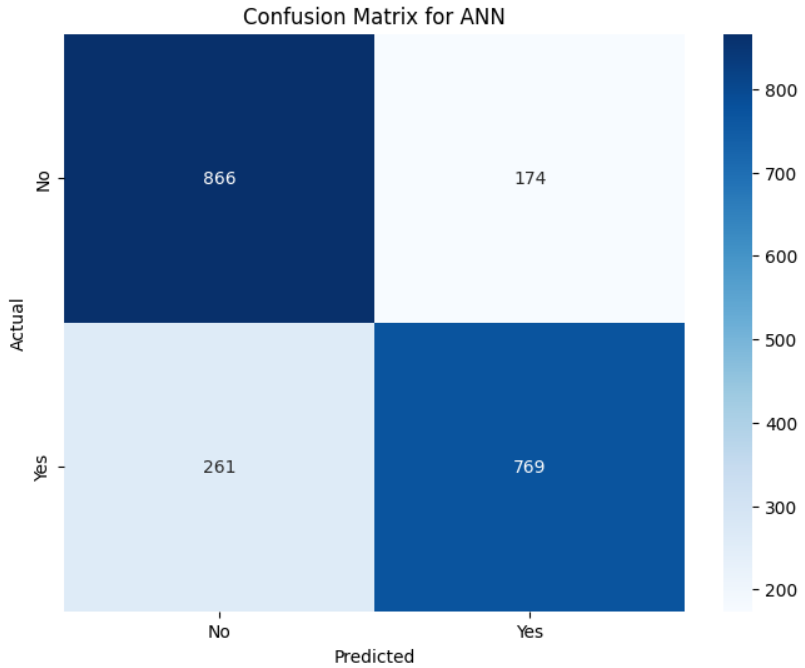

Figure 3.42: ANN Confusion Matrix

The confusion matrix of the Artificial Neural Network (ANN) model shows its prediction performance with the following specific values: 866 True Negative (TN) cases, meaning the number of customers who did not churn and were correctly predicted, and 769 True Positive (TP) cases, meaning the number of customers who churned and were correctly identified. However, there are 174 False Positive (FP) cases, meaning customers who were incorrectly predicted to stay when they actually churned. Additionally, there are 261 False Negative (FN) cases, representing customers who were incorrectly predicted to churn when they actually stayed. These numbers indicate that the ANN model has relatively good classification ability, but there are still notable limitations. In particular, the high False Negative rate could make it difficult to implement effective customer retention strategies, leading to missed opportunities for customers who genuinely need support. Furthermore, the False Positive rate wastes resources by focusing on customers who are not at risk of churning.

Compared to the XGBoost model, the ANN model appears to perform slightly worse in accurately predicting customer churn. This may suggest the need to further optimize the network architecture or fine-tune hyperparameters to improve performance.

Additionally, applying advanced data processing techniques or data balancing might help the ANN model perform better in this prediction task.

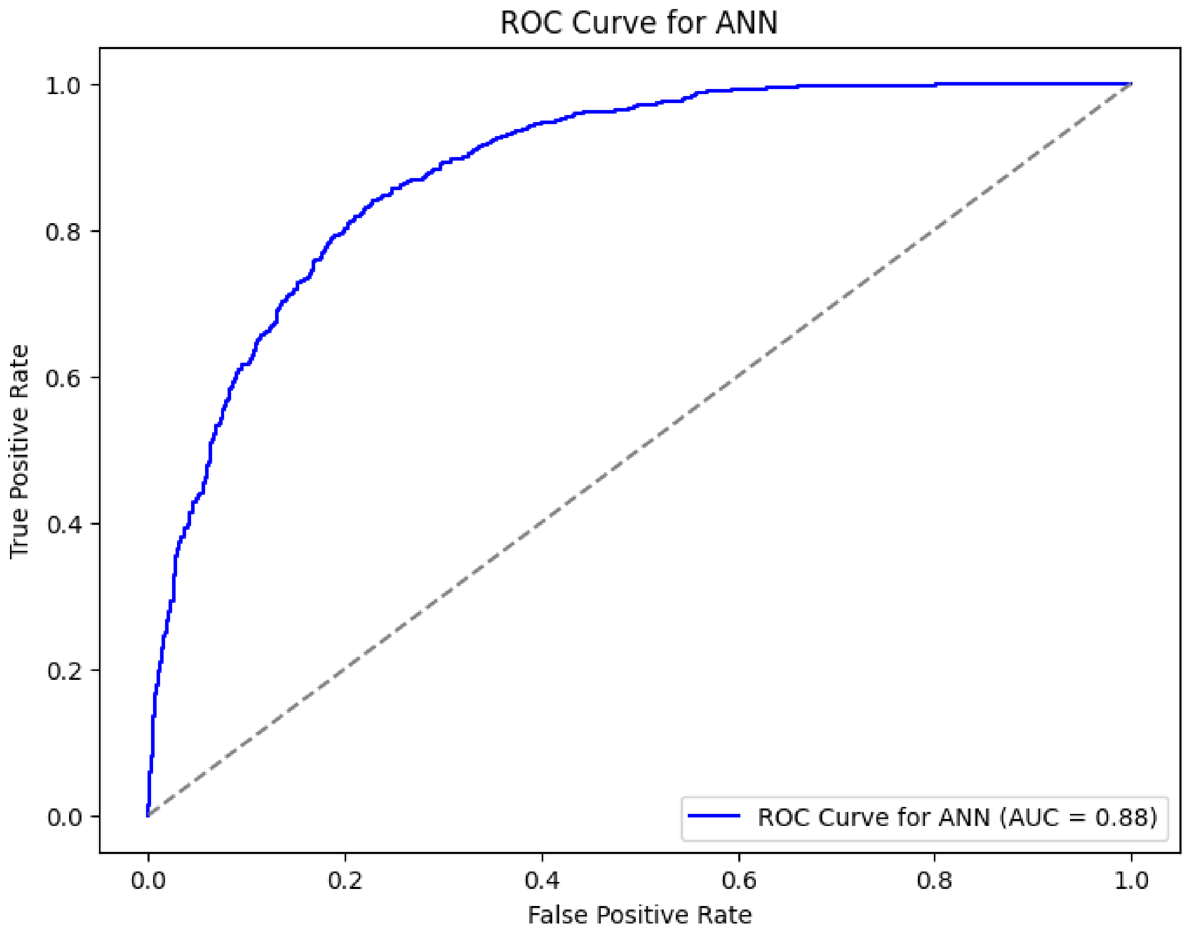

Figure 3.43: ANN ROC Curve

The ROC curve of the Artificial Neural Network (ANN) model shows that the model achieves an AUC (Area Under the Curve) value of 0.88, indicating good classification performance. With the ROC curve clearly positioned above the random diagonal line, the model demonstrates its ability to effectively distinguish between classes (e.g., positive and negative classes). An AUC value close to 1 indicates that the model balances well between the True Positive Rate (TPR) and the False Positive Rate (FPR). However, there is still potential for performance improvement, such as through hyperparameter tuning, expanding the training dataset, or applying techniques to reduce overfitting. Overall, these results show that the ANN model performs consistently and is suitable for classification tasks requiring high accuracy.

3.4.3 DNN

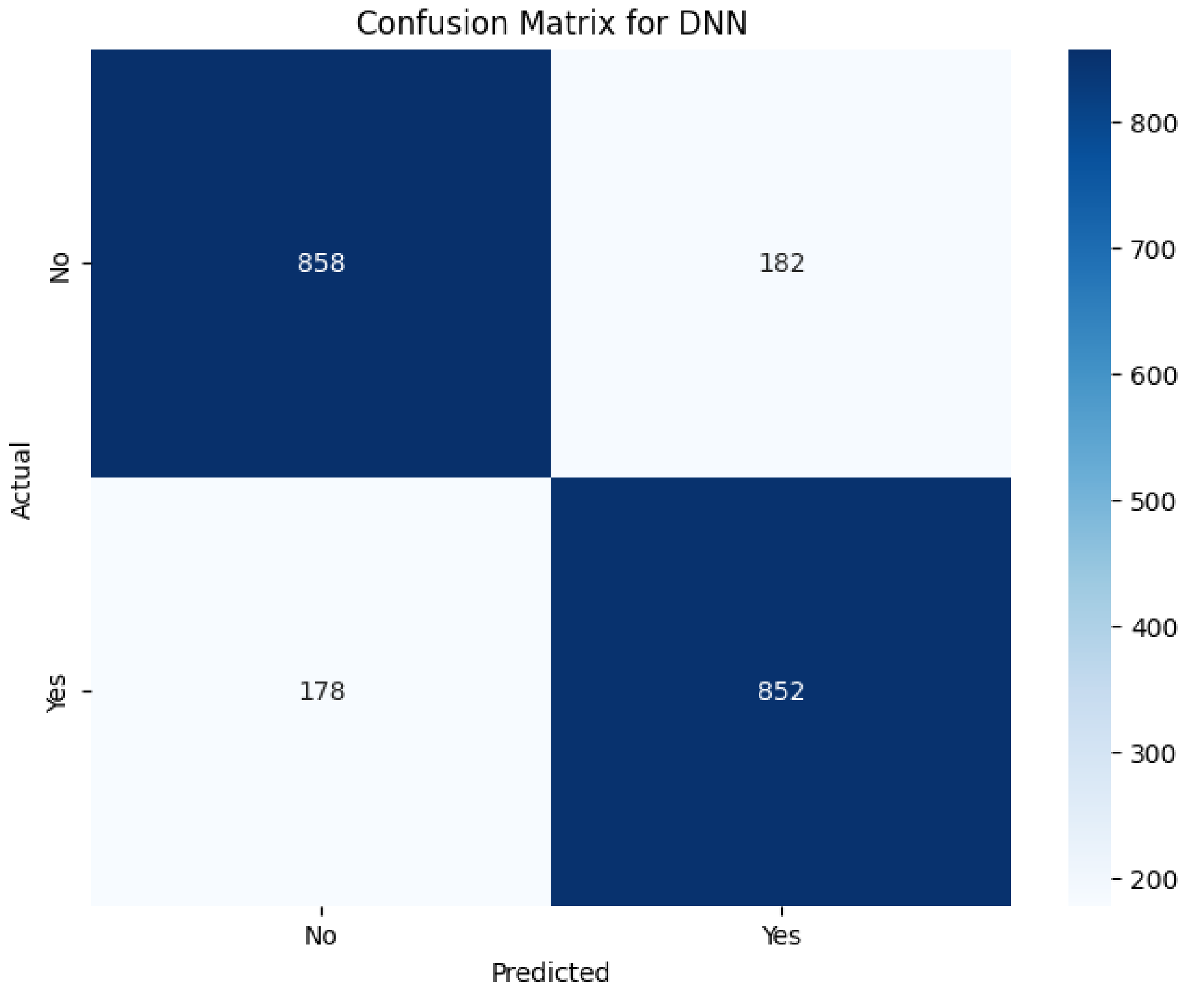

Figure 3.44: DNN Confusion Matrix

The confusion matrix of the Deep Neural Network (DNN) model shows overall good performance in classification. The model correctly predicted 858 cases as "No" (True Negative) and 852 cases as "Yes" (True Positive), while there were 182 incorrect predictions of "No" as "Yes" (False Positive) and 178 incorrect predictions of "Yes" as "No" (False Negative). This indicates that the model maintains a relatively good balance between the classes, with a high prediction accuracy in both directions. However, the number of incorrect predictions in both classes is still significant, suggesting that the model could benefit from further improvement, such as through hyperparameter optimization or applying data processing techniques to reduce classification errors. Overall, the DNN model is performing reliably, but there is room for improvement in accuracy.

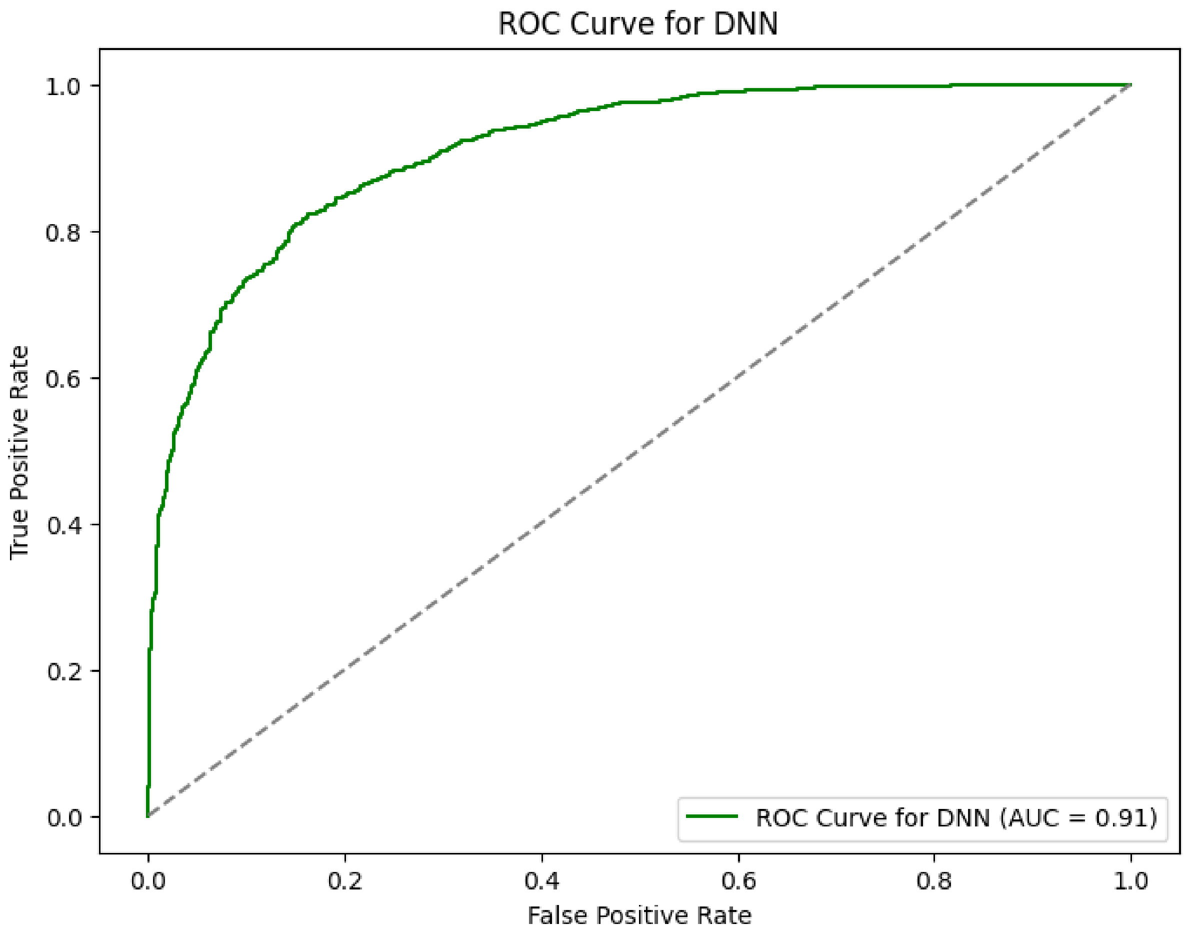

Figure 3.45: DNN ROC Curve

The ROC curve of the Deep Neural Network (DNN) model shows an AUC (Area Under the Curve) value of 0.91, indicating that the model performs very well in distinguishing between classes. The ROC curve (green) is clearly positioned above the random diagonal line, showing that the model has significantly higher classification performance than random chance. An AUC value greater than 0.9 indicates that the model has good accuracy and sensitivity in optimizing the True Positive Rate while keeping the False Positive Rate low. Overall, the DNN model not only outperforms the ANN model (based on AUC) but also demonstrates stability in handling complex data. However, there is still room for improvement, particularly in reducing the number of incorrect predictions, as observed in the confusion matrix earlier.

3.5 Evaluate the results of the models.

Model	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score	AUC Score
XGBoost	0.95	0.86	0.87	0.84	0.85	0.94
DNN	0.88	0.82	0.82	0.82	0.82	0.91
ANN	0.82	0.79	0.80	0.77	0.79	0.88

Figure 3.46: Results of XGBoost, DNN, and ANN models

The results show that the XGBoost model has the highest accuracy on both the training set (0.95) and the test set (0.86). Although there is a small difference between the two values (0.09), this indicates that the model learns well from the training data and generalizes well to the test data, despite possibly experiencing slight overfitting. Meanwhile, the Deep Neural Network (DNN) model achieves an accuracy of 0.88 on the training set and 0.82 on the test set, with a smaller gap (0.06), indicating a more stable generalization compared to XGBoost. Finally, the Artificial Neural Network (ANN) model has the lowest accuracy on both sets, reaching only 0.82 on the training set and 0.79 on the test set. The small gap (0.03) between these values reflects the generalization ability of ANN, but its learning efficiency from the training data is not as effective as the other models.

In terms of Precision, XGBoost continues to lead with the highest value (0.87), indicating that the model predicts positive cases most accurately among the three models. DNN achieves a Precision of 0.82, lower than XGBoost but still quite stable, while ANN reaches a Precision of 0.80, slightly higher than DNN but lower than XGBoost. This shows that all three models are capable of making accurate predictions, but XGBoost remains superior.

Regarding Recall, the XGBoost model also outperforms with the highest value (0.84), reflecting its better ability to detect positive cases compared to the other models. The DNN model achieves a Recall of 0.82, close to XGBoost, indicating similar performance in detecting positive cases. Meanwhile, ANN has the lowest Recall (0.77), indicating that this model tends to miss more positive cases compared to the other two models.

When considering the F1-Score, a measure that balances Precision and Recall, XGBoost continues to lead with a value of 0.85, demonstrating the best balance between these two factors. DNN achieves an F1-Score of 0.82, slightly lower than XGBoost but still high. ANN has the lowest F1-Score (0.79), indicating its overall performance is not as strong as the other two models.

Regarding the AUC Score, the XGBoost model achieves the highest value (0.94), demonstrating excellent class discrimination ability. DNN achieves an AUC Score of 0.91, continuing to show strong and reliable performance. ANN has the lowest AUC Score (0.88), but it is still within an acceptable range. However, the clear differences between the models indicate XGBoost's superiority in accurate classification.

In summary, XGBoost is the model with the best overall performance across most metrics, demonstrating excellent learning and generalization capabilities. It is the most suitable model for practical applications. DNN is also a viable option, with nearly equivalent performance and less overfitting. Meanwhile, ANN, although stable, has lower overall performance and needs improvement if it is to compete with the other two models.

CHAPTER 4: USING CAUSAL AI TO ANALYZE ATTRITION CAUSES

4.1 Identify Causal Factors Influencing Attrition Rate and Evaluate Causal Analysis Results

Causal graphs, grounded in Pearl's framework for causal inference, provide a visual representation of cause-and-effect relationships, facilitating targeted interventions. By leveraging such models, businesses can adopt a data-driven approach to understanding churn dynamics. The relationships identified enable prioritization of strategies, such as offering discounts for high-churn risk groups, improving technical support, or designing incentives for long-term contracts. From a methodological perspective, constructing these diagrams requires careful variable selection and assumptions about causal direction. Incorrect assumptions can lead to confounded results, undermining the reliability of the analysis. Validation through statistical methods such as propensity score matching or instrumental variables is essential to strengthen the causal claims suggested by the diagrams. The provided causal diagrams illustrate the relationships between various factors influencing customer churn. These graphs effectively depict how factors such as tenure, contract type, monthly charges, and other service-related attributes impact the likelihood of customer churn, enabling an exploration of causal relationships.

In the first graph, key predictors of churn include variables such as tenure, contract, monthly charges, and paperless billing. Notably, contract type serves as a mediating variable, linking multiple attributes like tenure and monthly charges to churn. The presence of a long-term contract often reduces the likelihood of churn, as customers are less likely to terminate agreements prematurely. Conversely, higher monthly charges could increase churn due to financial dissatisfaction, particularly among customers who perceive limited value for the cost.

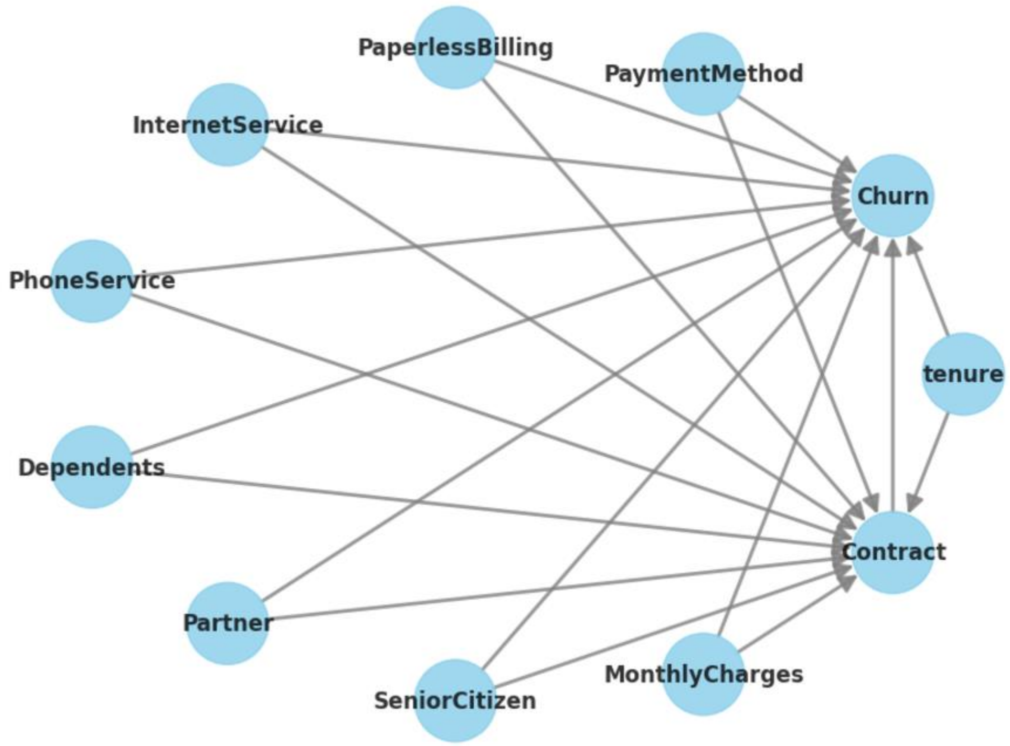

Figure 4.1: The Relationship of Attributes in Causal Model

The second graph extends this analysis, incorporating additional factors such as gender, online security, streaming services, and technical support. Services like OnlineSecurity, StreamingTV, and TechSupport directly contribute to churn, indicating that unmet expectations or poor service in these areas can drive customers to leave. Moreover, total charges and payment methods reflect financial factors that are pivotal in influencing customer retention. Customers with higher cumulative costs or those using payment methods prone to dissatisfaction (auto-payments without clear consent) are more likely to churn.

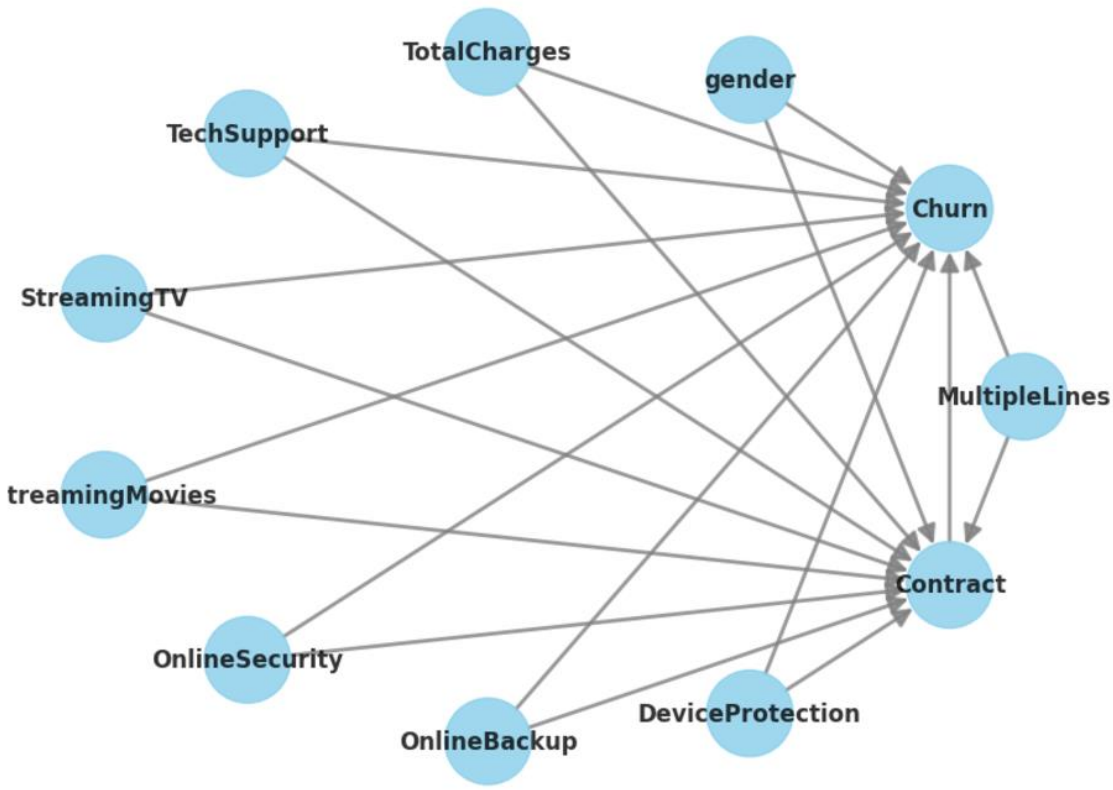

Figure 4.2: The Relationship of Attributes in Causal Model with Additional Attributes

4.1.1 Causal effect of contract types on customer churn: Backdoor adjustment analysis

The analysis employs a non-parametric approach to estimate the Average Treatment Effect (ATE), focusing on the causal relationship between the treatment variable, Contract, and the outcome, Churn. This estimation leverages Pearl's structural causal model, specifically utilizing the backdoor adjustment criterion. The backdoor estimand is derived under the assumption of unconfoundedness, whereby all relevant confounding variables are accounted for, ensuring that the treatment-outcome relationship is not biased by omitted variables. This assumption is critical, as it posits that, conditional on observed confounders, the potential outcomes are independent of the treatment assignment.

The expression of the backdoor estimand involves the partial derivative of the expected value of churn, conditioned on a set of observed variables including TechSupport, StreamingTV, OnlineBackup, DeviceProtection, OnlineSecurity, TotalCharges, MultipleLines, StreamingMovies, and gender. These variables serve as confounders that influence both the likelihood of selecting a specific contract type and the probability of customer churn. By conditioning on these confounders, the analysis isolates the causal effect of the contract on churn. The robustness of this approach hinges on the assumption that no unobserved confounders remain after adjustment for the included variables.

Instrumental variable (IV) and frontdoor adjustment methods were also explored but deemed infeasible within the constraints of the available dataset. The absence of valid instrumental variables precludes the use of IV estimation, as no variable meets the dual criteria of influencing the treatment without directly affecting the outcome except through the treatment. Similarly, the lack of mediators precludes the application of the frontdoor criterion, as no variable was identified that could sufficiently decompose the causal pathway between the contract and churn.

These limitations reinforce the appropriateness of the backdoor adjustment method in this context. The method's reliance on observed confounders aligns with the data's structural properties, providing a rigorous framework for estimating the causal effect of contract type on churn. This approach underscores the importance of selecting confounders grounded in domain knowledge to ensure the validity of the causal inference. By leveraging the backdoor criterion, the analysis not only estimates the direct effect of contract types on churn but also provides actionable insights for strategic interventions aimed at improving customer retention. Further studies could enhance this framework by incorporating additional data to enable the exploration of alternative causal pathways, such as IV and frontdoor adjustment, thereby broadening the scope of causal inference methods applicable to this problem.

4.1.2 Estimating the impact of contract types on customer churn using backdoor adjustment and machine learning

The causal effect of Contract on Churn was estimated using a backdoor adjustment approach within the framework of non-parametric Average Treatment Effect (ATE) estimation. The identified estimand assumes unconfoundedness, meaning that conditional on the observed covariates, the treatment assignment is independent of unobserved factors that could influence the outcome. This assumption is essential for isolating the causal impact of the treatment on the outcome.

The backdoor adjustment criterion was employed, with the treatment variable, Contract, and the outcome variable, Churn, conditioned on covariates such as TechSupport, StreamingTV, OnlineBackup, DeviceProtection, OnlineSecurity, TotalCharges, MultipleLines, StreamingMovies, and gender. These covariates serve as confounders, addressing potential biases arising from omitted variable confounding and ensuring that the causal pathway between the treatment and outcome is validly estimated.

The estimation was performed using a machine learning-based implementation of linear regression, incorporating Gradient Boosting Classifiers as models for treatment, outcome, and final predictions. This advanced method captures non-linear relationships and interactions between variables, improving the robustness of the causal estimate. The analysis targeted the average treatment effect (ATE) by comparing the outcome under two scenarios: the absence of the treatment (control value = 0) and the presence of the treatment (treatment value = 1).

The realized estimand reflects the relationship modeled as Churn ~ Contract + TechSupport + StreamingTV + OnlineBackup + DeviceProtection + OnlineSecurity + TotalCharges + MultipleLines + StreamingMovies + gender, emphasizing the importance of the included confounders. The estimated mean causal effect (ATE) of switching contract types was calculated to be approximately -0.1489. This negative value indicates a reduction in churn probability associated with the treatment, highlighting the potential benefits of specific contract interventions in reducing customer attrition.

This causal estimate provides actionable insights for decision-makers aiming to optimize contractual terms as a strategy to enhance customer retention. The model's reliance on Gradient Boosting Classifiers ensures that complex interactions and non-linear patterns in the data are effectively captured, strengthening the validity and practical utility of the findings.

4.1.3 Validating Causal estimates with sensitivity analysis: Robustness of contract effects on customer churn

The robustness of the causal estimate was evaluated through the application of a sensitivity analysis using the random common cause refutation method. This approach involves introducing a hypothetical, random confounder into the model to examine whether the estimated causal effect remains consistent despite the inclusion of an unobserved variable that could potentially influence both the treatment and the outcome.

The original causal estimate, derived through the backdoor adjustment method, was approximately -0.1489. After introducing the random common cause, the recalculated effect remained virtually unchanged at -0.1489, with a slight variation attributable to the stochastic nature of the introduced confounder. This minimal change in the effect size suggests that the original estimate is robust to the inclusion of additional unobserved variables.

Moreover, the high p-value of 0.92 reinforces the conclusion that the observed effect is not significantly altered by the presence of a random common cause, indicating that the causal relationship between the treatment and the outcome is unlikely to be confounded by omitted variables within this context.

The results of this refutation test provide strong evidence supporting the validity of the identified causal pathway and the estimated treatment effect. The negligible impact of the introduced random confounder underscores the strength of the unconfoundedness assumption and highlights the reliability of the causal inference model in capturing the true effect of the treatment on the outcome.

4.1.4 Predicting customer churn with machine learning: Feature importance analysis using XGBClassifier

A machine learning approach was utilized to predict the likelihood of customer churn, with a Random Forest model implemented via the XGBClassifier. The model was trained on a subset of the data (x_train and y_train) using 100 estimators, and it was fitted with a random state for reproducibility. To interpret the predictive capacity of the features included in the model, a permutation importance analysis was conducted. This analysis evaluates the relative importance of each feature by measuring the change in model performance when the values of a given feature are randomly shuffled.

The feature importance results reveal that tenure and MonthlyCharges have the highest influence on predicting churn, with importance weights of 0.0440 and 0.0438, respectively. The variable Contract follows closely, indicating that the type of customer contract significantly impacts churn behavior. Features such as PaymentMethod and Partner exhibit moderate importance, while other variables, including StreamingMovies, StreamingTV, and PhoneService, demonstrate comparatively lower predictive contributions.

The permutation importance values also include variability measures (e.g., ±0.0107 for tenure), which highlight the robustness of these estimates. The results provide a clear ranking of feature contributions, offering actionable insights for business decision-making. For example, retention strategies could prioritize targeting customers based on their tenure, monthly charges, and contract type, as these factors play a pivotal role in churn prediction.

Additionally, features with minimal or negative importance, such as InternetService, indicate that their inclusion may not significantly enhance the model's predictive performance. This underscores the potential for further model optimization by reducing dimensionality and focusing on more impactful predictors. Overall, this analysis provides a transparent understanding of the model's decision-making process and supports strategic planning to mitigate customer churn effectively.

4.1.5 Decision tree analysis for customer churn: Hierarchical insights and strategic applications

Lastly, the decision tree visualization is a hierarchical model used for binary classification, likely in the context of customer churn prediction. It operates by recursively partitioning the dataset based on feature values to achieve maximum separation between the target classes. Each node in the tree represents a decision boundary based on a specific feature and its threshold value. The ultimate goal of the model is to reduce impurity, measured by the Gini index, at each split.

At the top of the tree is the root node, which represents the entire dataset. Here, the feature Contract is selected for the first split, indicating its significant role in predicting the target variable. The Gini impurity at this level is 0.5, signifying a near-equal distribution of the two classes. This initial split divides the dataset into two primary groups: customers with shorter contracts and those with longer contracts. The separation reduces the Gini index, reflecting an improvement in classification purity.

Moving further into the left subtree, customers with shorter contracts are further analyzed. Features such as PaperlessBilling and TechSupport play a role in refining the classification. For instance, the feature TechSupport is highly effective in splitting the data, as evidenced by the significant reduction in the Gini index from 0.432 to 0.372. This suggests that the availability of technical support services strongly correlates with the likelihood of customers staying or leaving.

On the right subtree, customers with longer contracts are examined. The feature MonthlyCharges is pivotal here, where customers with lower monthly charges (≤ 0.75) demonstrate a much lower Gini impurity (0.207). This implies that affordability is a critical factor in determining customer retention for this segment. Subsequent splits, such as those based on OnlineSecurity, further refine the classification, with some terminal nodes achieving extremely low Gini values, indicating highly pure subsets where one class dominates.

The distribution of samples at each node provides additional insights into the dataset. Nodes with large sample sizes highlight general patterns in customer behavior, while nodes with smaller sizes represent more specific and potentially over-specialized groups. While deeper splits improve classification accuracy on the training data, they also risk overfitting, where the model may lose its ability to generalize to new data.

The interpretability of this decision tree makes it particularly useful for understanding the underlying factors driving customer behavior. Features like Contract, MonthlyCharges, and PaperlessBilling indicate that financial considerations are paramount, while variables like TechSupport and OnlineSecurity reflect the importance of service quality and customer support in influencing churn.

In terms of model performance, the decision tree demonstrates a clear capability to distinguish between classes, as evidenced by the progressively lower Gini indices across splits. However, this simplicity comes with certain limitations. Continuous variables like MonthlyCharges are discretized into thresholds, potentially leading to a loss of nuance. Additionally, deeper sections of the tree may overfit the data, capturing noise rather than meaningful patterns.

To enhance the model's performance and generalizability, pruning techniques could be applied to reduce overfitting, ensuring better adaptation to new data. Analyzing feature importance scores derived from the tree would provide further clarity on the most influential variables. Evaluating the model on a test dataset would confirm its predictive power and highlight any limitations, while integration into a customer relationship management system could enable real-time predictions and personalized recommendations.

This decision tree serves as a robust tool for understanding complex relationships in the data, balancing predictive accuracy with interpretability. It provides clear pathways for actionable strategies, enabling informed decisions based on well-structured insights.

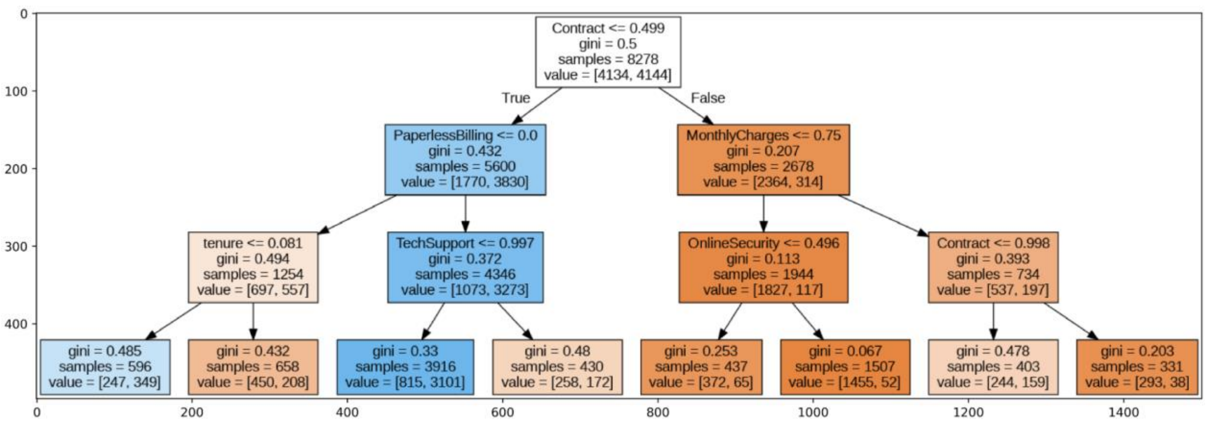

Figure 4.3: The Decision Tree of Considering Contract as Mediating Variable

4.2 Propose Customer Retention Strategies

As the owner of the telecommunications company, informed by the predictive analysis and causal inference insights, the development of strategic initiatives would focus on reducing customer churn, enhancing customer satisfaction, and fostering long-term loyalty. These strategies would address the most influential factors identified in the model and capitalize on the causal relationships revealed by the data-driven analysis.

A central strategy would involve optimizing contract structures. Given the significant impact of contract type on churn rates, transitioning customers from short-term contracts to longer-term options could stabilize the customer base. This could be achieved through targeted promotions, such as offering discounts or additional benefits for customers who agree to extend their contracts. For instance, bundling long-term contracts with premium services, like enhanced technical support or higher-tier streaming options, would add perceived value and reduce churn propensity.

Additionally, pricing strategies would address the notable influence of monthly charges. Offering flexible payment plans, loyalty rewards for long-tenured customers, and incentives such as rebates for consistent payments could mitigate financial concerns, particularly for cost-sensitive segments. Tailored recommendations based on customers' usage patterns would ensure that pricing aligns with individual needs, enhancing perceived fairness and value.

Investments in customer service and technical support would also be pivotal. Features such as tech support, online security, and device protection, although moderately influential, play a critical role in ensuring a seamless customer experience. Enhancing the quality and accessibility of these services—through proactive outreach, faster response times, and self-service options—would strengthen customer trust and satisfaction.

Efforts to improve customer tenure, identified as the most critical factor, would focus on engagement initiatives that create a sense of belonging and community. Loyalty programs that reward longevity, personalized communications that acknowledge milestones, and value-added services tailored to customers’ preferences would foster stronger relationships. For example, providing exclusive access to early product releases or offering tailored promotions for long-term customers could significantly enhance retention.

To address the influence of payment methods on churn, transitioning customers to more stable and reliable billing options would be critical. Encouraging the adoption of automated payments or digital wallets, combined with clear communication about the benefits of these methods such as ease of use or reduced risk of late payments, would reduce churn caused by billing frustrations.

Operationally, leveraging the insights from causal analysis and feature importance would inform the deployment of predictive analytics to identify at-risk customers in real time. This would enable the company to design proactive retention interventions, such as personalized offers or direct engagement with dissatisfied customers before they decide to leave.

Finally, the company would need to maintain an iterative approach, consistently reviewing the model outputs and refining strategies as new data becomes available. Incorporating customer feedback into these strategies would ensure alignment with evolving customer preferences and reinforce the company's commitment to customer-centricity. This comprehensive approach, rooted in data-driven decision-making, would position the company to achieve sustained growth, enhanced loyalty, and reduced churn rates.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

This study highlights the transformative potential of Causal AI in addressing the persistent challenge of customer churn in the telecommunications sector. By uncovering the root causes of churn, such as short-term contracts, service quality issues, and low engagement with value-added services, the research offers actionable insights that go beyond traditional correlation-based approaches. The findings demonstrate that integrating causal analysis into predictive modeling not only enhances accuracy but also provides a clearer roadmap for designing effective retention strategies. Ultimately, this approach enables companies to shift from reactive measures to proactive interventions, significantly improving customer loyalty and business performance.

5.2 Achievements and Limitations

The research achieved several important milestones, including the successful application of causal models to predict and explain churn dynamics. The use of counterfactual simulations provided a deeper understanding of how specific interventions could influence customer behavior, offering telecom companies a practical toolkit for decision-making. However, the study also faced limitations. The reliance on a single dataset constrained the generalizability of the findings, and the complexity of causal models posed challenges in communicating results to non-technical stakeholders. Additionally, implementing the recommended strategies may require significant organizational changes, which could pose barriers to adoption.

5.3 Future works

Future research should aim to address these limitations by incorporating more diverse datasets from multiple telecom providers to enhance model robustness and applicability. Expanding the scope of causal models to include external factors, such as market competition and economic conditions, could provide a more comprehensive understanding of churn dynamics. Advancements in technology, such as graph neural networks, offer opportunities to refine causal modeling further and improve predictive accuracy.

Additionally, the practical application of these models can be scaled by developing intuitive tools and dashboards that allow decision-makers to visualize and act on causal insights effectively. Pilot programs for targeted retention campaigns can validate the strategies proposed in this research, paving the way for broader implementation. By continuing to innovate and refine causal analysis techniques, the telecommunications industry can set new standards in customer retention and business resilience.

References

1. Saha, L., Tripathy, H. K., Gaber, T., El-Gohary, H., & El-kenawy, E. S. M. (2023). Deep churn prediction method for telecommunication industry. Sustainability, 15(5), 4543.

2. Hason Rudd, D., Huo, H., & Xu, G. (2022). Improved churn causal analysis through restrained high-dimensional feature space effects in financial institutions. Human-Centric Intelligent Systems, 2(3), 70-80.

3. Ceballos, D., López-Álvarez, D., Isaza, G., Tabares-Soto, R., Orozco-Arias, S., & Ferrin, C. D. (2019). A machine learning-based pipeline for the classification of CTX-M in metagenomics samples. Processes, 7(4), 235.

4. Data Science 365. (n.d.). Overview of a neural network's learning process. Medium. Retrieved November 18, 2024, from https://medium.com/data-science-365/overview-of-a-neural-networks-learning-process-61690a502fa

5. Papernot, N., McDaniel, P., Wu, X., Jha, S., & Swami, A. (2016, May). Distillation as a defense to adversarial perturbations against deep neural networks. In 2016 IEEE symposium on security and privacy (SP) (pp. 582-597). IEEE.

6. Arakawa, R., Takamichi, S., & Saruwatari, H. (2019). Implementation of DNN-based real-time voice conversion and its improvements by audio data augmentation and mask-shaped device. Proc. SSW10, 93-98.

7. Novotny, O., Plchot, O., Matejka, P., & Glembek, O. (2018). On the use of dnn autoencoder for robust speaker recognition. arXiv preprint arXiv:1811.02938.

8. Barsotti, A. (2024). PREDICTION AND CAUSAL ANALYSIS OF CHURN IN THE TELECOMMUNICATION DOMAIN.

9. Office of Electronic Communications (UKE). (2024). Report on the telecommunications market in 2023. Warsaw, Poland. Retrieved from https://www.uke.gov.pl/

10. Durvasula, S. (2024). Advancing AI: The role of causality in decision-making systems. Journal of Artificial Intelligence Research, 58(2), 112-130.

11. Latos, P., et al. (2024). Causal reasoning in AI: Challenges and opportunities. Proceedings of the International Conference on AI Applications, 45(7), 322-340.

12. Jaimini, P., et al. (2024). Neurosymbolic approaches in causal AI systems. Expert Systems with Applications, 134, 106-115.

13. Parkar, M., et al. (2024). Exploring causality in high-stakes AI applications. AI Ethics Journal, 19(1), 45-62.

14. Rawal, S., et al. (2024). Integrating causality into explainable AI frameworks. Machine Learning Review, 72(5), 211-228.

15. Ayesha, S., & Ahamed, R. (2024). Explainable AI in healthcare: Challenges and solutions. Journal of Medical Informatics, 11(3), 184-196.

16. Thakur, P. (2024). Enhancing transparency in AI systems with explainability. Artificial Intelligence and Society, 9(4), 45-57.

17. Kumar, R., et al. (2024). Feature importance analysis in explainable AI models. International Journal of Data Science, 18(6), 315-328.

18. Kostopoulos, N., et al. (2024). Ethical considerations in explainable AI systems. Ethics in Technology, 15(2), 77-94.

19. Evaluation of Explainable Artificial Intelligence using TOPSIS Method. (2024). International Journal of Decision Sciences, 13(2), 122-138.

20. Huang, C. (2009). Applications of artificial neural networks in agriculture and biological engineering. Journal of Agricultural Engineering, 35(4), 78-92.

21. Hashimoto, T., et al. (2020). Artificial intelligence applications in anesthesiology. Journal of Clinical Monitoring, 16(8), 249-268.

22. Shahin, M. A., et al. (2001). Artificial neural network applications in geotechnical engineering. Canadian Geotechnical Journal, 38(2), 254-265.

23. Ma, J., Yu, Z., Qu, Y., Xu, J., & Cao, Y. (2020). Application of the XGBoost machine learning method in PM2. 5 prediction: a case study of Shanghai. *Aerosol and Air Quality Research*, *20*(1), 128-138.

24. Zhang, P., Jia, Y., & Shang, Y. (2022). Research and application of XGBoost in imbalanced data. *International Journal of Distributed Sensor Networks*, *18*(6), 15501329221106935.

25. Study on XGBoost's application to network traffic classification in encrypted environments. Based on dataset analysis of a major French ISP. (n.d.). Journal of Network and Computer Applications, 170, 102884. https://doi.org/10.1016/j.jnca.2021.102884

26. Verhelst, T., Caelen, O., Dewitte, J. C., Lebichot, B., & Bontempi, G. (2020). Understanding telecom customer churn with machine learning: from prediction to causal inference. In *Artificial Intelligence and Machine Learning: 31st Benelux AI Conference, BNAIC 2019, and 28th Belgian-Dutch Machine Learning Conference, BENELEARN 2019, Brussels, Belgium, November 6-8, 2019, Revised Selected Papers 28* (pp. 182-200). Springer International Publishing.

27. Faritha Banu, J., Neelakandan, S., Geetha, B. T., Selvalakshmi, V., Umadevi, A., & Martinson, E. O. (2022). Artificial intelligence based customer churn prediction model for business markets. *Computational Intelligence and Neuroscience*, *2022*(1), 1703696.

28. Bhanja, S., & Das, A. (2021). Deep neural network for multivariate time-series forecasting. In *Proceedings of International Conference on Frontiers in Computing and Systems: COMSYS 2020* (pp. 267-277). Springer Singapore.

29. Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., & Kim, S. W. (2019). A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector. *IEEE access*, *7*, 60134-60149.

30. Lu, N., Lin, H., Lu, J., & Zhang, G. (2012). A customer churn prediction model in telecom industry using boosting. *IEEE Transactions on Industrial Informatics*, *10*(2), 1659-1665.

31. Dahiya, K., & Bhatia, S. (2015, September). Customer churn analysis in telecom industry. In *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)* (pp. 1-6). IEEE.

32. Du, S., Lee, J., & Ghaffarizadeh, F. (2019, July). Improve user retention with causal learning. In *The 2019 ACM SIGKDD Workshop on Causal Discovery* (pp. 34-49). PMLR.

33. Duong, T. D. (2023). *Causality for Interpretable Machine Learning* (Doctoral dissertation).