## ABSTRACT

Customer churn significantly impacts the retail industry, causing financial losses. Retaining customers is more cost-effective than acquiring new ones, making churn prediction crucial. This study develops a machine learning model to predict customer churn using historical customer data. Key features include purchase history, interaction frequency, and demographics. We evaluate several algorithms using metrics like accuracy, precision, recall, and ROC-AUC. Our results show effective prediction, offering actionable insights for targeted retention strategies. This research highlights data-driven approaches for enhancing customer retention in retail. Individualized customer retention is tough because most firms have a large number of customers and can't afford to devote much time to each of them. The costs would be too great, outweighing the additional revenue.

However, if a corporation could forecast which customers are likely to leave ahead of time, it could focus customer retention efforts only on these "high risk" clients. The ultimate goal is to expand its coverage area and retrieve more customers loyalty. The core to succeed in this market lies in the customer itself. Customer churn is a critical metric because it is much less expensive to retain existing customers than it is to acquire new customers. To reduce customer churn, retail business companies need to predict which customers are at high risk of churn.

**TABLE OF CONTENTS**

**CHAPTER NO. TITLE PAGE NO.**



**ABSTRACT v**

**LIST OF FIGURES vii**

**LIST OF TABLES viii**

1. **INTRODUCTION 1**
2. **LITERATURE SURVEY 4**
   1. Inferences from the Literature Survey 8
   2. Open Problems in Existing System 10

**3 REQUIREMENT ANALYSIS 12**

* 1. Feasibility Studies / Risk Analysis of the Project 12
  2. Hardware and Software Requirements 14

**4 DESCRIPTION OF PROPOSED SYSTEM 18**

* 1. Selected Methodology 18
  2. Description of Research Approach and Method 21
  3. Architecture 23
  4. Workflow Diagram 27
  5. Financial Report on Estimated Costing 30

1. **RESULTS AND DISCUSSION 34**

5.1 Analysis and Description of Results 34

* 1. Future Enhancements 43

**CONCLUSION 45**

**REFERENCES 46**

**APPENDIX 49**

1. **SOURCE CODE 49**
2. **SCREENSHOTS 64**
3. **RESEARCH PAPER** **70**

# LIST OF FIGURES

**FIGURE PAGE**

**NO. FIGURE NAME NO.**

4.3 Architecture of Machine Learning Pipeline for Churn Prediction 23

4.4 Workflow Diagram of Model Evaluation 27

5.1.1 Comparison between Churners and Non-churners 34

5.1.2 Correlation Plot 35

5.1.3 Checking the balance of dataset 36

5.1.4 Channel wise churners 36

5.1.5 Churners and non-churners percentage 37

5.1.6 Origin wise churners and non-churners percentage 37

5.1.7 Channel wise churn count 37

5.1.8 Confusion matrix of ANN model 38

5.1.9 Confusion matrix of Random Forest model 38

5.1.10 Confusion matrix of XGBoost model 39

5.1.11 Confusion matrix of KNN model 39

5.1.12 Confusion matrix of Decision Tree model 40

5.1.13 Confusion matrix of Support Vector Machine model 40

5.1.14 Confusion matrix of Logistic Regression model 41

**LIST OF TABLES**

**TABLE PAGE**

**NO. TABLE NAME NO.**

4.5.1 Financial report on estimated costing 33

5.1.15 Result of models 41

**CHAPTER 1**

**INTRODUCTION**

Customer churn is a critical challenge in the retail industry, where customers frequently switch between brands based on pricing, product quality, customer experience, and promotional offers. Customer churn refers to the phenomenon where customers stop purchasing from a business, leading to a decline in revenue and profitability. Unlike subscription-based industries such as telecom or streaming services, where churn is explicitly defined when a customer cancels a service, churn in retail is more complex to predict due to the absence of contractual obligations. Customers may become inactive for various reasons, making it essential for businesses to proactively identify and retain at-risk customers.

The retail industry faces churn rates between 15-25% annually, making customer retention a top priority. Studies show that acquiring new customers is significantly more expensive than retaining existing ones, often costing five to six times more. However, personalized customer retention is challenging due to the vast number of customers and the high operational costs of manually reaching out to each individual. A more efficient approach involves predicting customer churn using data-driven techniques, allowing businesses to focus retention efforts only on high-risk customers.

Retail businesses rely on customer loyalty to maintain consistent revenue streams. The more customers they retain, the lower the acquisition costs and the higher the overall profitability. Predicting customer churn enables businesses to implement effective retention strategies such as personalized promotions, improved customer service, loyalty programs, and better engagement strategies. By leveraging data-driven insights, businesses can identify potential churners early and take necessary actions to prevent attrition. A well-designed churn prediction system not only helps preserve market share but also allows companies to grow by enhancing customer satisfaction and long-term engagement.

To address this issue, this study focuses on developing an advanced Customer Churn Prediction Model for the Retail Industry. The goal is to analyze customer behavior patterns, purchasing history, and engagement metrics to identify high-risk churners. Traditional retention methods, which apply the same approach to all customers, are inefficient and costly. Instead, a targeted strategy based on predictive insights ensures that businesses allocate their resources effectively. This research aims to evaluate different predictive techniques and propose an optimized approach to enhance churn prediction accuracy.

By implementing a robust churn prediction model, businesses can reduce revenue losses, improve customer relationships, and strengthen their competitive position in the market. This study will explore key customer behavior indicators, compare different predictive approaches, and propose a strategy that enables businesses to take proactive measures to retain high-value customers before they leave. The ultimate objective is to help retail companies shift from reactive churn management to a proactive, data-driven retention approach that ensures long-term customer loyalty and business growth.

**OBJECTIVES AND SCOPE OF THE PROJECT :**

The primary objective of this project is to develop an effective Customer Churn Prediction Model for the Retail Industry that enables businesses to identify customers at high risk of leaving and implement proactive retention strategies. By analyzing historical customer data, transaction patterns, and behavioral trends, the model aims to provide accurate predictions that help businesses reduce customer attrition, enhance customer loyalty, and optimize marketing efforts. The study seeks to evaluate various predictive techniques, assess their performance, and propose an optimized approach to improve churn prediction accuracy. Additionally, this project focuses on offering actionable insights that businesses can use to design personalized engagement strategies, such as targeted promotions, loyalty programs, and improved customer support, ensuring that at-risk customers remain engaged with the brand.

The scope of this project encompasses data collection, preprocessing, feature selection, model development, evaluation, and deployment. The study considers various customer attributes, including purchase frequency, monetary value, engagement levels, and demographic factors, to build a predictive model that accurately identifies churn-prone customers. The project also explores methods to improve model performance through data optimization and advanced analytical techniques. Furthermore, the scope extends to integrating the churn prediction model into business operations, allowing organizations to make data-driven decisions that enhance customer retention and profitability. By focusing on the retail sector, this project aims to address industry-specific challenges, such as the lack of contractual obligations and the difficulty in identifying silent churners, making it a valuable solution for businesses seeking to maximize customer lifetime value and maintain a competitive edge.

# CHAPTER 2

# LITERATURE SURVEY

The problem of customer churn prediction has been extensively studied in the retail industry, with researchers leveraging machine learning and deep learning techniques to identify patterns and mitigate customer attrition. Various studies have proposed different models and methodologies to improve the accuracy and interpretability of churn predictions.

The study "Churn Detection Using Machine Learning in the Retail Industry" analyzed transactional data from a European retail company to develop predictive churn models. Using a dataset of 105,488 customers, it examined features like purchase history, frequency, engagement, and demographics to identify churn patterns. The study applied multiple machine learning models, comparing their predictive performance. While logistic regression provided baseline accuracy, deep learning techniques, particularly Artificial Neural Networks (ANN) and Multi-Layer Perceptron (MLP), significantly outperformed traditional models. These models effectively captured hidden customer behavior patterns, achieving a precision of 75.60%. The findings highlight deep learning's superior ability to predict churn, allowing businesses to implement proactive retention strategies[1].

The study "Churn Prediction of Customers in a Retail Business Using Exploratory Data Analysis (EDA)" explored customer churn patterns using EDA techniques and tested classification models like Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and XGBoost. Among these, XGBoost outperformed others, demonstrating superior predictive accuracy. The study emphasized data preprocessing and feature selection, including handling missing values, encoding categorical variables, and balancing datasets to improve performance. Recursive Feature Elimination (RFE) and Feature Importance Ranking were used to retain key attributes, aligning with "Predicting Customer Churn in Retailing"​. The study also reinforced the effectiveness of ensemble learning, as Random Forest and XGBoost captured complex customer behavior better than standalone classifiers. Similar findings were observed in "Customer Churn Prediction Based on Stacking Model," where XGBoost, Random Forest, and Logistic Regression improved accuracy​. The study confirms XGBoost’s effectiveness in churn prediction, enabling businesses to implement proactive retention strategies like personalized offers and loyalty programs[2].

The study "Churn Forecasting Using Deep Learning Models" explored Multi-Layered Perceptron (MLP) for improving churn prediction accuracy. It compared various machine learning models and found that deep learning techniques, especially Artificial Neural Networks (ANN), outperformed traditional methods. The study emphasized ANN’s ability to learn hidden patterns in customer data, making it more effective in identifying churn trends. Unlike traditional models, ANN captured complex, non-linear relationships, leading to higher predictive accuracy. This aligns with "Churn Detection Using Machine Learning in the Retail Industry," where deep learning models achieved a precision of 75.60%​. Additionally, "Customer Churn Prediction Based on Stacking Model" demonstrated that combining multiple models, including deep learning, further enhanced performance​. The study reinforces the importance of ANN in churn prediction, helping businesses implement more accurate and proactive retention strategies[3].

The study "Comparative Study of Bank Customers Churn Prediction Using AI/ML" analyzed customer churn in the banking sector while providing insights for retail businesses. It compared Decision Trees, Logistic Regression, and Random Forest, finding that Random Forest performed best, achieving 95.16% accuracy after applying oversampling techniques. The study highlighted the importance of data preprocessing and feature selection in improving model performance. These findings align with research in the retail sector, where ensemble models like Random Forest and XGBoost enhance churn prediction accuracy. The study reinforces the effectiveness of ensemble learning methods for data-driven retention strategies[4].

The study "Customer Churn Prediction Based on Stacking Model" proposed a stacking-based ensemble approach, combining XGBoost, Random Forest, and Logistic Regression for churn prediction. The findings showed that ensemble models significantly improved prediction accuracy and stability compared to individual classifiers. The study emphasized that stacking multiple models helps capture complex customer behavior patterns, making churn prediction more reliable. These results align with other research highlighting the effectiveness of ensemble learning techniques in enhancing customer retention strategies through more accurate and data-driven decision-making[5].

The study "Customer Churn Prediction for a Software-as-a-Service Inventory Management Software Company" focused on churn prediction for SaaS businesses while offering valuable insights applicable to retail churn analysis. It emphasized the importance of feature selection, demonstrating that Random Forest, combined with feature importance analysis, achieved high recall and precision scores. The findings highlight Random Forest's effectiveness in identifying key churn indicators, making it a reliable model for churn detection. This study reinforces the significance of feature engineering and model interpretability in improving customer retention strategies across various industries[6].

The study "Customer Churn Prediction in Telecommunication and Medical Industry Using Machine Learning Classification Models" evaluated the effectiveness of XGBoost, KNN, and Artificial Neural Networks (ANN) in predicting churn. While the research focused on telecommunication and medical industries, its methodology and findings are highly relevant to retail churn prediction. The study demonstrated that ensemble models and deep learning techniques improve churn detection accuracy by capturing complex patterns in customer behavior. These insights reinforce the applicability of advanced machine learning approaches in developing robust and scalable churn prediction models for the retail sector[7].

The study "Predicting Customer Churn in Retailing" analyzed a dataset of approximately 200,000 customers and applied RFM (Recency, Frequency, Monetary) analysis combined with Gradient Boosting models for churn prediction. The findings revealed that customers with fewer past purchases were more likely to churn, with a churn rate of 75% for those with only one previous purchase. The study emphasized the importance of feature selection and correlation analysis in understanding churn behavior. These insights highlight the significance of purchase frequency and customer engagement in predicting churn, reinforcing the effectiveness of machine learning-driven customer retention strategies in retail[8].

The study "Development of Churn Prediction Model using XGBoost – Telecommunication Industry in Sri Lanka" examined the effectiveness of XGBoost for churn prediction in the telecom sector. The results showed that XGBoost outperformed Decision Trees, Logistic Regression, and SVM, achieving an accuracy of 83.13% after hyperparameter tuning. This study highlights XGBoost’s ability to handle complex datasets and improve predictive accuracy, making it a preferred choice for churn prediction across various industries, including retail. The findings reinforce the importance of model optimization and tuning in enhancing churn detection performance[9].

These studies demonstrate that machine learning models, particularly ensemble methods like XGBoost and Random Forest, as well as deep learning techniques such as Artificial Neural Networks (ANN), are highly effective in predicting customer churn. The use of feature selection, data preprocessing, and augmentation techniques plays a crucial role in improving model performance. Additionally, hybrid modeling approaches, such as stacking multiple classifiers, have shown to enhance prediction accuracy and stability. By integrating these advanced techniques, businesses in the retail industry can develop more interpretable and scalable churn prediction models, enabling them to implement data-driven retention strategies and improve customer loyalty[10]

## 2.1 INFERENCES FROM LITERATURE SURVEY

The literature survey provides valuable insights into various approaches and methodologies employed in customer churn prediction within the retail sector. Several key inferences can be drawn from the reviewed research papers:

***Machine Learning-Based Churn Prediction*:**

Many studies highlight the effectiveness of machine learning techniques, including decision trees, support vector machines (SVM), logistic regression, and ensemble methods such as Random Forest, XGBoost, and AdaBoost, in predicting customer churn. These models help businesses identify at-risk customers and develop targeted retention strategies. Additionally, hybrid approaches combining traditional machine learning with deep learning techniques have been explored to improve predictive power. The adaptability of these models across different retail environments makes them suitable for large-scale deployment and real-time decision-making.

***Ensemble and Hybrid Models for Improved Accuracy*:**

Research has shown that hybrid models, such as combining decision trees with neural networks, or using stacking techniques, tend to outperform individual models by leveraging the strengths of multiple algorithms. Studies suggest that ensemble models, such as those using bagging and boosting techniques, provide higher accuracy and better generalization. Additionally, adaptive ensemble techniques that dynamically adjust model weights based on real-time customer data have been proposed for enhanced precision in churn prediction. The integration of reinforcement learning methods into hybrid churn models has also shown potential for continuous learning and model improvement.

***Feature Selection and Importance*:**

Feature engineering plays a crucial role in churn prediction. The inclusion of Recency, Frequency, and Monetary (RFM) features, customer engagement metrics, transaction histories, and demographic data has been shown to improve prediction performance. Some studies use Apriori-based feature selection to identify critical attributes that contribute to churn. Furthermore, automated feature selection techniques using deep learning algorithms have been proposed to handle large-scale retail datasets efficiently. The application of explainable AI (XAI) methods in feature selection is also gaining traction, allowing businesses to better interpret churn predictions and make data-driven decisions.

***Optimization Techniques for Better Performance*:**

The use of optimization algorithms such as Remora Optimization and Bayesian hyperparameter tuning has been explored to enhance the predictive performance of machine learning models. These methods help in fine-tuning model parameters for improved accuracy. Additionally, evolutionary algorithms and metaheuristic optimization methods, such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), have been applied to improve model efficiency. The combination of hyperparameter tuning with feature selection techniques has further improved model robustness and reduced computational overhead.

***Challenges in Churn Prediction for Retail Business:***

Unlike the telecom or banking sectors, where subscription-based models make churn prediction relatively straightforward, retail businesses often deal with non-contractual customers. This makes defining churn more complex, requiring businesses to rely on behavioral indicators rather than explicit contract terminations. Additionally, customer behavior in retail can be influenced by seasonal trends, external economic factors, and competitor strategies, making it harder to maintain consistent churn prediction accuracy. Addressing multi-channel data integration, including in-store and online transactions, remains a challenge in building comprehensive churn models for retail businesses.

***Emergence of Deep Learning in Churn Prediction*:**

While traditional machine learning models are widely used, some studies suggest that deep learning techniques, such as Artificial Neural Networks (ANN) and Multi-Kernel Extreme Learning Machines (MKELM), can further enhance predictive accuracy. However, these methods often require larger datasets and higher computational resources. The application of transformer-based architectures and attention mechanisms in churn prediction has recently gained interest for capturing sequential patterns in customer interactions. Additionally, federated learning approaches are being explored to enable privacy-preserving churn prediction models across multiple retailers while maintaining data security and regulatory compliance.

## 

## 2.2 OPEN PROBLEMS IN EXISTING SYSTEM

Despite advancements in customer churn prediction, several challenges remain unaddressed, limiting the real-world effectiveness of these models in the retail sector.

***Limited Real-Time Implementation:***

Most existing models focus on churn prediction as a static problem, analyzing historical data to predict churners. However, real-time implementation and continuous learning models are not widely explored, which limits businesses' ability to act on churn predictions dynamically. Retail businesses require adaptive models that can process streaming data and update predictions in real-time. Additionally, integrating churn prediction models with customer relationship management (CRM) systems for automated decision-making remains a challenge.

***Handling Imbalanced Datasets:***

Many studies report issues with imbalanced datasets, where churners are often underrepresented. While techniques such as SMOTE (Synthetic Minority Over-sampling Technique) and cost-sensitive learning have been proposed, finding the optimal balance between precision and recall remains a challenge. Additionally, data augmentation techniques and hybrid sampling methods are being explored to improve model generalization. However, ensuring that these techniques do not introduce data bias or overfitting remains an open research problem.

***Lack of Interpretability in Complex Models:***

While ensemble and deep learning models provide higher accuracy, they often act as "black-box" models, making it difficult for businesses to interpret the results and take actionable insights. More research is needed in explainable AI (XAI) to bridge this gap and provide clear justifications for churn predictions. Additionally, techniques like SHAP (Shapely Additive explanations) and LIME (Local Interpretable Model-agnostic Explanations) are being explored, but integrating interpretability without compromising model performance remains a challenge. The lack of transparency in AI-driven decisions can also hinder stakeholder trust and regulatory compliance in business applications.

***Scalability for Large-Scale Retail Businesses:***

Retail businesses often deal with massive customer datasets, requiring scalable machine learning solutions. Many proposed models perform well on small-scale datasets but struggle when applied to large-scale retail operations. Optimizing algorithms for scalability remains an open problem, especially when handling high-dimensional data from multiple sources such as online transactions, in-store purchases, and customer service interactions. Additionally, ensuring that churn prediction models remain computationally efficient while processing millions of customer records is a significant challenge. Cloud-based and distributed computing solutions are being explored, but their cost-effectiveness and implementation feasibility still require further research.

**CHAPTER 3**

**REQUIREMENT ANALYSIS**

* 1. **FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT**

The Customer Churn Prediction in Retail Business Using Machine Learning project involves various feasibility considerations and potential risks that need to be assessed for successful implementation. The feasibility study examines the technical, economic, operational, and legal aspects, while the risk analysis identifies challenges that may arise during model development and deployment.

***Technical Feasibility:***

The project relies on machine learning algorithms such as decision trees, random forests, XGBoost, and neural networks for churn prediction. The feasibility depends on the availability of sufficient customer data, including transaction history, RFM (Recency, Frequency, Monetary) metrics, and demographic details. Cloud-based platforms or high-performance computing resources may be required for model training, especially if deep learning methods are used.

***Economic Feasibility:***

Implementing churn prediction models can lead to long-term cost savings by improving customer retention. However, initial investments in data collection, storage, processing infrastructure, and model development must be considered. Businesses must assess whether the reduction in churn and increased revenue from retained customers will outweigh the implementation costs.

***Operational Feasibility:***

Integrating the churn prediction model into business operations requires seamless collaboration between data scientists, marketing teams, and customer service departments. The feasibility of deploying real-time churn prediction models depends on whether the business has the necessary technical expertise and workflow automation to act on churn alerts effectively.

***Legal and Ethical Feasibility:***

Handling customer data raises privacy and compliance concerns, especially with regulations such as GDPR (General Data Protection Regulation) and CCPA (California Consumer Privacy Act). Businesses must ensure secure data handling, anonymization, and ethical use of customer information to prevent legal issues.

***Data Availability and Quality Risks:***

The success of churn prediction depends on high-quality, comprehensive customer data. Incomplete, biased, or outdated data can lead to inaccurate predictions, reducing the effectiveness of retention strategies.

***Model Performance Risks:***

Machine learning models may suffer from overfitting (performing well on training data but poorly on real-world data) or underfitting (failing to capture meaningful patterns). Choosing the right model, performing hyperparameter tuning, and using techniques like cross-validation can mitigate these risks.

***Scalability and Deployment Risks:***

The model may perform well on sample datasets but struggle when applied to large-scale retail operations. Ensuring scalability requires robust cloud-based or distributed computing solutions.

***Customer Response Risks:***

Even if churn-prone customers are identified, retention strategies may not always be effective. Personalized offers and incentives must be carefully designed based on behavioral insights to maximize engagement.

***Compliance and Security Risks:***

Misuse or mishandling of customer data can lead to regulatory penalties and reputational damage. Strong data security measures, including encryption, access controls, and regular audits, are required to mitigate these risks.

* 1. **HARDWARE AND SOFTWARE REQUIREMENTS**

**Software Requirements:-**

* Operating System:Windows 8 and above
* Languages: Python
* Tools Used: Google colab and Jupyter Notebook

***Python Features for Churn Prediction***

*Easy-to-learn* – Python has a simple syntax with minimal keywords, making it easy to learn and implement for data preprocessing, feature engineering, and model building in churn prediction.

*Easy-to-read*– Python’s structured and indentation-based code improves readability, making it easier to analyze customer transaction data and churn patterns.

*Easy-to-maintain* – Python’s modular design allows seamless maintenance and updates for churn prediction models, ensuring long-term usability.

*Broad standard library* – Python provides extensive libraries like Pandas, NumPy, Scikit-Learn, TensorFlow, and XGBoost, which are essential for data manipulation, machine learning, and deep learning in churn prediction.

*Interactive Mode* – Python supports Jupyter Notebook and interactive environments, enabling real-time testing and debugging of machine learning algorithms for churn analysis.

*Portable* – Python runs across multiple platforms (Windows, Linux, and macOS), making it easy to deploy churn prediction models in different environments.

*Extendable* – Python allows integration with C, C++, and Java for performance optimization in large-scale churn prediction systems.

***Data Analysis and Machine Learning Libraries***

In the Customer Churn Prediction in Retail Business project, various programming languages, machine learning frameworks, data visualization tools, and version control systems are utilized to enhance data analysis, model training, and collaboration.

***Programming Languages***

***Python*** – The primary language for data analysis and machine learning due to its rich ecosystem of libraries, scalability, and ease of use.

***Libraries Used:***

*Pandas & NumPy* – For data manipulation, preprocessing, and numerical computations.

*Scikit-Learn* – For building and evaluating machine learning models like Decision Trees, Random Forest, and XGBoost.

*TensorFlow & PyTorch* – For developing deep learning-based churn prediction models.

*R (Alternative)*– Used for statistical modeling and data visualization with libraries like caret and ggplot2.

***Machine Learning Frameworks***

*Scikit-Learn* – Provides classification, regression, and clustering algorithms for churn prediction.

*TensorFlow & Keras* – Useful for building and training deep learning models such as Artificial Neural Networks (ANN) for predicting churn.

*XGBoost & LightGBM* – Gradient boosting libraries used for high-performance ensemble learning models.

***Data Visualization Tools***

*Matplotlib & Seaborn (Python)* – Used for visualizing customer churn trends, feature importance, and model performance metrics.

*ggplot2 (R)* – Provides high-quality visualizations for data insights and analysis.

***Version Control and Collaboration Tools***

*Git* – Version control system for tracking code changes and managing different iterations of the churn prediction model.

*GitHub, GitLab, or Bitbucket* – Platforms for collaborating with team members, sharing code, and maintaining project repositories.

**Hardware requirements**

To efficiently develop and deploy the Customer Churn Prediction in Retail Business project, adequate computational power, storage, networking, and security infrastructure are required. The hardware setup should support data analysis, model training, and cloud-based deployment to handle large customer datasets and complex machine learning models.

**Development and Testing Environment**

*Computers/Workstations*– Standard workstations or high-performance laptops/desktops with:

*CPU* – multi-core processor (Intel i7/i9, AMD Ryzen 7/9, or Apple M1/M2).

*RAM*– Minimum 16 GB RAM (recommended 32 GB for large datasets).

*GPU (Optional)* – NVIDIA RTX/Quadro GPUs or AMD equivalents for deep learning models.

*Cloud-Based Computing*– For training complex models, Google Colab, AWS EC2, Azure ML, or Google Cloud AI can be used for GPU/TPU acceleration.

***Data Storage***

*Local Storage* – SSD (Solid-State Drive) with at least 512 GB–1 TB for storing datasets, processed files, and model outputs efficiently.

*Cloud Storage* – For large-scale data storage and accessibility, cloud platforms such as:

AWS S3 (Amazon Web Services)

Google Cloud Storage

Azure Blob Storage

***Networking and Connectivity***

*High-Speed Internet* – A fast and stable internet connection (minimum 100 Mbps or higher) is required for:

Cloud-based model training

Large dataset transfers

Collaborative development using version control platforms like GitHub or GitLab.

# CHAPTER 4

# DESCRIPTION OF PROPOSED SYSTEM

The Customer Churn Prediction in Retail Business Using Machine Learning project aims to develop a predictive model that identifies customers likely to stop engaging with a business. The proposed system leverages advanced machine learning techniques to analyze customer behavior patterns, transaction history, and demographic details, enabling businesses to implement proactive retention strategies.

## 

## SELECTED METHODOLOGY OR PROCESS MODEL

In customer churn prediction, various methodologies are employed to identify and predict which customers are likely to leave a service or product. These methodologies range from traditional statistical techniques to advanced machine learning and artificial intelligence approaches. Here is an overview of the key methodologies used:

**Machine Learning Techniques for Customer Churn Prediction**

Several machine learning algorithms are used in customer churn prediction for the retail business, each offering unique advantages based on data complexity, interpretability, and predictive performance.

***Logistic Regression (LR):***

Logistic Regression is a statistical classification algorithm that predicts the probability of customer churn based on historical data. It is widely used due to its simplicity and interpretability in identifying key factors influencing churn. The model works well with binary classification problems, where the output is either "churn" or "no churn". Logistic Regression assigns weights to each feature, helping businesses understand how factors like purchase frequency, transaction value, and customer engagement impact churn. It is particularly effective when data is linearly separable but may struggle with complex, non-linear relationships. Regularization techniques (L1/L2) are often used to improve model performance and prevent overfitting. Despite its limitations, Logistic Regression serves as a strong baseline model for churn prediction in retail businesses.

***Support Vector Machine (SVM):***

Support Vector Machine (SVM) is a supervised learning algorithm that helps classify customers as likely to churn or stay by finding an optimal decision boundary. It works by identifying the hyperplane that best separates churners from non-churners in high-dimensional space. SVM is particularly useful when data is not linearly separable, as it employs kernel functions (e.g., radial basis function, polynomial kernel) to map data into a higher-dimensional space. The algorithm is robust to overfitting, especially in high-dimensional datasets with fewer records. However, SVM can be computationally expensive for large datasets, making it less practical for real-time churn prediction in retail. Despite its complexity, SVM can provide high accuracy for churn detection when tuned correctly.

***Random Forest:***

Random Forest is an ensemble learning algorithm that builds multiple decision trees and combines their outputs to improve accuracy and reduce overfitting. Each tree is trained on a random subset of data and features, ensuring diversity and robustness in churn prediction. This model is highly effective in handling missing values, noisy data, and feature interactions, making it ideal for retail churn analysis. Random Forest provides insights into feature importance, helping businesses understand which factors (e.g., recency, frequency, monetary value, and customer support interactions) contribute to churn. However, as the number of trees increases, training and prediction times can become slower, making it less efficient for real-time applications. Despite this, Random Forest is widely used due to its reliability, high accuracy, and ease of interpretation.

***XGBoost:***

XGBoost (Extreme Gradient Boosting) is an advanced boosting algorithm that sequentially builds decision trees while correcting errors from previous iterations. It is widely regarded as one of the best-performing models for churn prediction due to its high efficiency, scalability, and ability to handle imbalanced data. XGBoost uses gradient boosting techniques, optimizing performance through regularization, feature selection, and hyperparameter tuning. In retail churn prediction, it excels in identifying high-risk customers by analyzing behavioral patterns such as purchase frequency, cart abandonment, and transaction history. However, XGBoost is computationally intensive, requiring GPU acceleration or cloud computing resources for large datasets. Its ability to capture complex feature interactions makes it a preferred choice for businesses looking to improve customer retention strategies.

***Artificial Neural Network (ANN):***

Artificial Neural Networks (ANNs) are deep learning models inspired by the human brain, capable of learning complex, non-linear relationships in customer data. ANNs consist of multiple layers of interconnected neurons, allowing them to detect deep patterns in churn behavior. They work exceptionally well for large retail datasets, where customer interactions involve multiple factors like purchase history, demographics, engagement metrics, and sentiment analysis. However, ANNs require substantial computational power and large amounts of labeled data for training. Techniques like dropout regularization and batch normalization help prevent overfitting. While ANNs offer higher predictive power than traditional models, they lack interpretability, making it difficult for businesses to understand why a customer is likely to churn.

***K-Nearest Neighbor (KNN):***

K-Nearest Neighbor (KNN) is a non-parametric, instance-based learning algorithm that classifies customers based on their similarity to past churners. It calculates the distance between a new customer and existing customers in the dataset, assigning a churn probability based on the majority class among the K-nearest neighbors. KNN is simple to implement and works well with small datasets, but it becomes computationally expensive for large-scale retail applications. The model's accuracy depends on the choice of K (number of neighbors) and the distance metric used (e.g., Euclidean, Manhattan). KNN is highly sensitive to noisy data and requires feature scaling (e.g., standardization or normalization) to perform well. Despite these limitations, KNN can be useful in identifying customer groups prone to churn when combined with clustering techniques.

***Decision Tree:***

Decision Trees are hierarchical models that classify customers based on decision rules derived from historical data. Each node represents a decision point, splitting data based on feature values such as purchase frequency, transaction amount, or engagement level. Decision Trees are easy to interpret, making them useful for understanding why a customer is predicted to churn. However, single decision trees are prone to overfitting, requiring techniques like pruning or ensemble learning (e.g., Random Forest, XGBoost) to improve generalization. Despite their simplicity, Decision Trees remain a strong baseline model for churn prediction in retail due to their interpretability and low computational cost.

**4.2 DESCRIPTION OF RESEARCH APPROACH AND METHODS**

***Data Collection and Pre-processing:***

The first step in customer churn prediction using machine learning is collecting and preprocessing customer data. The dataset typically includes transaction history, customer demographics, engagement metrics, and behavioral data. Raw data often contains missing values, duplicates, and inconsistencies, which require data cleaning and normalization. Preprocessing steps include handling missing values, encoding categorical features, scaling numerical features, and balancing the dataset. The data is then segmented into different time frames (e.g., last 3, 6, or 12 months) to analyze customer behavior trends and improve model performance.

***Feature Extraction:***

Once the data is preprocessed, the next step is feature extraction, where important attributes are selected to enhance the model's predictive power. Feature engineering plays a crucial role in churn prediction by identifying key variables such as Recency, Frequency, and Monetary (RFM) scores, average transaction value, customer support interactions, and discount utilization. Advanced techniques such as Principal Component Analysis (PCA), Recursive Feature Elimination (RFE), and correlation analysis help in selecting the most relevant features while reducing dimensionality. Extracted features serve as input variables for training machine learning models.

***Machine Learning Models:***

Several machine learning algorithms are used for customer churn prediction, each with its own strengths:

Logistic Regression – Provides a baseline model for churn classification using probability scores.

Decision Tree & Random Forest – Helps in understanding feature importance and rule-based classification.

XGBoost – A powerful boosting algorithm that optimizes churn prediction accuracy.

Artificial Neural Networks (ANNs) – Captures complex, non-linear relationships in customer data.

Support Vector Machines (SVM) – Finds optimal decision boundaries for churn classification.  
These models are trained using historical customer data, and their performance is evaluated to select the most effective approach.

***Training and Testing:***

Once the models are developed, they are trained using a labeled dataset of churn and non-churn customers. The dataset is divided into training and testing sets, with 80% used for training and 20% for testing. The models learn from historical customer interactions, transaction patterns, and engagement metrics. Training is done using gradient descent, backpropagation (for deep learning models), and hyperparameter tuning (Grid Search, Random Search, or Bayesian Optimization). Once trained, the models are tested to evaluate how well they generalize to new customer data.

***Evaluation Metrics:***

To assess model performance, several evaluation metrics are used:

Accuracy – Measures the proportion of correctly predicted churners and non-churners.

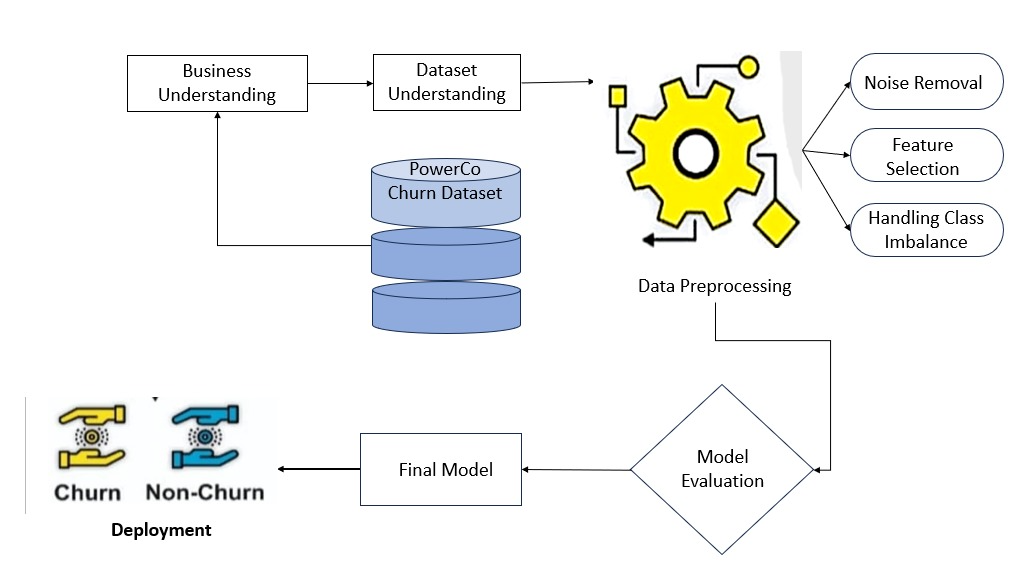
Precision – The ratio of correctly predicted churners to total predicted churners, reducing false positives.

Recall – The ratio of correctly identified churners to actual churners in the dataset, reducing false negatives.

F1-score – The harmonic mean of precision and recall, balancing both metrics.

ROC-AUC Score – Measures the model’s ability to distinguish between churners and non-churners.  
These metrics help in selecting the best model for real-world deployment in retail businesses.

**4.3 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM**



***Fig 4.3: Architecture of Machine Learning Pipeline for Customer Churn Prediction***

The architecture for Customer Churn Prediction in Retail Business consists of multiple stages, from understanding the business problem to deploying a machine learning model that predicts churn and enables proactive customer retention strategies. Each module in the architecture plays a vital role in ensuring the accuracy and effectiveness of the model.

***Business Understanding:***

This stage focuses on defining the business problem, understanding how customer churn impacts revenue, and setting objectives for the predictive model. The goal is to identify customers at risk of churning and develop data-driven retention strategies. Retail businesses face challenges such as non-contractual churn, changing customer preferences, and seasonal purchase behaviors, making churn prediction crucial for profitability. By understanding key performance indicators (KPIs) such as customer lifetime value (CLV), retention rate, and average revenue per user (ARPU), businesses can better shape their predictive model.

***Dataset Understanding:***

The dataset used for churn prediction includes customer transaction records, demographics, product preferences, engagement metrics, and customer service interactions. The PowerCo Churn Dataset is an example of a dataset containing such information. The dataset is explored for missing values, skewed distributions, and inconsistencies before modeling. Exploratory Data Analysis (EDA) is performed to visualize churn trends, identify correlations between features, and detect outliers. Businesses must determine which features influence churn the most, such as purchase frequency, discount usage, customer complaints, and inactivity periods.

***Data Preprocessing:***

Before training the machine learning model, raw data undergoes preprocessing to clean, normalize, and transform the dataset into a structured format. Several key steps are performed:

Feature Selection: Identifying critical factors that influence churn, such as Recency, Frequency, and Monetary (RFM) scores, average transaction value, and product category preferences.

Handling Class Imbalance: Since churners often represent a smaller portion of the dataset, techniques such as SMOTE (Synthetic Minority Over-sampling Technique) and under sampling are applied to balance the data.

Noise Removal: Redundant and irrelevant features are removed to enhance model accuracy. Feature engineering techniques such as one-hot encoding, binning, and standardization are applied to improve data quality.

Factor Identification: Customer behavioral patterns are identified to determine how interactions such as product reviews, email engagement, and promotional offer responses affect churn risk.

***Applying Machine Learning Techniques:***

Once the data is preprocessed, machine learning algorithms are applied to predict churn. Different models are evaluated to determine which method provides the best accuracy and interpretability:

*Logistic Regression*: Provides a baseline churn probability score.

*Decision Trees & Random Forest:* Helps identify the most important customer attributes affecting churn.

*XGBoost*: A high-performance boosting algorithm that excels in handling large retail datasets and complex feature interactions.

*Artificial Neural Networks (ANNs)*: Deep learning models that capture non-linear patterns in customer behavior.

*Support Vector Machines (SVM) & K-Nearest Neighbors (KNN)*: Alternative classification models that segment customers based on behavioral similarities.  
The model selection process depends on accuracy, interpretability, and scalability to ensure the most effective churn prediction system.

***Model Evaluation:***

Once trained, the machine learning models are evaluated using performance metrics to determine their effectiveness. The key evaluation metrics include:

*Accuracy*: Measures overall correctness in predicting churners vs. non-churners.

*Precision*: Determines how many of the predicted churners are actually churners (minimizing false positives).

*Recall*: Measures how well the model identifies actual churners (minimizing false negatives).

*F1-Score*: A balance between precision and recall for better predictive stability.

*ROC-AUC Score*: Assesses the model's ability to differentiate between churners and loyal customers.  
Hyperparameter tuning techniques such as Grid Search, Random Search, and Bayesian Optimization are used to further improve the model’s performance.

***Final Model Selection:***

After evaluation, the best-performing model is selected based on:

High accuracy and recall values to effectively capture churners.

Interpretability for business decision-making, ensuring that the model provides actionable insights.

Scalability and efficiency, making it deployable for large-scale customer data in real-time environments.  
Feature importance analysis is conducted to help businesses understand which factors most contribute to churn, guiding their retention strategies.

***Deployment:***

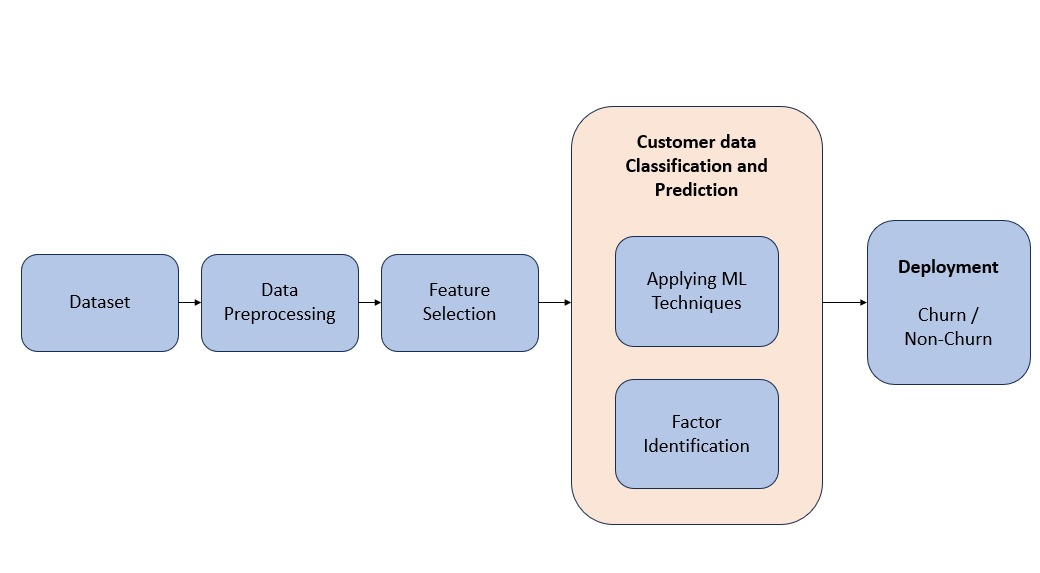
The final model is deployed into a real-time customer management system, where it continuously predicts customer churn based on live data. Businesses use these insights to take immediate action by offering personalized promotions, targeted discounts, or exclusive loyalty benefits to high-risk customers. Deployment options include:

Cloud-based APIs for integrating the model into CRM systems.

Automated dashboards that provide real-time churn insights using visualization tools such as Power BI or Tableau.

Alert-based customer engagement strategies, where the system triggers retention campaigns when a customer is predicted to churn.

**4.4 WORKFLOW DIAGRAM**



***Fig 4.4: Work Flow diagram of Model Evaluation***

***Dataset Collection:***

The success of customer churn prediction heavily relies on comprehensive and high-quality data. Customer data is collected from multiple sources, including transaction logs, CRM (Customer Relationship Management) databases, web/app interactions, customer feedback, and support center records. The dataset typically includes historical purchase behaviors, service usage patterns, subscription details, customer demographics, and engagement history. Gathering a diverse dataset ensures that the model captures various behavioral trends that influence churn. Additionally, data collection must comply with privacy regulations such as GDPR and CCPA, ensuring that customer information is securely handled and anonymized where necessary. To make the dataset robust and valuable, businesses may integrate third-party sources such as market trends, economic indicators, and sentiment analysis from social media to enrich customer insights.

***Data Preprocessing:***

Before using the data for training machine learning models, it must be cleaned and structured properly. Raw datasets often contain missing values, duplicate records, outliers, and inconsistent formats, which can negatively impact the model’s accuracy. The first step is handling missing values using techniques like mean, median imputation, or predictive modeling. Next, categorical variables (e.g., "Male/Female", "Subscription Type") are transformed using one-hot encoding or label encoding. Feature scaling is also applied to normalize numerical values like monthly charges, tenure, and average spending to ensure that all variables contribute proportionally to the model. Additionally, data imbalance is addressed using SMOTE (Synthetic Minority Over-Sampling Technique) or under sampling to prevent models from being biased toward the majority class (non-churn customers). The final preprocessed dataset is structured into training, validation, and test sets to facilitate accurate model evaluation.

***Feature Selection:***

Selecting the right features is crucial for building an efficient and interpretable model. Some variables in the dataset may be redundant, irrelevant, or highly correlated, which can negatively impact the model's performance. Feature selection is performed using statistical tests, correlation heatmaps, and machine learning techniques like Recursive Feature Elimination (RFE), Principal Component Analysis (PCA), and Feature Importance Ranking (using Random Forest or XGBoost). Key features commonly used in churn prediction include:

Tenure: The length of time a customer has been with the company.

Monthly Charges: Higher costs may lead to higher churn rates.

Total Usage: Low engagement with products/services may indicate dissatisfaction.

Customer Support Interactions: Frequent complaints or long resolution times can indicate frustration.

Promotional Engagement: Customers who ignore promotions may have lower retention potential.

By reducing the dataset to the most influential features, we improve model efficiency and interpretability while minimizing overfitting risks.

***Customer Data Classification and Prediction:***

Once the dataset is cleaned and features are selected, machine learning algorithms are applied to classify customers into churn or non-churn categories. Several supervised learning models are considered:

*Logistic Regression (LR):* A simple yet effective model for binary classification, predicting the probability of churn.

*Support Vector Machine (SVM)*: Works well with high-dimensional data and finds an optimal decision boundary.

*Decision Tree & Random Forest*: Tree-based models that capture complex relationships and interactions.

*XGBoost:* A powerful gradient boosting algorithm known for high accuracy and feature importance analysis.

*K-Nearest Neighbors (KNN):* Classifies customers based on similarity with existing churn/non-churn customers.

*Artificial Neural Networks (ANNs):* Deep learning models that capture intricate patterns in customer behavior.

These models are trained using backpropagation and stochastic gradient descent (SGD), with hyperparameter tuning performed using Grid Search or Bayesian Optimization. The models are validated using metrics like accuracy, precision, recall, F1-score, and AUC-ROC to determine the best-performing approach.

***Factor Identification:***

Beyond predicting churn, it is essential to understand the key drivers behind customer attrition. This module focuses on identifying the root causes of churn by analyzing model insights. Feature importance scores from models like Random Forest and XGBoost help determine which variables contribute the most to churn. Some common churn factors include:

High pricing compared to competitors

Poor customer support experiences

Lack of engagement with marketing efforts

Long resolution times for technical issues

Product/service dissatisfaction

By identifying these factors, businesses can implement targeted retention strategies, such as offering discounts to at-risk customers, personalizing product recommendations, or enhancing customer support quality. Predictive analytics dashboards are also created to visualize churn trends and support business decision-making.

***Deployment (Churn / Non-Churn Prediction):***

The final stage involves deploying the trained model into a real-world application where it continuously evaluates customer data and predicts churn probability. The deployment process includes:

Integration with CRM systems: Automating churn alerts for customer service teams.

Cloud deployment (AWS, Google Cloud, or Azure): Ensuring scalable real-time predictions.

API development: Allowing businesses to access churn predictions via REST APIs.

Dashboard visualization: Interactive tools for monitoring churn risk and customer behavior trends.

By implementing real-time churn prediction, businesses can take proactive retention measures, such as offering personalized loyalty programs, exclusive discounts, or better customer engagement strategies. Over time, model performance is monitored and refined using feedback loops to maintain high prediction accuracy.

**4.5 FINANCIAL REPORT ON ESTIMATED COSTING**

Developing a Customer Churn Prediction System involves multiple phases, including data acquisition, preprocessing, feature selection, machine learning model development, deployment, and maintenance. Implementing this system requires advanced computational resources, specialized software, skilled professionals, and ongoing maintenance to ensure accuracy and efficiency. This financial report provides an estimated breakdown of the costs associated with building and deploying the system.

***Hardware and Software Costs***

The development and training of machine learning models for customer churn prediction demand high-performance computing resources and specialized software frameworks. The estimated costs include:

High-Performance Computing Resources: Training models like XGBoost, Decision Trees, or Neural Networks requires substantial computational power. If using on-premise hardware, investing in a high-end GPU server may cost $5,000 - $15,000. Alternatively, cloud-based services like AWS, Google Cloud, or Azure can cost $500 - $3,000 per month, depending on usage.

Software and Development Tools: The system will rely on machine learning frameworks such as Scikit-Learn, TensorFlow, or PyTorch, which are open-source but may require premium cloud-based services, costing $500 - $2,000 per year.

Data Storage & Management: Given that customer data is sensitive, storage solutions like SQL databases, AWS S3, or Google Big Query may cost $200 - $1,000 per month.

Estimated Total Hardware & Software Cost: $5,700 - $20,000

***Data Acquisition and Preprocessing Cost***

A well-structured dataset is crucial for accurate churn prediction. Data preprocessing, cleaning, and feature engineering require both time and resources.

Data Collection: If using an existing customer transaction dataset, the cost may be minimal. However, acquiring external datasets from third-party providers may range between $1,000 - $10,000.

Data Cleaning & Preprocessing: Cleaning data, handling missing values, encoding categorical variables, and scaling numerical features require automated pipelines and manual intervention, costing $2,000 - $7,000.

Feature Engineering & Selection: Selecting the best features (e.g., customer tenure, monthly charges, payment history) for model training may require $1,500 - $5,000.

Estimated Total Data Acquisition & Preprocessing Cost: $4,500 - $22,000

***Machine Learning Model Development Cost***

The core component of the project is building and optimizing machine learning models to classify whether a customer will churn or not.

Algorithm Development & Model Training: Implementing XGBoost, Decision Trees, or Neural Networks requires training on large datasets, which may cost $3,000 - $10,000 in compute resources.

Hyperparameter Tuning & Optimization: To improve model accuracy, multiple iterations of fine-tuning will be necessary, costing $2,000 - $8,000.

Hiring ML Experts & Data Scientists: If outsourcing or hiring experienced professionals, costs may range between $40,000 - $100,000 per year, depending on expertise.

Estimated Total Model Development Cost: $45,000 - $118,000

***Deployment and Maintenance Cost***

After model development, deployment in a real-time business environment requires integration into a customer relationship management (CRM) system, API development, and ongoing model monitoring.

Deployment Infrastructure (Cloud vs. On-Premise): Deploying models on cloud services like AWS Lambda, Google Cloud AI, or Azure ML may cost $1,000 - $8,000 per year.

API Development for Model Integration: A REST API or Flask-based interface will be needed to integrate the model into business systems, costing $2,000 - $10,000.

Real-Time Monitoring & Model Retraining: To ensure the model remains accurate, periodic retraining is required, costing $3,000 - $12,000 per year.

Security & Compliance: Ensuring data privacy regulations (GDPR, CCPA) compliance may require an additional investment of $2,000 - $5,000.

Estimated Total Deployment & Maintenance Cost: $8,000 - $35,000

***Project Management & Operational Cost***

A structured workflow and management approach ensure smooth execution and success of the project.

Project Management & Coordination: Hiring a project manager or Agile team may cost $5,000 - $20,000.

User Training & Documentation: Employees and business analysts need training to interpret and use model insights effectively, costing $3,000 - $8,000.

Customer Support & Bug Fixes: Ongoing support and resolving deployment issues may cost $2,000 - $7,000 per year.

Estimated Total Project Management & Operations Cost: $10,000 - $35,000

**Table 4.5:Financial Report On Estimated Costing**

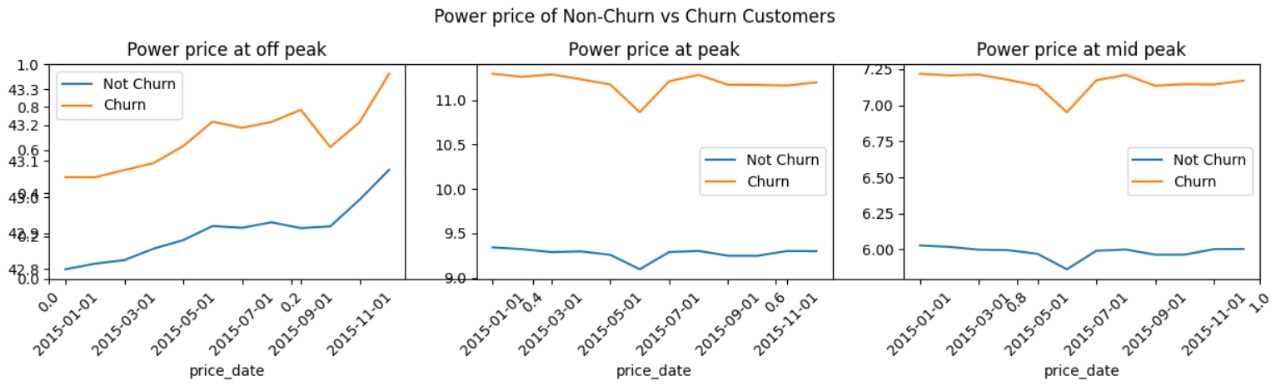
| **Cost Category** | **Estimated Cost (USD)** |
| --- | --- |
| Hardware & Software | $5,700 - $20,000 |
| Data Acquisition & Preprocessing | $4,500 - $22,000 |
| Model Development | $45,000 - $118,000 |
| Deployment & Maintenance | $8,000 - $35,000 |
| Project Management & Operations | $10,000 - $35,000 |
| Total Estimated Cost | $73,200 - $230,000 |

# CHAPTER 5

# RESULTS AND DISCUSSION

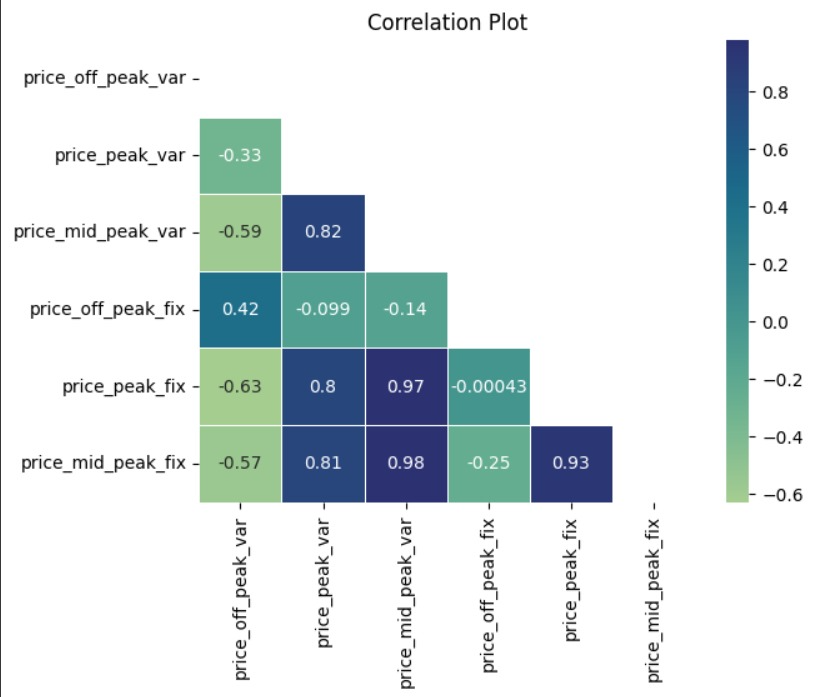
# After implementing the Customer Churn Prediction System, the results were analyzed based on the model's performance, accuracy, and business impact. The discussion focuses on the effectiveness of the prediction model, key insights from data analysis, and the implications for customer retention strategies.

* 1. **ANALYSIS OF RESULTS AND DISCUSSIONS**



### Fig.5.1.1: Comparison Between Churners And Non-Churners

### Churn clients have more high-power prices at off peak, peak and at mid peak. So, we can assume it is one of the factors to churn



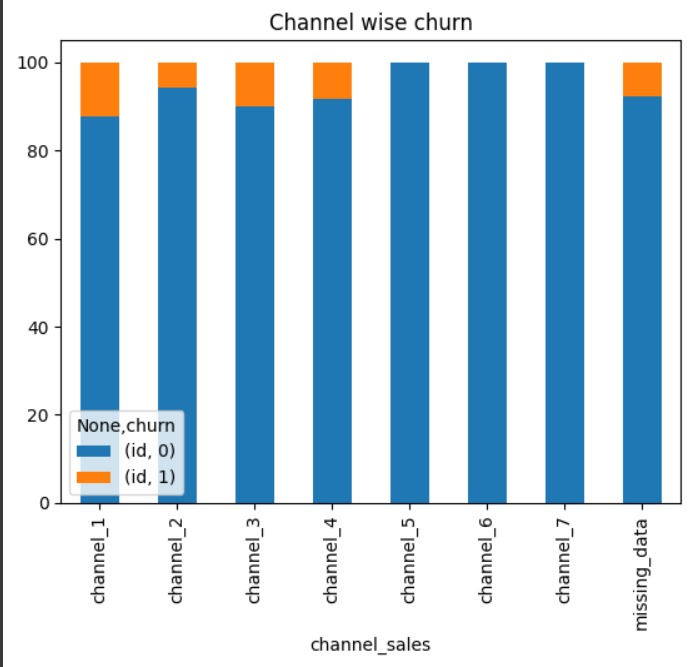
### Fig.5.1.2: Correlation Plot

### A correlation matrix is a table that displays the correlation coefficients between multiple variables in a dataset. Each value in the matrix represents the degree of relationship between two variables, typically ranging from -1 to 1. It helps in understanding how different variables are related to each other, which is useful in feature selection, predictive modeling, and data analysis.

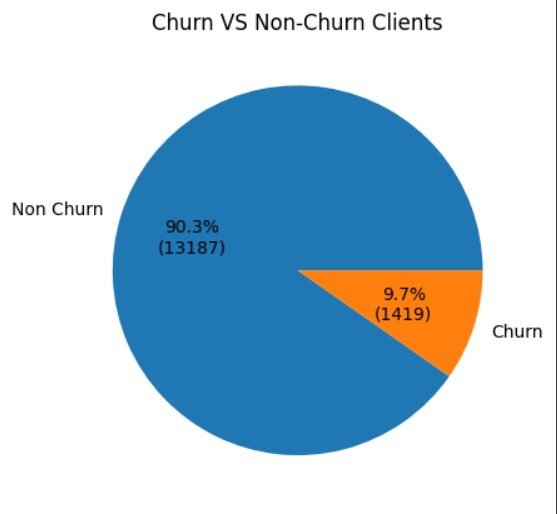
### Pricing schemes. A strong positive correlation (0.93) exists between price\_peak\_fix and price\_mid\_peak\_fix, indicating they move together. Similarly, price\_mid\_peak\_var and price\_peak\_var show a high correlation (0.82), suggesting dependency. A strong correlation (0.97) between price\_peak\_fix and price\_mid\_peak\_var further highlights their connection. Conversely, price\_peak\_fix and price\_off\_peak\_var (-0.63) have a negative correlation, meaning an increase in peak fixed pricing often leads to a decrease in off-peak variable pricing. Price\_mid\_peak\_fix and price\_off\_peak\_var (-0.57) also show an inverse relationship. Price\_off\_peak\_fix has a moderate correlation (0.42) with price\_off\_peak\_var, indicating some dependency. Off-peak prices appear independent of peak and mid-peak rates. These insights help businesses optimize pricing strategies by adjusting peak and mid-peak prices based on demand. Effective pricing models can maximize revenue while maintaining price stability across different time periods.

### 

### Fig.5.1.3: Checking The Balance Of Dataset



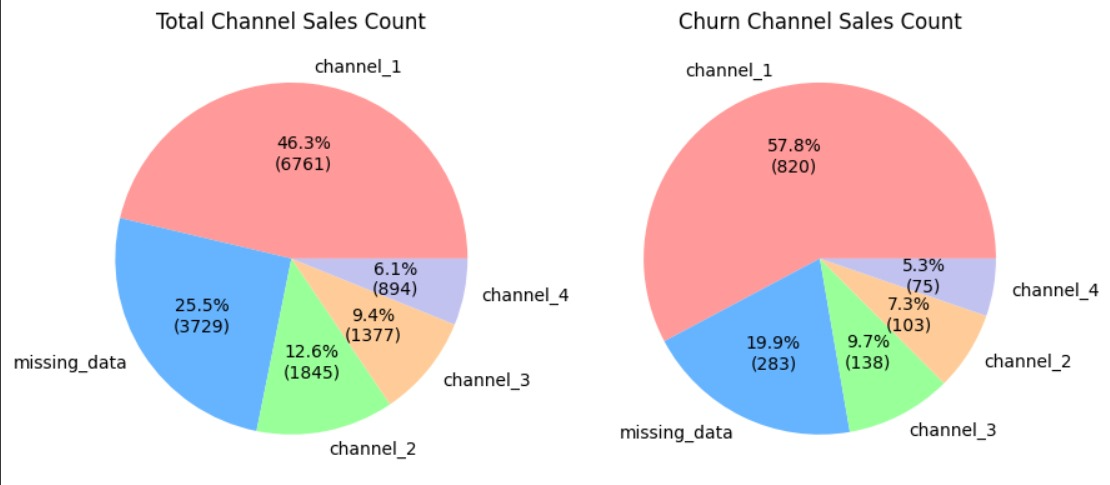
***Fig.5.1.4: Channel Wise Churners***



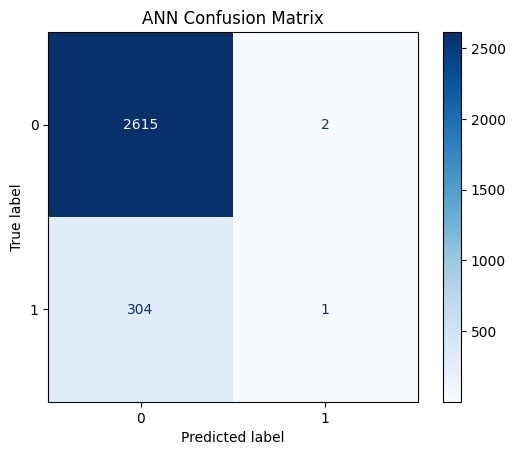
***Fig.5.1.5: Churners and Non-churners Percentage***



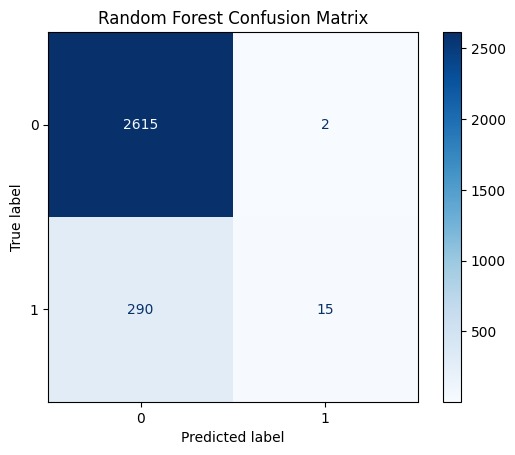
***Fig.5.1.6: Origin wise Churners and Non-churners Percentage***



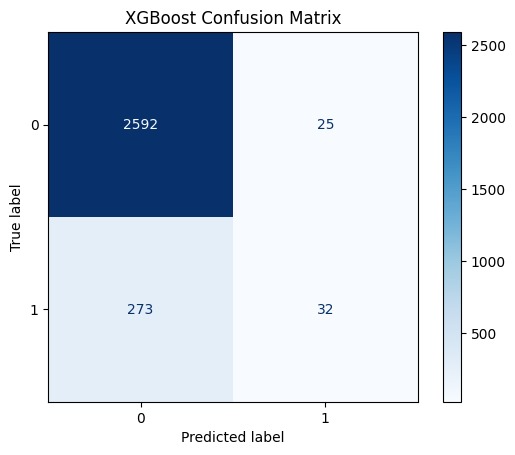
***Fig.5.1.7: Channel Wise Churn Count***



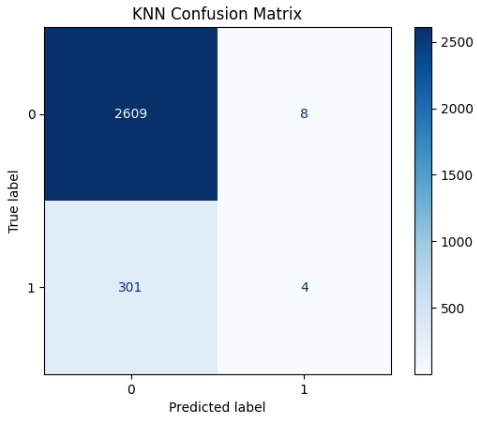
***Fig.5.1.8: Confusion Matrix of ANN Model***



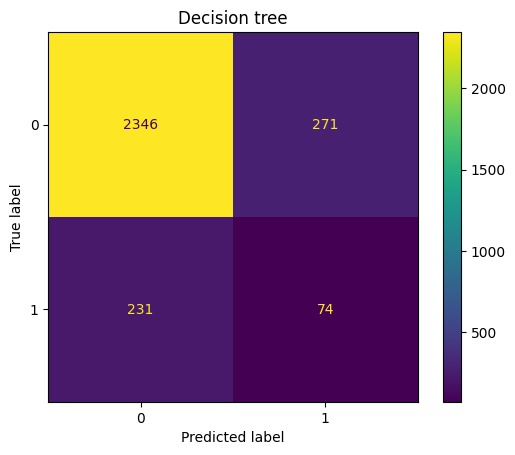
***Fig.5.1.9: Confusion Matrix of Random Forest Model***



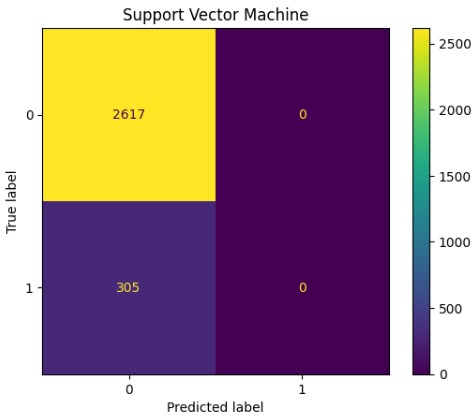
***Fig.5.1.10: Confusion Matrix of XGBoost Model***



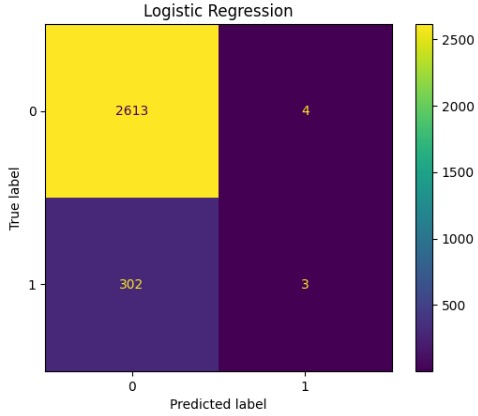
***Fig.5.1.11: Confusion Matrix of KNN Model***



***Fig.5.1.12: Confusion Matrix of Decision Tree Model***



***Fig.5.1.13: Confusion Matrix of Support Vector Machine Model***



***Fig.5.1.14: Confusion Matrix of Logistic Regression Model***

***Table 5.1.15: Results Of Models***

|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| Logistic Regression(LR) | 0.90 |
| Support vector machine | 0.90 |
| Random Forest | 0.90 |
| XG Boost | 0.90 |
| Artificial Neural Network | 0.90 |
| K-Nearest Neighbor | 0.89 |
| Decision Tree | 0.83 |

* 1. **FUTURE ENCHANCEMENTS**

For your Customer Churn Prediction in Retail Business project, future enhancements can focus on improving model accuracy, expanding data sources, and integrating real-time analytics. Here are some key areas for enhancement:

*Advanced Machine Learning & Deep Learning Models* – Implementing deep learning techniques like LSTMs or CNNs can enhance prediction accuracy by capturing complex customer behavior patterns.

*Real-Time Churn Prediction System* – Deploying the model in a cloud-based or real-time system to analyze customer interactions instantly and predict churn before it happens.

*Explainable AI (XAI) for Churn Prediction* – Using SHAP (SHapley Additive explanations) or LIME (Local Interpretable Model-Agnostic Explanations) to make churn predictions more interpretable for business users.

*Integration with Customer Relationship Management (CRM) Systems* – Embedding the churn prediction model within CRM software to enable proactive customer engagement strategies.

*Sentiment Analysis from Customer Reviews & social media* – Incorporating NLP techniques to analyze customer feedback and social media sentiment to enhance churn predictions.

*Dynamic Feature Engineering* – Implementing automated feature selection methods that adapt over time as new data is collected.

*Personalized Retention Strategies* – Using reinforcement learning or A/B testing to design and optimize targeted retention campaigns based on churn predictions.

*Cross-Industry Adaptability* – Expanding the model to different industries such as telecom, banking, or SaaS platforms to generalize churn prediction insights.

*Enhanced Data Privacy & Security Measures* – Ensuring compliance with GDPR, CCPA, or other data protection laws while handling sensitive customer data.

*Automated Model Updating & Performance Monitoring* – Implementing MLOps practices for continuous monitoring and retraining of the model to adapt to changing customer behavior trends.

These enhancements will make the project more robust, scalable, and impactful for real-world business applications.

**CHAPTER 6**

**CONCLUSION**

In this project, we successfully built a customer churn prediction model for the retail business using various machine learning algorithms. The goal was to identify customers who are likely to stop purchasing and provide businesses with insights to take preventive actions. By analyzing customer behavior patterns and historical data, we developed predictive models that help businesses improve customer retention strategies.

We tested multiple machines learning algorithms, including Random Forest, Decision Tree, XGBoost, Support Vector Machine (SVM), Artificial Neural Networks (ANN), Logistic Regression, and K-Nearest Neighbors (KNN). Among these, Random Forest, Decision Tree, SVM, and XGBoost performed the best, accurately identifying potential churners. ANN and SVM provided the highest classification accuracy, around 90%, while other models also performed well but with slightly lower accuracy.

Our findings confirm that machine learning-based churn prediction can significantly help businesses understand churn reasons, optimize marketing strategies, and improve customer experience. By leveraging these insights, companies can offer personalized discounts, special offers, and better services to retain valuable customers.

Overall, this project demonstrates the effectiveness of predictive analytics in reducing customer attrition and highlights the importance of data-driven decision-making in the retail sector. Future enhancements can include real-time prediction, sentiment analysis from customer reviews, and deeper feature engineering to improve accuracy further.

Overall, this project highlights the significant potential of machine learning in customer churn prediction and emphasizes the importance of using advanced analytics for strategic business growth.

# REFERENCES

# [1] A. Bisht and K. Kumar, "CHURN FORECASTING USING DEEP LEARNING MODEL," 2024 International Conference on Intelligent Systems for Cybersecurity (ISCS), Gurugram, India, 2024, pp. 01-05, Doi: 10.1109/ISCS61804.2024.10581266.

# [2] Amornvetchayakul, Phongsatorn and Naragain Phumchusri “Customer churn prediction for software-as-a-service inventory management software company: A case study in Thailand.” 2020 IEEE International Conference on Industrial Engineering and Applications (ICIEA). IEEE, 2020.

# [3] A. Soni, J. Mishra and M. Dixit, "Comparative Study of Bank Customers Churn Prediction using AI/ML," 2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT), Jabalpur, India, 2024, pp. 1359-1365, Doi: 10.1109/CSNT60213.2024.10545969.

[4] B. Sandhiya and R. A. Canessane, "An Extensive Study of Scheduling the Task using Load Balance in Fog Computing," 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2023, pp. 1586- 1593, Doi: 10.1109/ICSCDS56580.2023.10105030.

# [5] D. Azzam, M. Hamed, N. Kasiem, Y. Eid and W. Medhat, "Customer Churn Prediction Using Apriori Algorithm and Ensemble Learning," 2023 5th Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 2023, pp. 377-381, Doi: 10.1109/NILES59815.2023.10296608.

# [6] D. Sweidan, U. Johansson, A. Gidenstam and B. Alenljung, "Predicting Customer Churn in Retailing," 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA), Nassau, Bahamas, 2022, pp. 635-640, Doi: 10.1109/ICMLA55696.2022.00105.

# [7] F. Chen, X. Wei, S. Yu, P. Ma and S. He, "Customer Churn Prediction based on Stacking Model," 2023 4th International Conference on Computer Vision, Image and Deep Learning (CVIDL), Zhuhai, China, 2023, pp. 518-521, Doi: 10.1109/CVIDL58838.2023.10165721.

# [8] H. A. S and M. C, "Evaluative study of cluster-based customer churn prediction against conventional RFM based churn model," 2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), Tiruchirappalli, India, 2023, pp. 1-6, Doi: 10.1109/ICEEICT56924.2023.10156962.

# [9] Hu, Xin, et al. “Research on a customer churn combination prediction model based on decision tree and neural network.” 2020 IEEE 5th International Conference on Cloud Computing and Big DATA Analytics (ICCCBDA). IEEE, 2020.

# [10] I. S. Putra, O. S. Sitompul and E. B. Nababan, "Customer Churn Prediction using Confident Learning and XGBoost," 2024 International Conference on Computer, Control, Informatics and its Applications (IC3INA), Bandung, Indonesia, 2024, pp. 315-318, Doi: 10.1109/IC3INA64086.2024.10732510.

[11] J. S. P, J. D, B. Sandhiya, M. Vanathi and J. Karthika, "Bus Tracking System using Mobile GPS Technology," 2024 International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, 2024, pp. 1811-1813, Doi: 10.1109/ICICT60155.2024.10544750.

# [12] J. Yang, "Design of E-commerce Customer Churn Prediction System Based on Data Mining Techniques," 2023 IEEE 3rd International Conference on Social Sciences and Intelligence Management (SSIM), Taichung, Taiwan, 2023, pp. 114-118, Doi: 10.1109/SSIM59263.2023.10468983.

# [13] S.K. B. Agbemadon, R. Couturier and D. Laiymani, "Churn Detection Using Machine Learning in the Retail Industry," 2022 2nd International Conference on Computer, Control and Robotics (ICCCR), Shanghai, China, 2022, pp. 172-178, Doi: 10.1109/ICCCR54399.2022.9790213.

[14] L. Ou, "Customer Churn Prediction Based on Interpretable Machine Learning Algorithms in Telecom Industry," 2023 International Conference on Computer Simulation and Modeling, Information Security (CSMIS), Buenos Aires, Argentina, 2023, pp. 644-647, Doi: 10.1109/CSMIS60634.2023.00120.

[15] N. Khandelal and V. Sakalle, "Customer Churn Prediction in Telecommunication, Medical Industry Using Machine Learning Classification Models," 2023 6th International Conference on Contemporary Computing and Informatics (IC3I), Gautam Buddha Nagar, India, 2023, pp. 1727-1734, Doi: 10.1109/IC3I59117.2023.10398008.

[16] P. Aggarwal and V. Vijayakumar, "Customer Churn Prediction in the Telecom Sector," 2024 3rd International Conference on Artificial Intelligence for Internet of Things (AIIoT), Vellore, India, 2024, pp. 1-6, Doi: 10.1109/AIIoT58432.2024.10574660.

[17] P. Amornvetchayakul and N. Phumchusri, "Customer Churn Prediction for a Software-as-a-Service Inventory Management Software Company: A Case Study in Thailand," 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA), Bangkok, Thailand, 2020, pp. 514-518, Doi: 10.1109/ICIEA49774.2020.9102099.

[18] P. Pulkundwar, K. Rudani, O. Rane, C. Shah and S. Virnodkar, "A Comparison of Machine Learning Algorithms for Customer Churn Prediction," 2023 6th International Conference on Advances in Science and Technology (ICAST), Mumbai, India, 2023, pp. 437-442, Doi: 10.1109/ICAST59062.2023.10455051.

[19] P. Senthan, R. Rathnayaka, B. Kuhaneswaran and B. Kumara, "Development of Churn Prediction Model using XGBoost - Telecommunication Industry in Sri Lanka," 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 2021, pp. 1-7, Doi: 10.1109/IEMTRONICS52119.2021.9422657.

[20] S. Kumar, P. Baruah, S. Kirubakaran, A. S. Kumar, K. Singh and M. V. J. Reddy, "Remora Optimization with Machine Learning Driven Churn Prediction for Business Improvement," 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2023, pp. 416-420, Doi: 10.1109/IDCIoT56793.2023.10053554.

[21] W. Abbas, M. Usman and U. Qamar, "Churn Prediction of Customers in a Retail Business using Exploratory Data Analysis," 2022 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2022, pp. 130-135, Doi: 10.1109/FIT57066.2022.00033.

[22] X. Hu, Y. Yang, L. Chen and S. Zhu, "Research on a Customer Churn Combination Prediction Model Based on Decision Tree and Neural Network," 2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), Chengdu, China, 2020, pp. 129-132, Doi: 10.1109/ICCCBDA49378.2020.9095611.

# APPENDIX

## SOURCE CODE

**##CUSTOMER CHURN PREDICTION##**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import warnings #ignore unwanted warnings

warnings.simplefilter('ignore')

client\_data = pd.read\_csv('client\_data.csv')

price\_data = pd.read\_csv('price\_data.csv')

client\_data.head(3)

price\_data.head(2)

client\_data.info()

price\_data.info()

#checking duplicates

print(client\_data[client\_data.duplicated()])

print(price\_data[price\_data.duplicated()])

client\_data.describe()

client\_data.shape

print('Number of Unique Clinets: ',price\_data.id.nunique())

price\_data.describe()

price\_data.shape

#Merging Client & Price dataset using Customer ID

client\_churn\_info = client\_data[['id','churn']]

price\_df = client\_churn\_info.merge(price\_data,on='id')

price\_df.head()

#Plotting histogram to see the distribution of the data using the mean values of both clinet and price datasets

price\_df.groupby(['id','price\_date']).mean().hist(figsize=(20,10))

plt.show()

#Since the price date is in object state, convert it into date format

#Changing datatype : price date => object -> datetime64

price\_data = price\_data.astype({'price\_date' : 'datetime64[ns]'})

#Plotting both Energy and Power prices as per the CHURN category by price dated months

churn\_grp\_price = price\_df[price\_df['churn']==1].groupby(['price\_date'])[['price\_off\_peak\_var','price\_peak\_var','price\_mid\_peak\_var','price\_off\_peak\_fix','price\_peak\_fix','price\_mid\_peak\_fix']].mean()

non\_churn\_grp\_price = price\_df[price\_df['churn']==0].groupby('price\_date')[['price\_off\_peak\_var','price\_peak\_var','price\_mid\_peak\_var','price\_off\_peak\_fix','price\_peak\_fix','price\_mid\_peak\_fix']].mean()

#Plotting average price of energy by month

plt.figure(figsize=(15,3))

plt.xticks(rotation=45)

plt.subplot(131)

non\_churn\_grp\_price.price\_off\_peak\_var.plot()

churn\_grp\_price.price\_off\_peak\_var.plot()

plt.xticks(rotation=45)

plt.legend(['Not Churn','Churn'])

plt.title('Power price at off peak')

plt.subplot(132)

non\_churn\_grp\_price.price\_peak\_var.plot()

churn\_grp\_price.price\_peak\_var.plot()

plt.legend(['Not Churn','Churn'])

plt.title('Power price at peak')

plt.xticks(rotation=45)

plt.subplot(133)

non\_churn\_grp\_price.price\_mid\_peak\_var.plot()

churn\_grp\_price.price\_mid\_peak\_var.plot()

plt.legend(['Not Churn','Churn'])

plt.title('Power price at mid peak')

plt.xticks(rotation=45)

plt.suptitle('Power price of Non-Churn vs Churn Customers')

plt.subplots\_adjust(top=0.8)

plt.show()

#Replotting by the average price of power through month

plt.figure(figsize=(15,3))

plt.xticks(rotation=45)

plt.subplot(131)

non\_churn\_grp\_price.price\_off\_peak\_fix.plot()

churn\_grp\_price.price\_off\_peak\_fix.plot()

plt.xticks(rotation=45)

plt.legend(['Not Churn','Churn'])

plt.title('Power price at off peak')

plt.subplot(132)

non\_churn\_grp\_price.price\_peak\_fix.plot()

churn\_grp\_price.price\_peak\_fix.plot()

plt.legend(['Not Churn','Churn'])

plt.title('Power price at peak')

plt.xticks(rotation=45)

plt.subplot(133)

non\_churn\_grp\_price.price\_mid\_peak\_fix.plot()

churn\_grp\_price.price\_mid\_peak\_fix.plot()

plt.legend(['Not Churn','Churn'])

plt.title('Power price at mid peak')

plt.xticks(rotation=45)

plt.suptitle('Power price of Non-Churn vs Churn Customers')

plt.subplots\_adjust(top=0.8)

plt.show()

pd\_corr = price\_data.corr(numeric\_only=True)

mask = np.triu(np.ones\_like(pd\_corr))

sns.heatmap(pd\_corr,annot=True,cmap="crest",linewidth=.5,mask=mask)

plt.title('Correlation Plot')

plt.show()

#Since there are multiple non-necessary price values, let's sort them using the above data of Heatmap

price\_data.drop(['price\_peak\_var','price\_peak\_fix','price\_mid\_peak\_var'],axis=1,inplace=True)

#Filtering out the January and December energy off peak price

price\_off\_peak\_energy = price\_data[['id','price\_off\_peak\_var']]

jan\_prices = price\_off\_peak\_energy.groupby('id').price\_off\_peak\_var.first().reset\_index().rename(columns={'price\_off\_peak\_var':'price\_off\_peak\_var\_jan'})

dec\_prices = price\_off\_peak\_energy.groupby('id').last().price\_off\_peak\_var.reset\_index().rename(columns={'price\_off\_peak\_var':'price\_off\_peak\_var\_dec'})

price\_data.drop('price\_off\_peak\_var',axis=1,inplace=True)

#Taking average of Power off-peak and mid-peak

price\_data = price\_data.groupby('id').mean().reset\_index()

price\_data = price\_data.merge(jan\_prices,on='id').merge(dec\_prices,on='id')

price\_data['energy\_off\_peak\_variation'] = price\_data.price\_off\_peak\_var\_jan - price\_data.price\_off\_peak\_var\_dec

price\_data.drop(['price\_off\_peak\_var\_jan','price\_off\_peak\_var\_dec'],axis=1,inplace=True)

#Final price dataset

price\_data.head()

Principal Component Analysis

from sklearn.decomposition import PCA

from sklearn.preprocessing import StandardScaler

for col in df.select\_dtypes(include=['datetime64']).columns:

df[col] = df[col].astype(np.int64) // 10\*\*9

df = pd.get\_dummies(ds, drop\_first=True)

x, y = df.drop('churn', axis=1), df.churn

pca = PCA(n\_components=2)

pca\_df = pd.DataFrame(pca.fit\_transform(StandardScaler().fit\_transform(x)), columns=['PCA1', 'PCA2'])

pca\_df['churn'] = df['churn']

pca\_df.head()

plt.figure(figsize=(15,5))

plt.subplot(131)

sns.scatterplot(data=pca\_df[pca\_df['churn']==1],x='PCA1',y='PCA2')

plt.title('Chrun Plot')

plt.subplot(132)

sns.scatterplot(data=pca\_df[pca\_df['churn']==0],x='PCA1',y='PCA2')

plt.title('Non Chrun Plot')

plt.subplot(133)

sns.scatterplot(data=pca\_df,x='PCA1',y='PCA2',hue='churn')

plt.title('PCA plot')

plt.show()

from sklearn.model\_selection import train\_test\_split

from sklearn import tree

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay, classification\_report, f1\_score

import matplotlib.pyplot as plt

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size=0.27,stratify=y,random\_state=42)

DECISION TREE CLASSIFIER

dt\_model = tree.DecisionTreeClassifier()

dt\_model.fit(x\_train,y\_train)

y\_train\_pred = dt\_model.predict(x\_train)

y\_pred = dt\_model.predict(x\_test)

cm\_pred2 = confusion\_matrix(y\_test,y\_pred,labels = dt\_model.classes\_)

ConfusionMatrixDisplay(confusion\_matrix=cm\_pred2,display\_labels=dt\_model.classes\_).plot()

plt.show()

print("Model's f1 score for training dataset :",f1\_score(y\_train,y\_train\_pred),

"\nModel's f1 score for test dataset :",f1\_score(y\_test,y\_pred))

print(classification\_report(y\_test,y\_pred))

dt\_model = tree.DecisionTreeClassifier()

dt\_model.fit(x\_train,y\_train)

y\_train\_pred = dt\_model.predict(x\_train)

y\_pred = dt\_model.predict(x\_test)

cm\_pred2 = confusion\_matrix(y\_test,y\_pred,labels = dt\_model.classes\_)

ConfusionMatrixDisplay(confusion\_matrix=cm\_pred2,display\_labels=dt\_model.classes\_).plot()

plt.show()

print("Model's f1 score for training dataset :",f1\_score(y\_train,y\_train\_pred),

"\nModel's f1 score for test dataset :",f1\_score(y\_test,y\_pred))

print(classification\_report(y\_test,y\_pred))

#Feature distribution

x.hist(figsize=(15,15),bins=10)

plt.show()

LOGISTIC REGRESSION

from sklearn.metrics import accuracy\_score, classification\_report, ConfusionMatrixDisplay

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import LabelEncoder, StandardScaler

merged\_data = pd.merge(client\_data, price\_data, on="id", how="inner")

label\_encoder = LabelEncoder()

for column in ['channel\_sales', 'has\_gas', 'origin\_up']:

merged\_data[column] = label\_encoder.fit\_transform(merged\_data[column])

features = merged\_data.drop(columns=['id', 'price\_date', 'churn', 'date\_activ', 'date\_end',

'date\_modif\_prod', 'date\_renewal'])

target = merged\_data['churn']

scaler = StandardScaler()

features\_scaled = scaler.fit\_transform(features)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features\_scaled, target, test\_size=0.2, random\_state=42)

log\_reg = LogisticRegression(max\_iter=1000, random\_state=42)

log\_reg.fit(X\_train, y\_train)

y\_pred = log\_reg.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

report = classification\_report(y\_test, y\_pred)

print(f"Accuracy: {accuracy:.2f}")

print("Classification Report:")

print(report)

ConfusionMatrixDisplay.from\_predictions(y\_test, y\_pred, cmap='viridis')

plt.title("Logistic Regression")

plt.show()

SVM

merged\_data = pd.merge(client\_data, price\_data, on="id", how="inner")

x\_train, x\_test, y\_train, y\_test = train\_test\_split(

features\_scaled, target, test\_size=0.2, random\_state=42

)

print("Shape of x\_train:", x\_train.shape)

print("Shape of y\_train:", y\_train.shape)

from sklearn.svm import SVC #Import SVC from the correct module

svc\_model = SVC(random\_state=0).fit(x\_train, y\_train)

yt\_pred, y\_pred = svc\_model.predict(x\_train), svc\_model.predict(x\_test)

cm\_pred2 = confusion\_matrix(y\_test, y\_pred, labels=svc\_model.classes\_)

ConfusionMatrixDisplay(confusion\_matrix=cm\_pred2, display\_labels=svc\_model.classes\_).plot()

plt.title("Support Vector Machine")

plt.show()

KNN

print("Model's f1 score for training dataset:", f1\_score(y\_train, yt\_pred))

print("Model's f1 score for test dataset:", f1\_score(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

from sklearn.neighbors import KNeighborsClassifier

knn\_model = KNeighborsClassifier(n\_neighbors=11)

knn\_model.fit(X\_train, y\_train)

predicted\_y\_knn = knn\_model.predict(X\_test)

accuracy\_knn = knn\_model.score(X\_test, y\_test)

conf\_matrix\_knn = confusion\_matrix(y\_test, predicted\_y\_knn)

ConfusionMatrixDisplay(confusion\_matrix=conf\_matrix\_knn, display\_labels=knn\_model.classes\_).plot(cmap=plt.cm.Blues)

print("KNN Accuracy:", accuracy\_knn)

print("KNN Classification Report:")

print(classification\_report(y\_test, predicted\_y\_knn))

plt.title("KNN Confusion Matrix")

plt.show()

RANDOM FOREST

from sklearn.ensemble import RandomForestClassifier

rf\_model = RandomForestClassifier(random\_state=0).fit(x\_train, y\_train)

yt\_pred, y\_pred = rf\_model.predict(x\_train), rf\_model.predict(x\_test)

ConfusionMatrixDisplay(confusion\_matrix=confusion\_matrix(y\_test, y\_pred),

display\_labels=rf\_model.classes\_).plot()

plt.title("Random Forest")

plt.show()

print("Model's f1 score for training dataset:", f1\_score(y\_train, yt\_pred))

print("Model's f1 score for test dataset:", f1\_score(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

RANDOM FOREST ROC CURVE

from sklearn.metrics import roc\_curve, roc\_auc\_score

rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)

rf\_model.fit(X\_train, y\_train)

y\_probs = rf\_model.predict\_proba(X\_test)[:, 1]

fpr, tpr, thresholds = roc\_curve(y\_test, y\_probs)

roc\_auc = roc\_auc\_score(y\_test, y\_probs)

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, color='blue', lw=2, label=f'ROC Curve (AUC = {roc\_auc:.2f})')

plt.plot([0, 1], [0, 1], color='gray', linestyle='--')

plt.title('Random Forest ROC Curve', fontsize=16)

plt.xlabel('False Positive Rate', fontsize=14)

plt.ylabel('True Positive Rate', fontsize=14)

plt.legend(loc='lower right')

plt.grid(alpha=0.3)

plt.show()

DECISION TREE

from sklearn import tree

dt\_model = tree.DecisionTreeClassifier().fit(x\_train, y\_train)

y\_train\_pred, y\_pred = dt\_model.predict(x\_train), dt\_model.predict(x\_test)

ConfusionMatrixDisplay(confusion\_matrix=confusion\_matrix(y\_test, y\_pred),

display\_labels=dt\_model.classes\_).plot()

plt.title("Decision tree")

plt.show()

print(f"Model's f1 score for training dataset: {f1\_score(y\_train, y\_train\_pred)}")

print(f"Model's f1 score for test dataset: {f1\_score(y\_test, y\_pred)}")

print(classification\_report(y\_test, y\_pred))

ANN

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Dropout

from tensorflow.keras.optimizers import Adam

from sklearn.metrics import classification\_report, confusion\_matrix, ConfusionMatrixDisplay

model = Sequential([

Dense(32, activation='relu', input\_dim=X\_train.shape[1]),

Dropout(0.2),

Dense(16, activation='relu'),

Dense(1, activation='sigmoid')

])

model.compile(optimizer=Adam(learning\_rate=0.001),

loss='binary\_crossentropy',

metrics=['accuracy'])

history = model.fit(X\_train, y\_train, epochs=20, batch\_size=32, validation\_split=0.2, verbose=1)

y\_pred\_ann = (model.predict(X\_test) > 0.5).astype(int)

accuracy\_ann = model.evaluate(X\_test, y\_test, verbose=0)[1]

print("ANN Accuracy:", accuracy\_ann)

print("ANN Classification Report:")

print(classification\_report(y\_test, y\_pred\_ann))

conf\_matrix\_ann = confusion\_matrix(y\_test, y\_pred\_ann)

ConfusionMatrixDisplay(confusion\_matrix=conf\_matrix\_ann, display\_labels=[0, 1]).plot(cmap=plt.cm.Blues)

plt.title("ANN Confusion Matrix")

plt.show()

XGBOOST CONFUSION MATRIX

import xgboost as xgb

xgb\_model = xgb.XGBClassifier(use\_label\_encoder=False, eval\_metric='logloss')

xgb\_model.fit(X\_train, y\_train)

y\_pred\_xgb = xgb\_model.predict(X\_test)

accuracy\_xgb = xgb\_model.score(X\_test, y\_test)

print("XGBoost Accuracy:", accuracy\_xgb)

print("XGBoost Classification Report:")

print(classification\_report(y\_test, y\_pred\_xgb))

conf\_matrix\_xgb = confusion\_matrix(y\_test, y\_pred\_xgb)

ConfusionMatrixDisplay(confusion\_matrix=conf\_matrix\_xgb, display\_labels=xgb\_model.classes\_).plot(cmap=plt.cm.Blues)

plt.title("XGBoost Confusion Matrix")

plt.show()

ROC CURVE FOR XGBOOST

import xgboost as xgb

from xgboost import XGBClassifier

from sklearn.metrics import roc\_curve, roc\_auc\_score

import matplotlib.pyplot as plt

# Now XGBClassifier is defined

xgb\_model = XGBClassifier(n\_estimators=100, learning\_rate=0.1, max\_depth=3,random\_state=42)

xgb\_model.fit(X\_train, y\_train)

y\_probs = xgb\_model.predict\_proba(X\_test)[:, 1]

fpr, tpr, thresholds = roc\_curve(y\_test, y\_probs)

roc\_auc = roc\_auc\_score(y\_test, y\_probs)

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, color='blue', lw=2, label=f'ROC Curve (AUC = {roc\_auc:.2f})')

plt.plot([0, 1], [0, 1], color='gray', linestyle='--')

plt.title('Receiver Operating Characteristic (ROC) Curve for XGBoost', fontsize=16)

plt.xlabel('False Positive Rate', fontsize=14)

plt.ylabel('True Positive Rate', fontsize=14)

plt.legend(loc='lower right')

plt.grid(alpha=0.3)

plt.show()

COMPARING RANDOM FOREST AND XGBOOST

import time

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix, ConfusionMatrixDisplay

import xgboost as xgb

rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)

start\_time = time.time()

rf\_model.fit(X\_train, y\_train)

rf\_train\_time = time.time() - start\_time

y\_pred\_rf = rf\_model.predict(X\_test)

accuracy\_rf = rf\_model.score(X\_test, y\_test)

print("Random Forest Accuracy:", accuracy\_rf)

print("Random Forest Classification Report:")

print(classification\_report(y\_test, y\_pred\_rf))

conf\_matrix\_rf = confusion\_matrix(y\_test, y\_pred\_rf)

ConfusionMatrixDisplay(confusion\_matrix=conf\_matrix\_rf, display\_labels=rf\_model.classes\_).plot(cmap=plt.cm.Blues)

plt.title("Random Forest Confusion Matrix")

plt.show()

xgb\_model = xgb.XGBClassifier(use\_label\_encoder=False, eval\_metric='logloss')

start\_time = time.time()

xgb\_model.fit(X\_train, y\_train)

xgb\_train\_time = time.time() - start\_time

y\_pred\_xgb = xgb\_model.predict(X\_test)

accuracy\_xgb = xgb\_model.score(X\_test, y\_test)

print("XGBoost Accuracy:", accuracy\_xgb)

print("XGBoost Classification Report:")

print(classification\_report(y\_test, y\_pred\_xgb))

conf\_matrix\_xgb = confusion\_matrix(y\_test, y\_pred\_xgb)

ConfusionMatrixDisplay(confusion\_matrix=conf\_matrix\_xgb, display\_labels=xgb\_model.classes\_).plot(cmap=plt.cm.Blues)

plt.title("XGBoost Confusion Matrix")

plt.show()

print(f"Random Forest Training Time: {rf\_train\_time:.4f} seconds")

print(f"XGBoost Training Time: {xgb\_train\_time:.4f} seconds")

print("\nModel Comparison Summary:")

if accuracy\_rf > accuracy\_xgb:

print("Random Forest performs better in terms of accuracy.")

elif accuracy\_rf < accuracy\_xgb:

print("XGBoost performs better in terms of accuracy.")

else:

print("Both models have the same accuracy.")

if rf\_train\_time < xgb\_train\_time:

print("Random Forest is faster in training.")

else:

print("XGBoost is faster in training.")

RESULT

Random Forest Training Time: 4.5922 seconds

XGBoost Training Time: 0.4476 seconds

Model Comparison Summary:

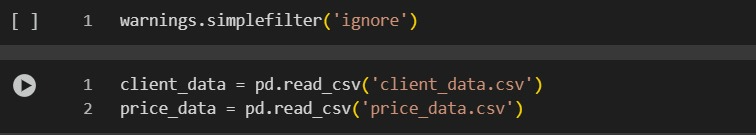
Random Forest performs better in terms of accuracy.

XGBoost is faster in training.

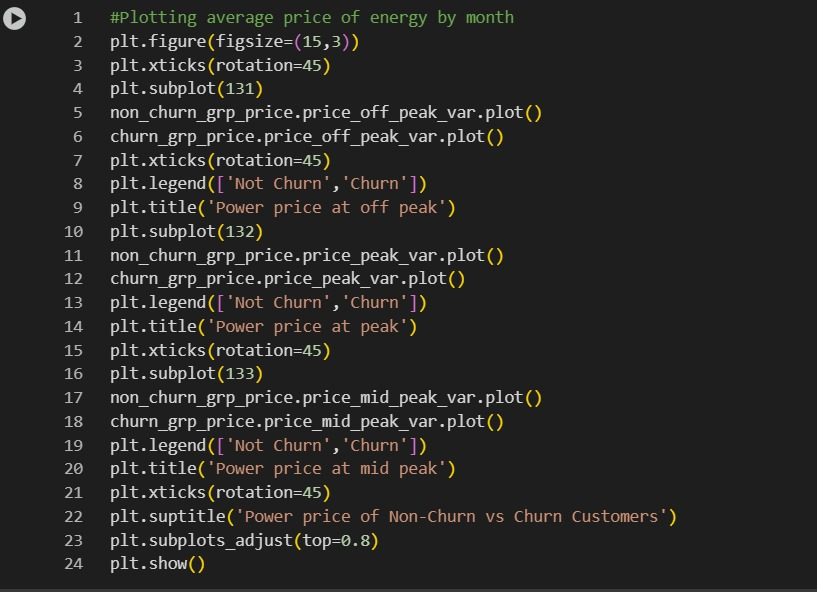
1. **SCREENSHOTS**



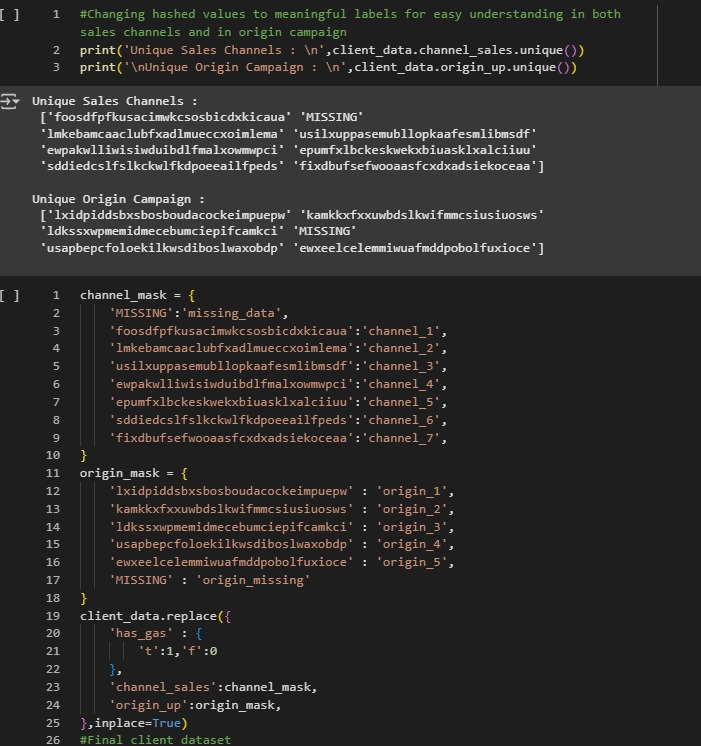
***B.1: Importing Required Libraries***



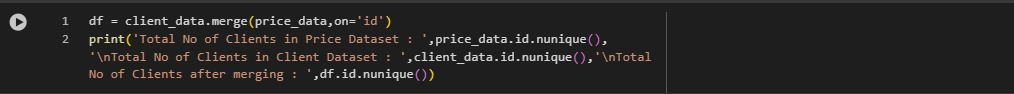
***B.2: Loading Data***



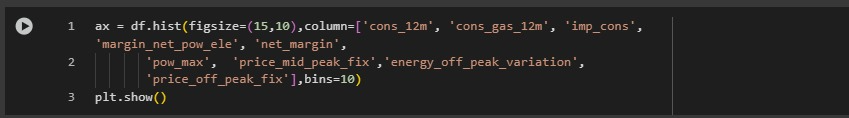
***B.3 Preprocessing the Data***



***B.4:.Extracting Data***



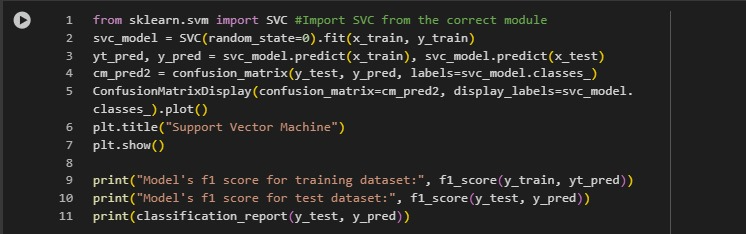
***B.5: Merging Datasets***



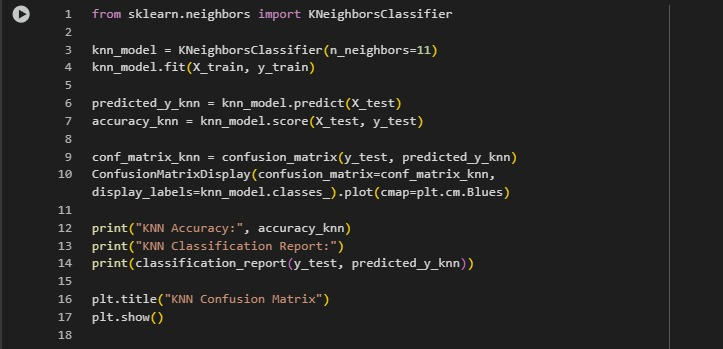
***B.6: Plotting data***



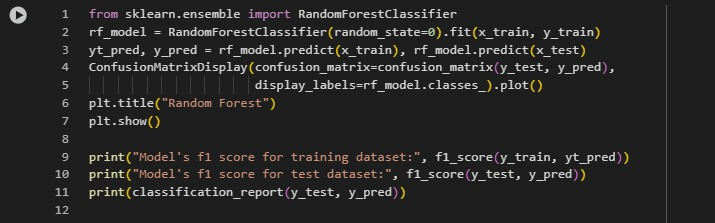
***B.7: Evaluating the Logistic Regression Model***



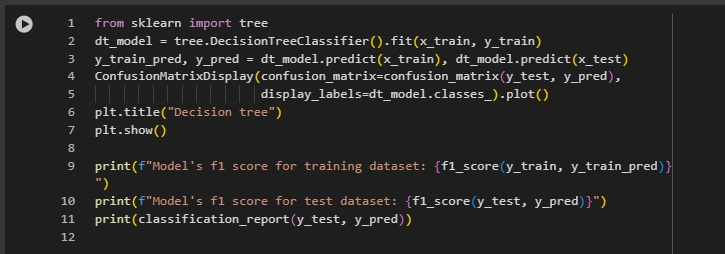
***B.8: Evaluating the Support Vector Machine Model***



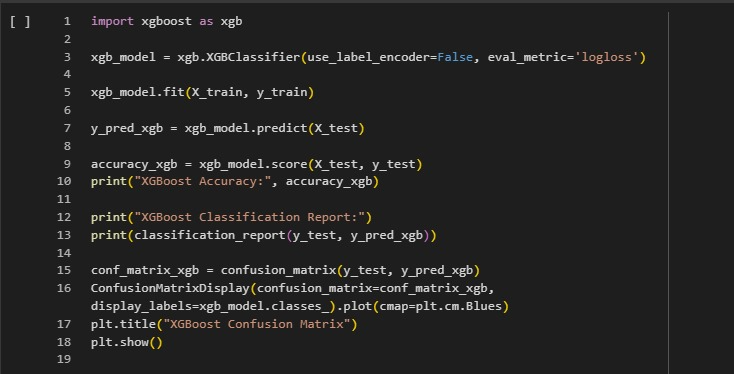
***B.9. Evaluating the KNN Model***



***B.10: Evaluating the Random Forest Model***



***B.11: Evaluating the Decision Tree Model***



***B.12: Evaluating the XGBoost Model***



***B.13: Evaluating the ANN Model***

**Acceptance Intimation**

On behalf of the Technical Program Committee of the **“International Conference on Circuit Power and Computing Technologies”, ICCPCT-2024,** we are pleased to inform you that your paper has been accepted for oral presentation. **Camera ready submission, copyright form submission and view review options were enabled for the selected paper in your CMT author console**. All the papers selected and presented will be published in IEEE Digital Xplore , indexed by Scopus. You can complete the registration process within **a week from the date of acceptance**.The conference will be conducted on **8th & 9th August 2024**. The detailed conference schedule will be send to you later. All review comments and suggestions should be incorporated while preparing camera ready paper. Review comments are available in **CMT account>author console>view reviews**.

**PAYMENT SCREENSHOT**

