CUSTOMER CHURN PREDICTION USING MACHINE LEARNING ALGORITHMS

A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF COMPUTER APPLICATION

By

SHUBHAM TIWARI
(2300290140179)

Under the Supervision of
Mr. Ritesh Kumar Gupta
Assistant Professor
KIET Group of Institutions, Ghaziabad Uttar Pradesh-201206

Submitted to
Department of Computer Applications
KIET GROUP OF INSTITUTIONS, GHAZIABAD
UTTAR PRADESH – 201206

MAY, 2025

DECLARATION

I hereby declare that the work presented in this report entitled "**Customer Churn Prediction using Machine Learning Algorithms**" (**Major-Project-KCA-451**), was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Name: SHUBHAM TIWARI
Roll. No: 2300290140179
Branch: MCA

CERTIFICATE

Certified that **Shubham Tiwari (2300290140179)** has/have carried out the project work having "**Customer Churn Prediction using Machine Learning Algorithms**" (**Major-Project-KCA-451**) for Master of Computer Application from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Mr. Ritesh Kumar Gupta
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Dr. Akash Rajak
Dean
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Customer Churn Prediction using Machine Learning Algorithms

(Shubham Tiwari)

ABSTRACT

Customer attrition is a serious problem for companies, particularly in cutthroat sectors like banking, e-commerce, and telecoms. Businesses may take proactive steps to keep consumers by anticipating which ones are most likely to depart. The goal of this project is to create a customer churn prediction system that leverages machine learning and is connected to a safe and engaging front-end. Churn probability is predicted by analysing past customer data using machine learning techniques like XG Boost, Random Forest, Decision Trees, and Logistic Regression. By identifying important indications like customer behaviour, use trends, and service interactions, the model enables firms to concentrate their retention efforts where they are most required. An integrated chatbot is built into a web-based front-end to improve user engagement and system accessibility. Furthermore, OTP (One-Time Password) verification is used to guarantee safe login and safeguard private client information. A complete approach to client attrition control is offered by the integration of predictive analytics, a sophisticated chatbot, and strong authentication. Through early identification and focused action, it facilitates improved decision-making, raises consumer engagement, and improves overall business success. Hence, this project incorporates a chatbot, OTP verification, and a secure, interactive front-end with machine learning algorithms to forecast client attrition. In order to improve customer engagement, data security, and overall company success, the system assists companies in identifying at-risk clients, offering tailored support, and strengthening retention tactics.

ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, Mr. Ritesh Kumar Gupta for his/ her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Akash Rajak, Dean, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Shubham Tiwari

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Overview

Using machine learning to anticipate customer churn entails determining which consumers are most likely to discontinue using a product or service. ML models like logistic regression, decision trees, random forests, and neural networks may identify trends associated with churn by examining historical data, including customer behavior, transactions, demographics, and interactions. Businesses may use these forecasts to proactively retain clients by making tailored offers or taking other appropriate action. Preprocessing the data, choosing features, training the model, and assessing it using metrics like accuracy, precision, and recall are important processes. Efficient churn prediction aids in lowering revenue loss, enhancing customer happiness, and directing strategic choices for marketing and customer relationship management.

1.2 Objective

Finding clients who are most likely to quit a company or cease utilizing its services in the near future is the main goal of machine learning-based customer churn prediction. By correctly forecasting attrition, businesses can:

- Determine which clients are at danger of leaving.
- Increase client retention by taking early action.
- Prevent churn to minimize revenue loss.
- Use tailored campaigns to maximize marketing efforts.
- Increase client happiness by taking proactive measures to resolve problems.
- Reduce the expense of acquiring new customers by keeping current consumers.
- Divide up your clientele according to their behavior and churn risk.
- Distribute resources to high-risk clients effectively.
- Boost the consumers' lifetime value (LTV).
- Use data-driven insights to assist in strategic decision-making.
- Automate churn detection to keep an eye on things in real time.
- Customize consumer communications according to the probability of churn.
- To improve predictions, track turnover trends over time.
- Evaluate how well retention tactics are working.
- Boost company performance overall with proactive management.

1.3 Applicability

The following are application areas of machine learning-based customer churn prediction, along with a brief description of each:

- **Telecommunications:**
  To determine which users are most likely to move to competitors, telecom firms employ churn prediction. ML models assist lower churn through improved service and targeted offers by examining call patterns, use statistics, and customer support interactions.
- **Financial Services and Banking:**
  Churn models are used by banks to identify clients who could terminate their accounts or cease utilizing their services. In order to initiate retention measures like financial advising or loyalty programs, machine learning (ML) examines transaction behaviour, service usage, and feedback.
- **E-commerce and Retail:**
  Online merchants use product reviews, browsing history, and purchase frequency to forecast client attrition. Personalized product recommendations, timely discounts, and re-engagement emails are all made possible by machine learning to keep customers.
- **Services based on subscriptions**:
  Churn prediction is used by services like Netflix and Spotify to predict cancellations. ML assists with content customization and enhances user engagement by monitoring usage trends, content preferences, and subscription behavior.
- **Software as a Service, or SaaS**:
  Churn prediction is used by SaaS providers to track support interactions, feature usage, and user engagement. Early detection of disengaged users using ML enables the provision of lessons, onboarding assistance, or customized support to keep them on board.

1.4 Scope

Machine learning (ML)-based customer churn prediction has a wide range of applications and influences many different sectors. Businesses are using machine learning (ML) more and more to predict and lower customer attrition as it becomes more cost-effective to retain existing customers rather than acquire new ones. Large volumes of structured and unstructured data, including transaction histories, customer demographics, behavioural patterns, and service encounters, may be processed by ML models to find minute signs of discontent or disengagement. In industries where client loyalty is crucial, such as banking, retail, e-commerce, insurance, telecommunications, and subscription-based services, this predictive capacity is especially useful. Businesses may improve customer experiences, optimize marketing efforts, and apply tailored retention tactics via early detection of at-risk clients. The scope goes beyond simple forecasting. ML is a dynamic tool for strategic planning since it makes it possible to segment churn factors, monitor in real-time, and continuously learn from fresh data. To guarantee prompt and pertinent actions, it facilitates interaction with CRM systems, marketing automation tools, and

customer support platforms. Advanced machine learning techniques like deep learning, ensemble models, and natural language processing are becoming more and more important as data availability and computing capacity rise. This development guarantees that churn prediction will always be a crucial part of data-driven decision-making, assisting businesses in remaining competitive in quickly shifting markets.

1.5 Intended Users

Machine learning-powered customer churn prediction systems are made to benefit a variety of stakeholders in various organizational divisions. Machine learning-based customer churn prediction benefits a number of important stakeholders in a company. In order to re-engage at-risk consumers through promotions and loyalty programs, marketing teams use churn analytics to create tailored offers and targeted retention campaigns. By using behavioral data to identify customers who are likely to leave and proactively address their complaints, customer support teams may improve the quality of their services. Using churn estimates, sales teams may target high-value clients that are at danger and adjust their communication tactics to minimize customer attrition. Churn analytics are used by product managers to comprehend user displeasure, enhance product features, and direct future development. Lastly, it is the duty of data scientists and business analysts to track the effectiveness of churn models, retrain them as necessary, and transform churn data into useful insights that aid in strategic decision-making. When together, these users employ churn prediction to boost growth, increase customer retention, and enhance overall business results.

CHAPTER 2
FEASIBILITY STUDY

A feasibility study using machine learning (ML)-based customer churn prediction assesses the system's operational, financial, and technological viability. The availability of consumer data (such as transactions, use, and demographics) plus sophisticated machine learning methods (such as logistic regression, random forests, and neural networks) that can precisely simulate churn behavior make it technically possible. For operational purposes, companies may incorporate these models into their current analytics or CRM systems to make decisions in real time. The potential return on investment is considerable because of better customer retention and lower acquisition costs, even though the initial setup expenses include data infrastructure, trained staff, and model development. Although data privacy and quality are important factors, dangers may be controlled with the right approach. All things considered, the study demonstrates that machine learning (ML)-based churn prediction is a workable, affordable, and expandable solution that may greatly improve customer retention tactics and corporate performance in a variety of sectors.

2.1 Technical Feasibility

With the availability of data, scalable infrastructure, and contemporary technologies, it is theoretically possible to anticipate customer attrition using machine learning. Accurate predictive models must be developed, implemented, and maintained using the proper hardware, software, and data management system combinations.

2.1.1 Hardware Requirements:

- Processor (CPU/GPU): GPUs (like the NVIDIA RTX or A100) speed up deep learning work, whereas multi-core CPUs (like the Intel i5/i7) are enough.
- Memory (RAM): For huge datasets and speedier computing, at least 4 GB of RAM is required; 8 GB or more is advised.
- Storage: 512 GB is the best range for managing high data volumes; SSDs are recommended for quick data access.
- Operating System: Windows, macOS, or Linux (Ubuntu) can be utilized for machine learning activities.

2.1.2 Software Requirements:

- Hyper Text Markup Language, or HTML:
  The standard language for building web page structures is HTML. It

provides definitions for things like forms, links, paragraphs, and headers.

- Cascading Style Sheets, or CSS:
  The layout, colors, fonts, and responsiveness of online pages are all controlled by CSS. For clearer code, it keeps design and content apart.
- JavaScript:
  JavaScript is a dynamic programming language that gives web pages interaction through form validation, animations, and API calls. The browser is where it operates.
- Flask:
  A lightweight Python web framework called Flask is used to create APIs and web apps. It works well for small to medium-sized tasks and is adaptable and simple to learn.
- Programming Languages:
  Python is the main language for machine learning.
- ML Libraries & Frameworks:
  For creating and training models, Scikit-learn, XGBoost, TensorFlow are frequently utilized.
- Data Tools:
  Pandas and NumPy handle data preparation.
- Visualization Tools:
  To analyze churn patterns and show findings, utilize Tableau, Seaborn, or Matplotlib.

2.1.3. Algorithmic Analysis:

Here is a quick rundown of each algorithm or technique, particularly as it relates to machine learning-based customer churn prediction:

2.1.3.1 Random Forest Entropy

In order to anticipate customer turnover, Random Forest with entropy employs numerous decision trees, choosing the optimal splits according to entropy, which quantifies the impurity of the data. As a result, classification accuracy is increased and uncertainty is decreased. It is resilient for churn prediction in a variety of customer groups because it minimizes overfitting and manages complicated, high-dimensional customer data efficiently by averaging the predictions from many trees.

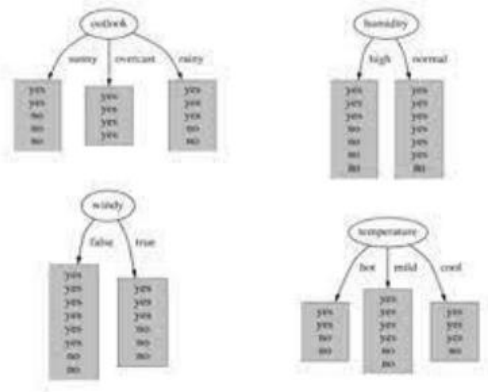

Fig 2.1: Random Forest Entropy

2.1.3.2 Random Forest Information Gain

By choosing splits that maximize information gain—a metric that quantifies the amount of uncertainty a split reduces—Random Forest uses information gain to construct decision trees. This guarantees improved decision-making when forecasting client attrition. Random Forest's ensemble nature improves accuracy and lowers variance, which makes it a very powerful tool for identifying churn-causing trends in consumer behavior.

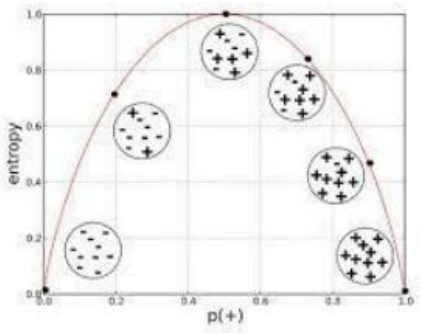

Fig 2.2: Random Forest Information Gain

2.1.3.3 Logistic Regression

By determining a linear link between input variables (such as usage habits and demographics) and churn outcomes, logistic regression estimates the likelihood of customer attrition. It produces a number that ranges from 0 to 1, which is the probability of churn. It is commonly used in churn prediction because of its effectiveness, despite being straightforward and interpretable. It functions best when customer behavior is linearly separable.

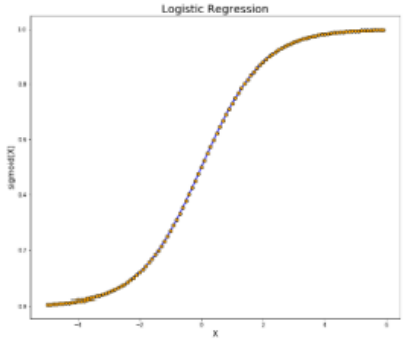

Fig 2.3: Logistic Regression

2.1.3.4 XGBoost

An ensemble of decision trees is constructed successively using the gradient boosting technique XGBoost, with each tree fixing the mistakes of the one before it. Because of its robustness against overfitting, excellent predictive accuracy, and capacity to manage intricate, non-linear connections in huge datasets, it is incredibly successful for predicting customer attrition. By identifying minor patterns in consumer behavior, XGBoost can enhance churn predictions.

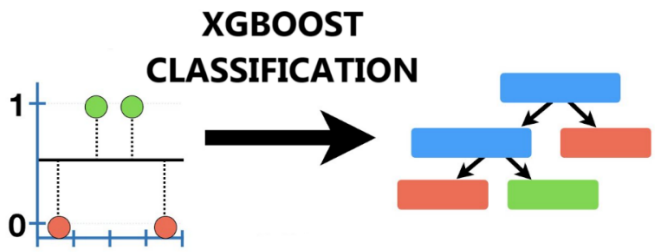

Fig 2.4: XGBoost Classifier

2.1.3.5 Decision Tree

To determine whether a client will leave, decision trees for churn prediction divide customer data according to characteristics (such as age or use). They construct a framework like a flowchart, with branches signifying choices. Despite being simple to understand, they may overfit if improperly pruned. They offer a precise rule of thumb and are especially helpful in figuring out what elements affect churn behavior.

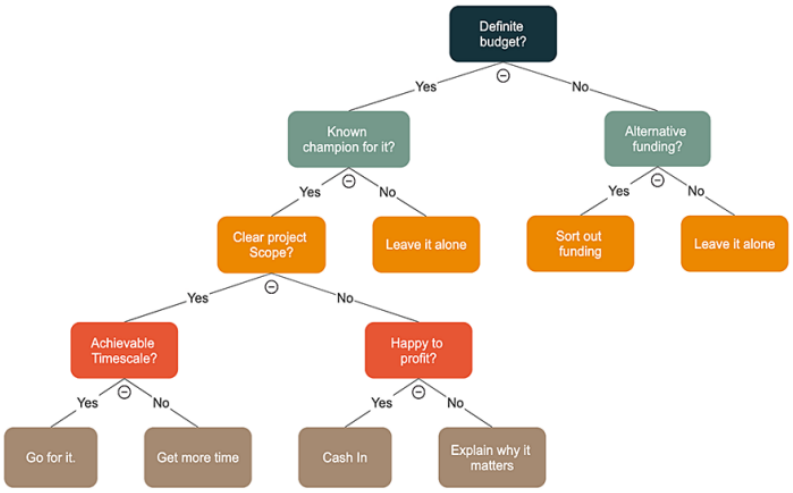

Fig 2.5: Decision Tree

2.1.3.6 SVM

SVM is a potent classifier that determines the best hyperplane to distinguish between consumers who have churned and those who have not. It performs well in high-dimensional domains, particularly feature-rich datasets like consumer behavior. SVM is useful for churn prediction when there is a complicated border between customers who will stay and those who will depart, and it can handle non-linear correlations with the use of kernel functions.

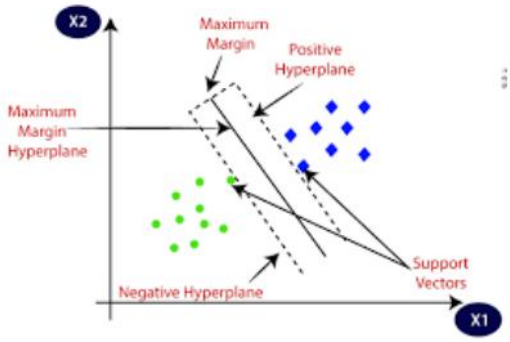

Fig 2.6: SVM

2.1.3.7 Gaussian Naïve Bayes

Gaussian Naïve Bayes predicts turnover based on the likelihood of each class given the data, assuming that customer attributes follow a normal distribution. Particularly for ongoing characteristics like transaction amounts or usage trends, it is quick and effective. When data can be roughly described as Gaussian and customer behavior is independent, it performs well for churn prediction despite its severe assumptions.

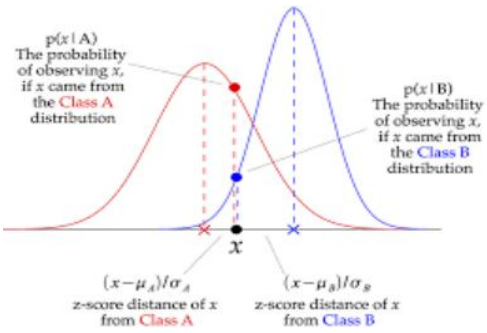

Fig 2.7: Gaussian Naïve Bayes

2.1.3.8 Bernoulli Naïve Bayes

For churn prediction using binary information (such as whether a client made a purchase or utilized a service), Bernoulli Naïve Bayes is employed. Depending on whether these characteristics are present or not, it forecasts the likelihood of churn. This approach works well with sparse data and is especially helpful when examining if certain behaviors (such as signing in or utilizing a product) are associated with customer attrition.

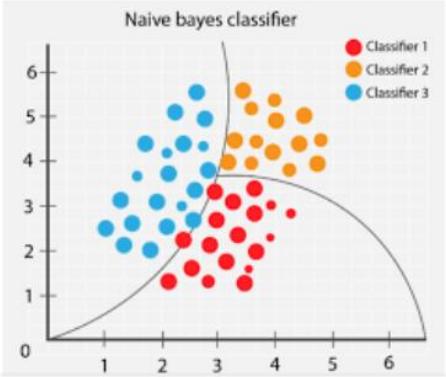

Fig 2.8: Bernoulli Naïve Bayes

2.1.3.9 KNN

KNN uses a customer's k-nearest neighbors in the feature space to classify them according to how similar they are to other customers. By examining the majority class of the closest clients, it forecasts churn. When customer behavior patterns are comparable, KNN performs well for churn prediction; but, because of its high computational cost, it may not be able to handle huge datasets. Personalized churn analysis based on client proximity can benefit from it.

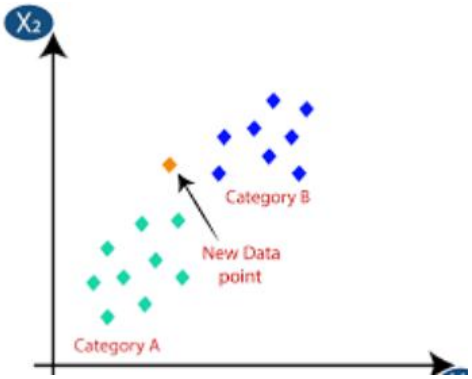

Fig 2.9: KNN

2.1.4 Evaluation Criteria

2.1.4.1 Accuracy

Accuracy = (True Positives + True Negatives) / Total Samples

The percentage of total accurate predictions the model makes is known as accuracy. The number of customers (both churners and non-churners) who were properly identified is displayed in churn prediction.

2.1.4.2 Precision

Precision = True Positives / (True Positives + False Positives)

Precision shows the proportion of consumers who actually churned out of those who were anticipated to do so. By lowering false positives, marketing teams may avoid wasting money on users who are unlikely to depart.

2.1.4.3 Recall

Recall = True Positives / (True Positives + False Negatives)

Recall quantifies the number of real churners that the model accurately recognized. It's essential for reducing false negatives, or missing clients that churned.

2.1.4.4 F1 Score

F1 Score = 2 × (Precision × Recall) / (Precision + Recall)

The F1 Score is calculated by taking the harmonic mean of recall and accuracy. In churn prediction, where the dataset may be unbalanced, it is particularly helpful

because it balances both measures.

2.1.4.5 Support

The amount of real instances of each class (churn and non-churn) in the dataset is referred to as support. By displaying the number of samples in each class, it provides context for other metrics.

2.1.4.6 Confusion Matrix

A table that displays the true positives, true negatives, false positives, and false negatives is called a confusion matrix. A comprehensive overview of the model's performance in both churn and non-churn classification is provided.

2.1.4.7 Weighted Average

Precision, recall, and F1 are calculated using weighted average, which accounts for each class's support (number of occurrences). It works well with churn datasets that have an imbalance between churn and non-churn classes.

2.1.4.8 Macro Average

Without taking into account class imbalance, the macro average determines the average of metrics (precision, recall, and F1) for each class separately. It provides an overall notion of performance by treating churn and non-churn equally.

2.2 Operational Feasibility

The operational viability of machine learning-based customer churn prediction assesses whether the solution can be successfully adopted and maintained inside a company's regular business operations. This is a summary:

- **Integration is simple:**
  ML models may be included into marketing, support, and customer relationship management (CRM) systems that are already in place. This enables companies to easily respond in real-time or almost real-time to churn projections.
- **Process Automation:**
  By enabling automated workflows, such as delivering offers or notifications to consumers who are at danger, churn prediction lowers manual labor and guarantees prompt actions.
- **Scalability:**
  When implemented on cloud platforms with scalable computation and storage, the system can manage expanding customer data without requiring significant infrastructure modifications.

- **User-Friendly GUI**
  GUI's that display insights from machine learning models can help non-technical teams (like marketing or customer support) make better decisions and respond faster.

- **Maintenance and Monitoring**
  Internal teams can keep an eye on model performance and make necessary updates with little training. To keep models accurate over time, there are tools for retraining them using fresh data.

2.3 Economic Feasibility

- **Cost of Implementation:**
  Data infrastructure, machine learning development, software tools, and qualified staff are among the upfront costs. However, a lot of cloud platforms and open-source technologies (including Python, Scikit-learn, and TensorFlow) provide reasonably priced solutions, lowering the initial outlay of funds.
- **Operational Savings:**
  Businesses may cut expenses related to wide, untargeted marketing and customer retention initiatives by automating churn detection and focusing primarily on at-risk clients.
- **Increased Customer Retention**:
  It is far less expensive to keep current clients than to get new ones. Long-term profitability may be improved with even a little decrease in churn, which can result in significant revenue increases.
- **Scalable and Sustainable:**
  ML models are cost-effective over time because, once implemented, they can scale across departments or geographical areas with minimal marginal costs.
- **Competitive Advantage:**
  By enabling companies to take strategic action, early churn detection raises customer lifetime value and gives them a competitive edge that makes the investment worthwhile.

2.4 Time Feasibility

Machine learning-based customer churn prediction is time-feasible, which means it can be created, put into practice, and provide outcomes in a manageable amount of time. There were four main phases to the project:

- **Phase 1: Planning and Requirement Analysis**
  This phase entailed specifying the goals of the project, finalizing the algorithms, and making a design document. The tools and technologies were selected on feasibility and functionality grounds.
- **Phase 2: Frontend and Backend Development**
  The UI/UX was designed in this phase with the use of HTML, CSS, and JavaScript.

Flask integration was completed and algorithm implementations tested for correctness.

- **Phase 3: Integration and Testing**
  Integration of all components and thorough testing were conducted for the identification and elimination of bugs. Feedback from the users was also integrated for improvement of the system.
- **Phase 4: Deployment and Documentation**
  The final product was released for deployment, and a proper report/documentation was developed for the ease of operating the system for the users.
  For most firms, customer churn prediction using machine learning (ML) is a feasible and time-efficient option since it can be accomplished in a matter of months with the correct tools and preparation.

CHAPTER 3
SURVEY OF TECHNOLOGIES

3.1 Problem Statement

In the fiercely competitive corporate world of today, keeping current clients is more economical than finding new ones. But until it's too late, businesses frequently find it difficult to determine which clients are most likely to churn. By examining transactional, demographic, and behavioral data, machine learning (ML)-based customer churn prediction seeks to proactively identify these at-risk clients. Businesses may take prompt action, like providing individualized incentives or enhancing customer assistance, by using machine learning (ML) models to find hidden trends and accurately forecast future churn. By doing this, businesses may maximize retention tactics, lower revenue loss, and improve customer happiness. To guarantee accurate forecasts, the main problem is to collect clean, pertinent data and use the best algorithms. Some points on the same are:

- Finding At-Risk Clients:
  Companies frequently lack effective instruments to identify clients who are prone to leave. By examining trends in complaints, consumption, or inactivity, machine learning algorithms can assist in identifying the most vulnerable individuals.
- Managing Unbalanced Datasets:
  Customer retention is typically higher than customer turnover in churn datasets. Accurate prediction is challenging due to this imbalance, necessitating machine learning approaches like resampling or the use of specialized algorithms.
- Feature Selection and Data Quality:
  For prediction, it is essential to choose the appropriate characteristics (such as tenure, call length, and support interactions). Inaccurate results may result from features that are irrelevant or from poor data quality.
- Assessing and Enhancing Model Accuracy:
  Companies need to make sure that churn forecasts are accurate. Long-term success depends on selecting the appropriate performance criteria (such as accuracy and recall) and regularly retraining the model.

3.2 Literature Review

Table 3.1: Literature Review

S.No	Year	Author	Contribution
1	2024	A. Manzoor et al.	With the Provided Review, evaluating techniques like SVM and Bayesian networks, with SVM

			achieving up to 93% accuracy, offers insights for business practitioners and enhances the ability of making decisions for the betterment.
2	2023	A. Khattak et al.	This study introduces a composite deep learning technique for churn prediction, achieving an average accuracy of 91% and identifying critical behavioral features driving customer churn.
3	2022	V. Agarwal et al.	The research investigation offers a machine learning- based churn prediction tactics which incorporates numerous methods (e.g., SVM and neural networks) to achieve an accuracy of 89%, supporting real-time retention strategies.
4	2020	S. Momin	It assesses customer churn, focusing on feature selection and achieving an overall accuracy of 87% with a random forest model applied to subscription-based data.
5	2019	A. K. Ahmad et al.	Using a big data platform, this study applies machine learning to telecom customer churn prediction, demonstrating superior performance with logistic regression and decision trees, achieving an accuracy of 88%.
6	2019	I. Ullah et al.	This research employs random forest techniques to predict churn in the telecom sector, achieving high accuracy (upto 92%) and identifying key factors influencing customer attrition for better decision making.

CHAPTER 4
SYSTEM DESIGN

4.1 About Dataset

The features in the dataset are as follows:

- **Gender:** The customer's gender may have an impact on their turnover likelihood and service preferences.
- **SeniorCitizen:** Shows whether the client is elderly considering churn behaviour is often driven by demographic characteristics such as age.
- **Partner:** Whether an end user has partner or not that mirrors the dynamics of the residence, which at first might dictate the use of service.
- **Dependents:** Indicates if the customer has dependents which can influence financial priorities and churn decisions.
- **Tenure:** A key metric of customer loyalty and churn risk is the span of time the client has been a subscriber.
- **PhoneService:** Analysing for the customer's phone service correlating with the services they use.
- **MultipleLines:** Reveals if the client has a number of phone line which may correlate with higher engagement or complexity in service needs.
- **InternetService:** One important component of consumer happiness is the particular type of internet service that is being consumed.
- **OnlineSecurity:** If the client uses online security services, it shows that they have incorporated other features.
- **OnlineBackup:** Does the client have access to online backup services reflecting their engagement with value- added services.
- **DeviceProtection:** Shows whether the client has device protection, which could affect how much they think the service is worth.
- **StreamingMovies:** A possible clue to a customer's entertainment needs is whether or not they have a subscription to streaming movie services.
- **Contract:** The customer's contract type is an essential marker of the possibility of churn.
- **PaperlessBilling:** Whether the client likes the billing without paper depicting desire for digital interactions
- **PaymentMethod:** The mode of payment used by the client, which could reveal financial activity.
- **MonthlyCharges:** A notable financial aspect impacting churn decisions is the customer's monthly payments.
- **TotalCharges:** The sum of the fees incurred throughout the client's stay, which represents their complete outlay of funds for the service.

4.2 Methodology

It starts with Data Collection, followed by Exploratory Data Analysis (EDA) and Data Preprocessing. The cleaned data is then split into training and testing sets and used to train various ML models. The best model is selected, and finally, it's used to make predictions on new or unknown data.

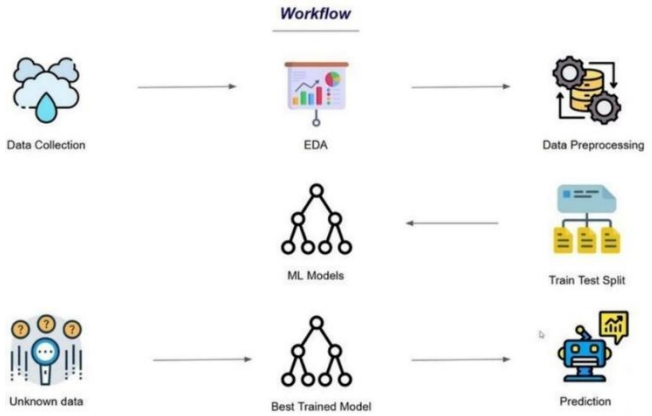

Fig 4.1: Methodology

4.3 System Architecture

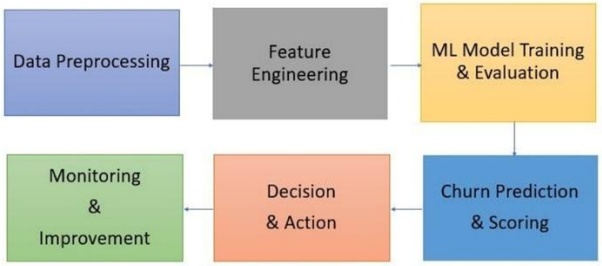

Fig 4.2: System Architecture

4.4 Use Case Diagram

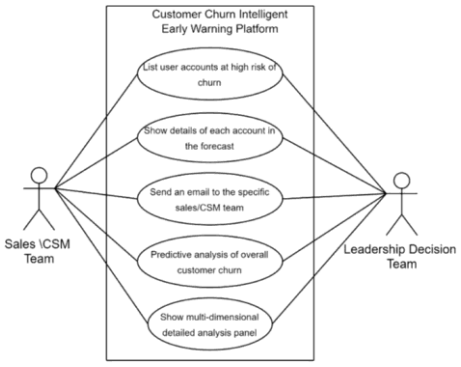

Fig 4.3: Use Case Diagram

4.5 Gantt Chart

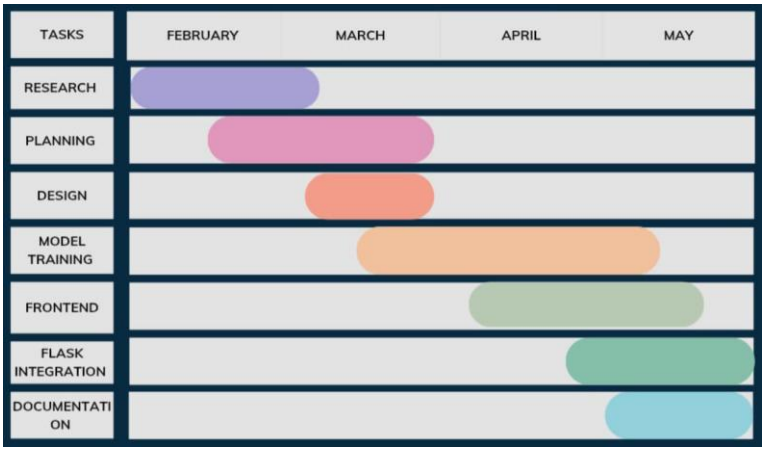

Fig 4.4 Gantt Chart

4.6 Elements in the project

The image below shows the accuracy attained by training the models through different algorithms for customer churn prediction.

```
Decision Tree
Decision Tree cross validation accuracy : 0.78
------------------------------------------------
Random Forest-InformationGain
Random Forest-InformationGain cross validation accuracy : 0.84
------------------------------------------------
Random Forest-Entropy
Random Forest-Entropy cross validation accuracy : 0.84
------------------------------------------------
XGBoost
XGBoost cross validation accuracy : 0.81
------------------------------------------------
LogisticRegression
LogisticRegression cross validation accuracy : 0.79
------------------------------------------------
SVM
SVM cross validation accuracy : 0.64
------------------------------------------------
KNeighbors
KNeighbors cross validation accuracy : 0.78
------------------------------------------------
Bernoulli Naive Bayes
Bernoulli Naive Bayes cross validation accuracy : 0.76
------------------------------------------------
Gaussian Naive Bayes
Gaussian Naive Bayes cross validation accuracy : 0.77
------------------------------------------------
```

Fig 4.5: Different algorithms

- **Attractive GUI**

  A visually appealing ML-based customer churn prediction GUI may significantly improve usability and user engagement. A carousel design makes it simple for users to browse through a variety of functions, including performance measurements, model selection, and data upload. The interface seems lively and engaging because to the smooth wave animations in the background, which give it a contemporary, eye-catching touch. These design components not only enhance appearance but also provide consumers with

straightforward navigation across the whole machine learning process.

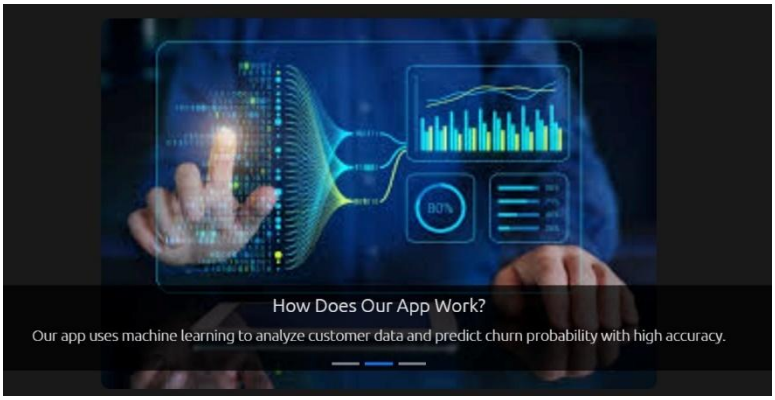

Fig 4.6: Attractive GUI

- **Chatbot**
  Using ML to forecast customer attrition through chatbot integration with pre-input questions improves user engagement and streamlines data collecting. By posing structured queries on customer demographics, service usage, happiness levels, and support interactions—all of which are important determinants of churn—the chatbot may assist users. Even for non-technical users, the chatbot makes the procedure simple and approachable by gathering this data in a conversational manner.

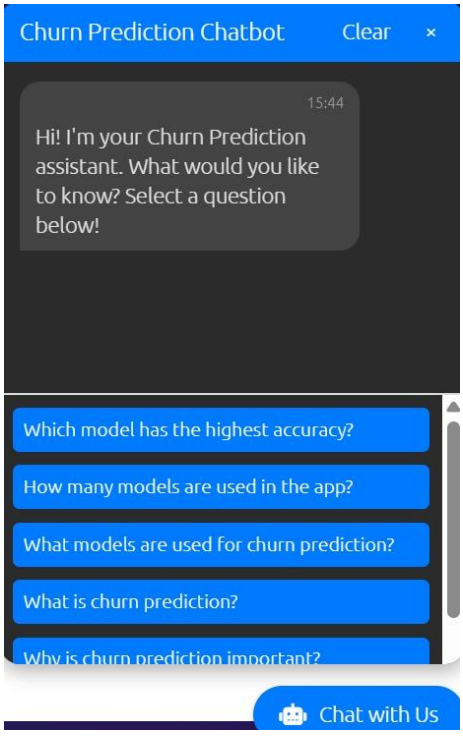

Fig 4.7: Chatbot

- **OTP Generation and Verification**
  Authenticated access is ensured and security is improved when a customer churn prediction machine learning system incorporates OTP (One-Time Password) creation and

verification. An OTP is issued by email for identity verification whenever a user tries to utilize the prediction tool or submit sensitive information. This phase safeguards client data needed for churn analysis and prohibits unwanted access. Users can safely submit data or view prediction results when the OTP has been validated. By using OTP authentication, you may increase trust, protect data privacy, and adhere to security requirements while also improving the entire churn prediction process's dependability and usability in practical applications.

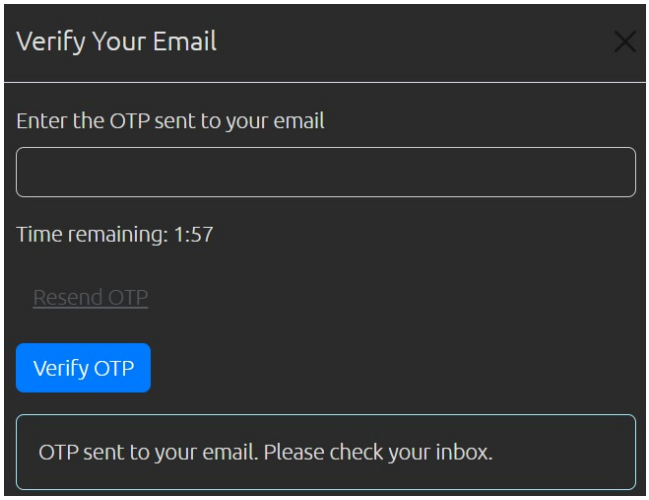

Fig 4.8: OTP Verification

- **Dark and Light Theme**
  A customer churn prediction machine learning platform's user experience is improved by combining both bright and dark themes in the front end, which provide comfort and visual versatility. By allowing users to choose between themes according to their tastes or the lighting conditions, prolonged usage can be less taxing on the eyes. While a dark theme delivers a sleek, contemporary appearance perfect for low light conditions, a light theme offers a clear, bright interface appropriate for daylight or well-lit locations. By increasing accessibility and interaction, this customisation makes the site easier to use. The ML application's professional appeal and inclusiveness for a range of user groups are further enhanced by supporting numerous themes.
- **Feedback on Email**
  Using email feedback for machine learning-based customer churn prediction improves user engagement and aids in system performance improvement. Following a customer's interaction with the churn prediction tool, an automated email may ask for input about the tool's usability, forecast accuracy, and general satisfaction. This insightful input aids in pinpointing places where the user interface and ML model need to be improved. Email-based real-time feedback collection increases user confidence and guarantees that the solution adapts to user demands.

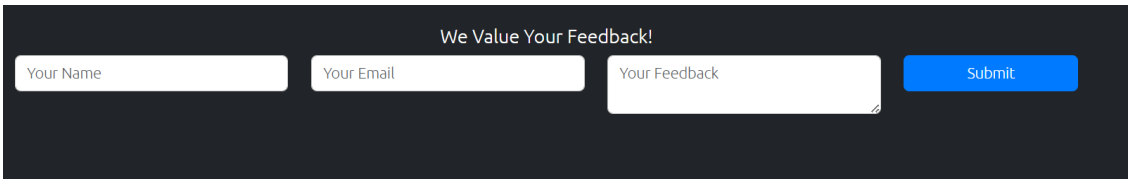

Fig 4.9: Feedback

4.7 Non Functional Requirements

For a machine learning (ML)-based customer churn prediction system to be dependable, scalable, and easy to use, non-functional requirements (NFRs) are essential. Rather than defining what the system does, these requirements specify how it operates. Relevant non-functional needs for such a system are listed below:

- Security:
  When handling sensitive client data, security is essential. Regulations like GDPR and HIPAA must be followed in order to ensure data privacy and guarantee that personally identifiable information is handled appropriately. Strong access control procedures must be in place to limit authorized personnel's access to output results, prediction models, and client data. To secure the end-to-end data lifecycle and avoid unwanted interception or modification, data encryption should also be used both in transit and at rest.
- Performance:
  The effectiveness of the churn prediction system under particular circumstances is referred to as performance. To satisfy user expectations, it should reply in less than two seconds and provide real-time forecasts with minimal latency. The system should be able to process a lot of predictions per second for batch inference or large-scale activities. Furthermore, to guarantee commercial value, the model's prediction performance should continuously be strong, keeping measures like accuracy, precision, recall, or an area under the curve (AUC) above a predetermined threshold (e.g., AUC > 0.85).
- Scalability:
  As user demand and data volume increase, the system should be built to scale easily. This includes the capacity to manage millions of client records without seeing a decline in dependability or performance. To effectively handle workload increases, horizontal scaling—adding more instances of APIs or data processing services—should be provided. This guarantees that the system will continue to function well even when the company grows or the number of contacts with customers rises.
- Maintainability:
  The system's capacity to be readily upgraded and enhanced over time is guaranteed by maintainability. This features a modular code structure that enables independent modification of various components (e.g., feature extraction, model training, API services). Models, datasets, and code should all be subject to version control in order to monitor changes and enable rollbacks as needed. Furthermore, teamwork and long-term viability depend on comprehensive documentation of the machine learning pipeline and lifecycle, from data intake to model deployment.

- Usability:
  Making the system user-friendly and accessible for business users who engage with the churn forecasts is the main goal of usability. In order for non-technical people to make data-driven decisions, the results should be presented in an easy-to-use dashboard or interface. To foster confidence in the model and encourage strategic action, the system should also offer interpretable insights.

CHAPTER 5
IMPLEMENTATION AND CODING

5.1 Implementation Approach

Data gathering is the first of numerous methodical processes involved in implementing machine learning-based customer churn prediction. Companies collect past customer information, such as demographics, transaction history, use trends, support correspondence, and comments. To deal with missing values, encode category variables, and normalize features, this data is cleaned and preprocessed. Next, statistical approaches or feature importance methods are used to choose significant features that influence churn. To assess model performance, the dataset is then divided into training and testing sets. The data is used to train a variety of machine learning methods, including Support Vector Machine (SVM), Random Forest, XGBoost, and Logistic Regression. Prediction quality is evaluated using model assessment measures like as accuracy, precision, recall, F1-score, and ROC-AUC, particularly in light of the class imbalance that is frequently found in churn datasets. Following the selection of the top-performing model, it is implemented for batch or real-time predictions using frameworks such as Flask or FastAPI. Lastly, to ensure accuracy, the model is frequently retrained using fresh data. Through data-driven decision-making, this solution assists companies in proactively managing customer attrition, increasing client retention, and boosting profitability.

5.2 Coding

Importing the dependencies

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

from sklearn.preprocessing import LabelEncoder
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBRFClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
```

```
from sklearn.naive_bayes import BernoulliNB
from sklearn.metrics import accuracy_score , confusion_matrix , classification_report
import pickle
#Data Loading and Understanding
# Import pandas
import pandas as pd
# Data Loading and Understanding
df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv', encoding='utf-8-sig')
#showing the description of columns
df.info()
```

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

Fig 5.1: Data Column Description

```
sns.pairplot(df)
```

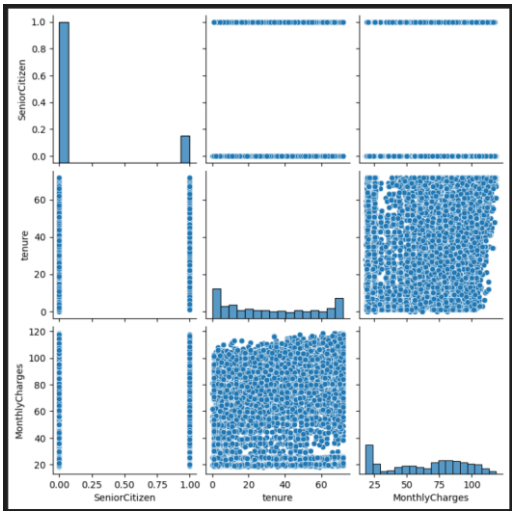

Fig 5.2: Pair-plot Graph

```
#dropping the column as it is not required
df = df.drop(columns=["customerID"])
df.head(2)
df["TotalCharges"] = df["TotalCharges"].replace({" ": "0.0"})
df["TotalCharges"] = df["TotalCharges"].astype(float)
df.describe()
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692	2279.734304
std	0.368612	24.559481	30.090047	2266.794470
min	0.000000	0.000000	18.250000	0.000000
25%	0.000000	9.000000	35.500000	398.550000
50%	0.000000	29.000000	70.350000	1394.550000
75%	0.000000	55.000000	89.850000	3786.600000
max	1.000000	72.000000	118.750000	8684.800000

Fig 5.3: Statistical Measures of data

```
#correlation matrix - heatmap
plt.figure(figsize=(8,4))
sns.heatmap(df[["tenure" , "MonthlyCharges" , "TotalCharges"]].corr() , annot=True)
plt.title("Correlation Matrix - Heatmap")
plt.show()
```

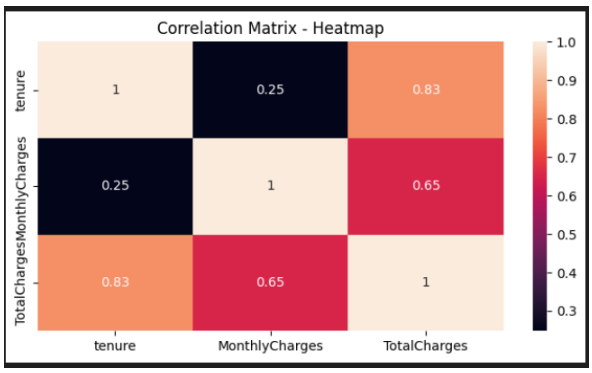

Fig 5.4: Heatmap

```
#data preprocessing
#label encoding of target column
#replace yes with 1 and no with 0 in churn column
df["Churn"] = df["Churn"].replace({"Yes": 1 , "No": 0})
df.head(3)
object_columns = df.select_dtypes(include="object").columns
print(object_columns)
from sklearn.preprocessing import LabelEncoder
import pickle
# Ensure object_columns is defined (if not already defined earlier)
object_columns = df.select_dtypes(include=['object']).columns
```

```
# Initialize a dictionary to store label encoders
encoders = {}
# Apply LabelEncoder to each categorical column
for column in object_columns:
    label_encoder = LabelEncoder()
    df[column] = label_encoder.fit_transform(df[column])
    encoders[column] = label_encoder
# Save the encoders to model/encoders.pkl (outside the loop)
with open("model/encoders.pkl", "wb") as f:
    pickle.dump(encoders, f)
#splitting the features and target
x = df.drop(columns=["Churn"])
y = df["Churn"]
#splitting the train and test set
x_train , x_test , y_train , y_test = train_test_split(x , y , test_size=0.2,random_state=42)
#smote - synthetic minority oversampling technique (only on training data)
smote = SMOTE(random_state=42)
x_train_smote , y_train_smote = smote.fit_resample(x_train , y_train)
#Model Training
models = {
    "Decision Tree" : DecisionTreeClassifier(random_state=42),
    "Random Forest-InformationGain" : RandomForestClassifier(random_state=42),
    "Random Forest-Entropy" : RandomForestClassifier(criterion='entropy'),
    "XGBoost" : XGBRFClassifier(random_state=42),
    "LogisticRegression" : LogisticRegression(max_iter=1000,solver='liblinear'),
    "SVM" : SVC(cache_size=100),
    "KNeighbors" : KNeighborsClassifier(n_neighbors=3),
    "Bernoulli Naive Bayes" : BernoulliNB(),
    "Gaussian Naive Bayes" : GaussianNB()
#perform 5 fold cross validation on each model
cv_scores = {}
for model_name , model in models.items():
    f=(model_name)
    print(f)
        scores = cross_val_score(model , x_train_smote , y_train_smote , cv=5 ,
scoring="accuracy")
    cv_scores[model_name] = scores
    print(f"{model_name} cross validation accuracy : {np.mean(scores):.2f}")
    print("."*50)
# Calculate mean accuracy for each model
mean_scores = {model: np.mean(scores) for model, scores in cv_scores.items()}
# Convert to lists for plotting
model_names = list(mean_scores.keys())
accuracy_scores = list(mean_scores.values())
```

```
# Create bar plot
plt.figure(figsize=(10, 5))
ax = sns.barplot(x=model_names, y=accuracy_scores, palette="pastel")
# Add accuracy labels on top of each bar
for i, score in enumerate(accuracy_scores):
    ax.text(i, score + 0.02, f"{score:.2f}", ha="center", fontsize=12, fontweight="bold")
# Add titles and labels
plt.xlabel("Models", fontsize=12)
plt.ylabel("Accuracy", fontsize=12)
plt.title("Cross-Validation Accuracy of Different Models", fontsize=14)
plt.ylim(0, 1)  # Accuracy is between 0 and 1
# Rotate model names for readability
plt.xticks(rotation=45)
# Show the plot
plt.show()
rfc = RandomForestClassifier(random_state=42)
rfc.fit(x_train_smote , y_train_smote)
#model evaluation
#evaluate on test data
y_test_pred = rfc.predict(x_test)
print("Accuracy Score : \n" , accuracy_score(y_test , y_test_pred))
print("Confusion Matrix : \n" , confusion_matrix(y_test , y_test_pred))
print("Classification Report : \n" , classification_report(y_test , y_test_pred))
```

```
Accuracy Score :
 0.7785663591199432
Confusion Matrix :
 [[878 158]
 [154 219]]
Classification Report :
              precision    recall  f1-score   support

           0       0.85      0.85      0.85      1036
           1       0.58      0.59      0.58       373

    accuracy                           0.78      1409
   macro avg       0.72      0.72      0.72      1409
weighted avg       0.78      0.78      0.78      1409
```

Fig 5.5: Main accuracy

```
#save the model as pickle file
import os
# Create the model/ directory if it doesn't exist
if not os.path.exists('model'):
    os.makedirs('model')
model_data = {"model": rfc, "features_name": x.columns.tolist()}
with open("model/customer_churn_prediction_model.pkl", "wb") as f:
    pickle.dump(model_data, f)
```

```
#Load the saved model and build a predictive system
import pickle
# Load the saved model
with open("model/customer_churn_prediction_model.pkl", "rb") as f:
    model_data = pickle.load(f)
print("Type of model_data:", type(model_data))  # Check the data type
print("Contents of model_data:", model_data)    # Print to inspect
# Ensure model_data is a dictionary before accessing keys
if isinstance(model_data, dict):
    loaded_model = model_data["model"]
    features_name = model_data["features_name"]
    print("Model loaded successfully!")
else:
    print("Error: model_data is not a dictionary!")
input_data = {
    "gender": "Female",
    "SeniorCitizen": 0,
    "Partner": "Yes",
    "Dependents": "No",
    "tenure": 1,
    "PhoneService": "No",
    "MultipleLines": "No phone service",
    "InternetService": "DSL",
    "OnlineSecurity": "No",
    "OnlineBackup": "Yes",
    "DeviceProtection": "No",
    "TechSupport": "No",
    "StreamingTV": "No",
    "StreamingMovies": "No",
    "Contract": "Month-to-month",
    "PaperlessBilling": "Yes",
    "PaymentMethod": "Electronic check",
    "MonthlyCharges": 29.85,
    "TotalCharges": 29.85 }
input_data_df = pd.DataFrame([input_data])
with open("model/encoders.pkl", "rb") as f:
    encoders = pickle.load(f)
for column , encoder in encoders.items():
    input_data_df[column] = encoder.transform(input_data_df[column])
#make a prediction
prediction = loaded_model.predict(input_data_df)
pred_prob = loaded_model.predict_proba(input_data_df)
print(prediction)
#results
```

```
print(f"Prediction: {'Churn' if prediction[0] == 1 else 'No Churn'}")
print(f"Prediction Probability : {pred_prob}")
```

CHAPTER 6
SOFTWARE TESTING

6.1 Testing Approach

In Machine Learning, a model is put to the test to judge how well it performs and how well it can predict outcomes from new data. The following steps are frequently included in the testing process:

- Data Split: The given dataset is split into two or three subsets: the training set, the optional validation set, and the test set. The validation set is used to fine-tune hyperparameters and model selection (if necessary), and the test set is saved for the final assessment. The training set is utilised to train the model.
- Feature Pre-Processing: Data in the test set is pre-processed using features in a similar way to the training set. This might entail actions like scaling, addressing missing values, normalisation, or feature engineering. To preserve consistency, it's crucial to employ the same preprocessing procedures as during training
- Model Prediction: The trained ML model receives the pre-processed test data and uses it to make predictions based on the correlations and patterns it has discovered in the training data. The predictions of the model may take the form of continuous values (such as regression) or class labels (such as binary or multi-class classification).
- Evaluation Metrics: The model's anticipated outputs are compared to the test set's ground truth labels or values. Depending on the nature of problem, several assessment measures are employed. Metrics including accuracy, precision, recall, F1 score, and Area Under the ROC Curve (AUC-ROC) are frequently employed for classification tasks. Metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), or Rsquared value can be used for regression jobs.
- Performance Evaluation: The model's performance is shown by the evaluation metrics derived from comparing the model predictions with the ground truth labels. It aids in assessing the model's ability to generalise to new data. By computing the evaluation metrics, it is possible to do a quantitative analysis of the performance. A qualitative analysis involves visualising the predictions using graphs, charts, or confusion matrices
- Iterative Refinement: Refinement through iteration may be necessary if the model's performance is unsatisfactory. This may entail changing hyperparameters, choosing various features, experimenting with different architectures or methods, or accumulating more training data. After that, the model is iteratively retrained and tested until the target performance is attained.In customer churn prediction using Machine Learning, the model is rigorously tested to evaluate how effectively it can identify customers likely to stop using a service. The process begins with splitting

the dataset into training, validation (optional), and testing subsets. The training set is used to teach the model, while the validation set helps in tuning hyperparameters and selecting the best algorithm. The test set is reserved for final evaluation. Before testing, feature pre-processing is performed on the test data using the same methods applied during training, such as handling missing values, normalization, encoding categorical data, and creating new relevant features. The trained model then makes predictions on this cleaned test data, typically classifying customers as likely to churn or not. To evaluate these predictions, classification metrics like accuracy, precision, recall, F1-score, and AUC-ROC are used, helping to assess how well the model distinguishes churners from non-churners. The performance evaluation includes both quantitative analysis through these metrics and qualitative analysis using tools like confusion matrices or ROC curves. If the results are not satisfactory, iterative refinement is conducted—this may involve retraining with more data, tuning model parameters, or trying different algorithms—until optimal performance in churn prediction is achieved.

6.1.1 Train and Test Split

When creating machine learning models for customer churn prediction, the train_test_split function in scikit-learn (sklearn) is an essential tool for splitting a dataset into training and testing subsets. This feature makes it possible to randomly divide the data according to a predetermined percentage, which aids in assessing the model's capacity to generalize to new, untested data. We utilize this function to divide our dataset when forecasting customer churn, allowing the model to learn patterns from a subset of the data and then be assessed on the remaining subset.

Table 6.1: Train and Test Split Data

Description	**Number of Customers**
No of customers' data for training	80% (e.g., 800 if total is 1000)
No of customers' data for testing	20% (e.g., 200 if total is 1000)

```
from sklearn.model_selection import train_test_split
# X is the feature matrix and y is the target variable (churn: yes/no)
X, y = load_churn_data()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In this code, The feature matrix, denoted as X, contains variables such as service type, monthly prices, client tenure, and consumption trends. The goal variable, y, indicates whether a client has churned; it is usually binary, with 0 denoting no churn and 1 denoting

churn. Test_size=0.2 ensures that the model is tested on data that was not observed during training by allocating 20% of the data for testing. Reproducibility of findings is ensured by random_state=42.

X_train, X_test, y_train, and y_test are the four subsets that are returned by the function. The training subsets are used to train the model, while the testing subsets are used to assess it. In order to evaluate the model's predicted performance on unseen consumers, this assessment may comprise evaluating accuracy, precision, recall, F1-score. Train_test_split helps validate the model's generalizability and dependability in real-world applications by simulating real-world deployment scenarios where the model must forecast churn for new customers.

6.2 Unit Testing

Table 6.2: Unit Testing Results

S.No.	Module	Testing Status
1	Data Loading	Successful!
2	Data Training	Successful!
3	Logistic Regression	Successful!
4	Decision Tree	Successful!
5	Random Forest – Entropy	Successful!
6	Random Forest – Information Gain	Successful!
7	SVM	Successful!
8	KNN	Successful!
9	Gaussian Naïve Bayes	Successful!
10	Bernoulli Naïve Bayes	Successful!
11	Model Training	Successful!
12	Accuracy and cross validation	Successful!
13	Dark/Light Theme	Successful!
14	Chatbot	Successful!
15	OTP Generation and Verification	Successful!
16	Feedback	Successful!

6.3 Model Validation Testing

Table 6.3: Model Validation Testing Result

S.No.	Module	Testing Status
1	Precision	Successful!
2	Recall	Successful!
3	F1 Score	Successful!
4	Accuracy	Successful!

Businesses may safely act on predicted insights when churn prediction is well tested to assure high performance, dependability, and continual development.

CHAPTER 7
RESULTS AND DISCUSSION

7.1 Running Snapshots

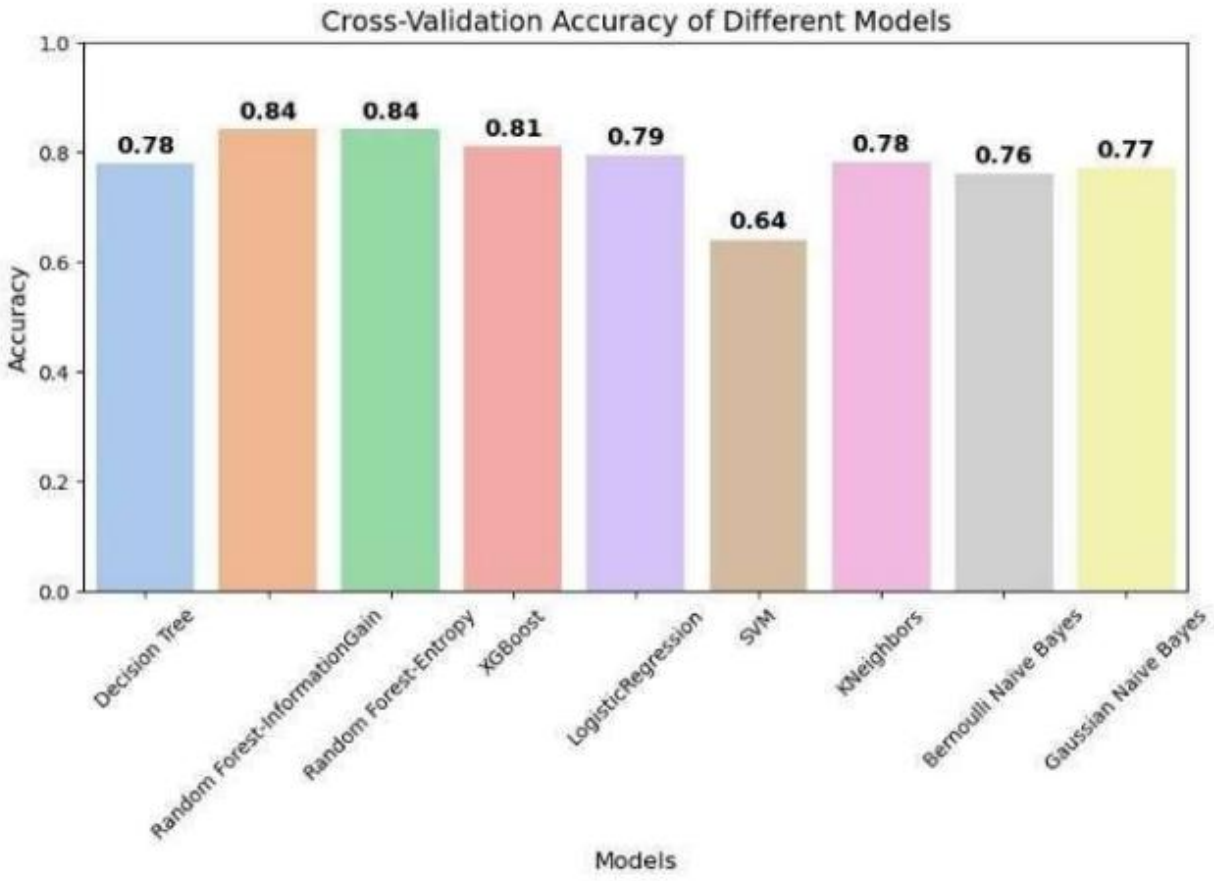

Fig 7.1: Accuracy Graph

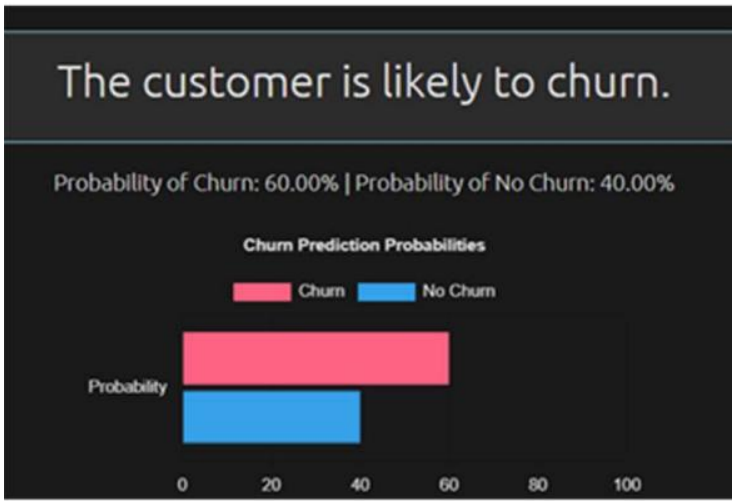

Fig 7.2: Result - Churn

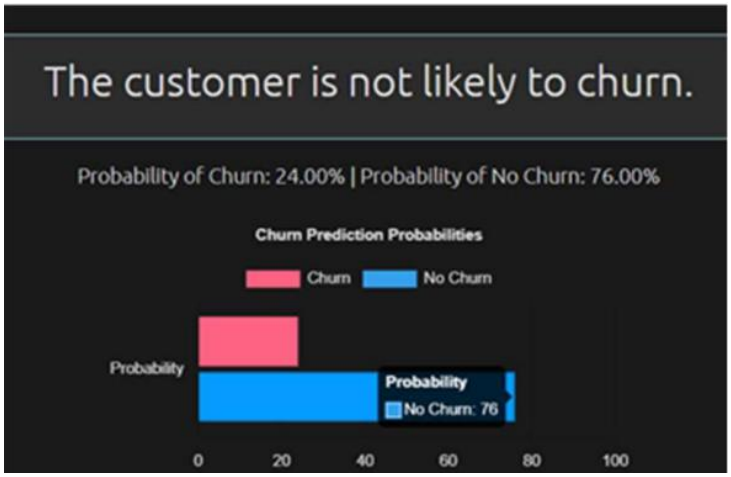

Fig 7.3: Result – No Churn

```
----------------------------------------------------
LogisticRegression
LogisticRegression cross validation accuracy : 0.79
.....................................................
Classification Report for LogisticRegression:

              precision    recall  f1-score   support

           0       0.91      0.76      0.83      1036
           1       0.55      0.79      0.65       373

    accuracy                           0.77      1409
   macro avg       0.73      0.78      0.74      1409
weighted avg       0.81      0.77      0.78      1409
```

Fig 7.4: Logistic Regression Result

```
Decision Tree
Decision Tree cross validation accuracy : 0.78
.....................................................
Classification Report for Decision Tree:

              precision    recall  f1-score   support

           0       0.83      0.78      0.81      1036
           1       0.48      0.55      0.51       373

    accuracy                           0.72      1409
   macro avg       0.65      0.67      0.66      1409
weighted avg       0.74      0.72      0.73      1409
```

Fig 7.5: Decision Tree Result

```
Random Forest-InformationGain
Random Forest-InformationGain cross validation accuracy : 0.84
.....................................................
Classification Report for Random Forest-InformationGain:

              precision    recall  f1-score   support

           0       0.85      0.84      0.85      1036
           1       0.58      0.58      0.58       373

    accuracy                           0.78      1409
   macro avg       0.71      0.71      0.71      1409
weighted avg       0.78      0.78      0.78      1409
```

Fig 7.6: Random Forest Information Gain Result

```
Random Forest-Entropy
Random Forest-Entropy cross validation accuracy : 0.84
.............................................
Classification Report for Random Forest-Entropy:

              precision    recall  f1-score   support

           0       0.85      0.85      0.85      1036
           1       0.58      0.58      0.58       373

    accuracy                           0.78      1409
   macro avg       0.71      0.71      0.71      1409
weighted avg       0.78      0.78      0.78      1409
```

Fig 7.7: Random Forest Entropy Result

```
-----------------------------------------------------
SVM
SVM cross validation accuracy : 0.64
.............................................
Classification Report for SVM:

              precision    recall  f1-score   support

           0       0.84      0.71      0.77      1036
           1       0.44      0.64      0.52       373

    accuracy                           0.69      1409
   macro avg       0.64      0.67      0.64      1409
weighted avg       0.74      0.69      0.70      1409
```

Fig 7.8: SVM Result

```
Classification Report for KNeighbors:

              precision    recall  f1-score   support

           0       0.86      0.73      0.79      1036
           1       0.47      0.67      0.55       373

    accuracy                           0.71      1409
   macro avg       0.67      0.70      0.67      1409
weighted avg       0.76      0.71      0.73      1409
```

Fig 7.9: KNN Result

```
Gaussian Naive Bayes
Gaussian Naive Bayes cross validation accuracy : 0.77
.............................................
Classification Report for Gaussian Naive Bayes:

              precision    recall  f1-score   support

           0       0.90      0.73      0.81      1036
           1       0.51      0.77      0.61       373

    accuracy                           0.74      1409
   macro avg       0.70      0.75      0.71      1409
weighted avg       0.79      0.74      0.75      1409
```

Fig 7.10: Gaussian Naïve Bayes Result

```
Bernoulli Naive Bayes
Bernoulli Naive Bayes cross validation accuracy : 0.76
................................................
Classification Report for Bernoulli Naive Bayes:

              precision    recall  f1-score   support

           0       0.90      0.71      0.79      1036
           1       0.49      0.79      0.61       373

    accuracy                           0.73      1409
   macro avg       0.70      0.75      0.70      1409
weighted avg       0.80      0.73      0.74      1409
```

Fig 7.11: Bernoulli Naïve Bayes Result

```
XGBoost
XGBoost cross validation accuracy : 0.81
................................................
Classification Report for XGBoost:

              precision    recall  f1-score   support

           0       0.89      0.77      0.82      1036
           1       0.53      0.73      0.61       373

    accuracy                           0.76      1409
   macro avg       0.71      0.75      0.72      1409
weighted avg       0.79      0.76      0.77      1409
```

Fig 7.12: XGBoost Result

CHAPTER 8
CONCLUSION

8.1 Conclusion Summary

Machine learning-based customer churn prediction gives companies who want to keep their key clients a competitive edge. Machine learning algorithms can precisely determine which consumers are most likely to depart by examining past customer data, including demographics, service consumption, and behavior patterns. This makes it possible for companies to take proactive measures to lower attrition rates and boost client loyalty through focused marketing, tailored communications, and enhanced customer support. Complex patterns and trends that are frequently overlooked in traditional analysis can be found with the use of machine learning algorithms like SVM, Random Forest, XGBoost, and Logistic Regression. To guarantee dependability and performance, these models are assessed using measures such as accuracy, precision, recall, and F1 score. Businesses may detect probable churners with high accuracy if they undertake performance evaluation, model optimization, and data pretreatment correctly. Thanks to open-source frameworks, cloud infrastructure, and contemporary machine learning methods, churn prediction models may now be implemented both cheaply and operationally. Overall, by improving customer happiness and retention, machine learning (ML)-based customer churn prediction not only helps minimize revenue loss but also promotes long-term growth. It gives companies the knowledge they need to be competitive in a changing industry and make data-driven choices.

8.2 Explanation

The following accuracy scores were observed:

- Random Forest (Entropy) as well as Random Forest (Information Gain) achieved the highest accuracy at 0.84.
- Logistic Regression followed with an accuracy of 0.81.
- XGBoost recorded an accuracy of 0.79.
- Decision Tree and SVM both achieved an accuracy of 0.78.
- Gaussian Naive Bayes had an accuracy of 0.77.
- Bernoulli Naive Bayes and KNN had the lowest accuracies at 0.76 and 0.64, respectively.

Table 8.1: Result Table

Model Name	Accuracy
Random Forest (Entropy)	84%
Random Forest (Information Gain)	84%
Logistic Regression	81%
XGBoost	79%
Decision Tree	78%
SVM	78%
Gaussian Naïve Bayes	77%
Bernoulli Naïve Bayes	76%
KNN	64%

8.3 Recommendation

Random Forest (Entropy) = Random Forest (Information Gain) > Logistic Regression > XGBoost > Decision Tree = SVM > Gaussian Naïve Bayes > Bernoulli Naïve Bayes > KNN

8.4 Limitations

Machine learning has several drawbacks even if it provides strong tools for forecasting client attrition. The reliance on data quality is a major drawback; erroneous, lacking, or out-of-date data can result in subpar model performance. The model's capacity to identify real churn can also be diminished by unbalanced datasets, when the number of non-churners much exceeds that of churners. Additionally, feature selection is crucial; omitting crucial characteristics might reduce predicted accuracy, while unnecessary or noisy features could confound the model. Furthermore, machine learning models—particularly intricate ones like ensemble or deep learning techniques—frequently act as "black boxes," making it challenging to determine the reasons behind a given customer's propensity to leave. Finally, since consumer behavior evolves over time, static models may become outdated and need constant retraining and observation to be successful. These drawbacks emphasize the necessity of cautious data handling, method choice, and continuous model assessment.

8.5 Challenges

Using machine learning to anticipate client attrition presents a number of difficulties. Organizations must collect data from several sources, sometimes in different forms, making data collection and preparation a significant problem. Another difficulty is making predictions in real-time or almost real-time, particularly for big datasets that need a lot of computing power. When processing sensitive consumer data, privacy and ethical issues can surface, necessitating stringent adherence to data protection laws. It takes effort and experience to select the best algorithm and adjust its hyperparameters for best results. Model retraining and flexibility are crucial because shifting market conditions or consumer behavior can also lower model accuracy. Ensuring cross-departmental collaboration is another difficulty, as technical teams and business stakeholders need to agree on objectives, standards, and how to interpret outcomes. Finally, there are technological challenges and infrastructure preparedness involved in integrating models with CRM systems and putting them into production. A well-rounded combination of technology, strategy, and subject expertise is needed to meet these difficulties.

FUTURE SCOPE OF THE PROJECT

As more companies use data-driven decision-making, the potential applications of machine learning (ML) for customer attrition prediction are growing quickly. It is anticipated that ML models will become increasingly complex, scalable, and accurate in detecting churn tendencies as big data grows and computing power increases. The viability of real-time churn prediction will grow, enabling companies to respond quickly to keep at-risk clients. Richer insights on churn behavior will be possible through the study of unstructured data, including customer reviews, social media activity, and support interactions, made possible by integration with technologies like artificial intelligence (AI), natural language processing (NLP), and deep learning. Furthermore, explainable AI (XAI) will be crucial in assisting companies in comprehending the reasons behind client attrition, encouraging openness and well-informed decision-making. Retention efforts will also be streamlined by integrating machine learning (ML) models into marketing automation tools and customer relationship management (CRM) systems. Churn prediction will move from being a supplementary tool to a key element of customer experience management as consumer expectations change and competition heats up. These developments will be especially advantageous for sectors like SaaS, retail, banking, and telecommunications. All things considered, ML-based churn prediction has enormous potential for the future to improve customer loyalty, lower revenue loss, and spur long-term company success.

.

REFERENCES

[1] **Khattak, A., Mehak, Z., Ahmad, H., et al.** (2023). Customer churn prediction using composite deep learning technique. Scientific Reports, 13, 17294. https://doi.org/10.1038/s41598-023-44396-w

[2] **Ullah, I., Raza, B., Malik, A. K., et al.** (2019). A Churn Prediction Model Using Random Forest Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector. IEEE Access, 7, 60134–60149. https://doi.org/10.1109/ACCESS.2019.2914999

[3] **Manzoor, A., Qureshi, M. A., Kidney, E., & Longo, L.** (2024). A Review on Machine Learning Methods for Customer Churn Prediction and Recommendations for Business Practitioners. IEEE Access, 12, 70434–70463. https://doi.org/10.1109/ACCESS.2024.3402092

[4] **Momin, S., Bohra, T., & Raut, P.** (2020). Prediction of Customer Churn Using Machine Learning. In A. Haldorai, A. Ramu, S. Mohanram, & C. Onn (Eds.), EAI International Conference on Big Data Innovation for Sustainable Cognitive Computing (pp. 237–245). Springer. https://doi.org/10.1007/978-3-030-19562-5_20

[5] **Ahmad, A. K., Jafar, A., & Aljoumaa, K.** (2019). Customer churn prediction in telecom using machine learning in big data platform. Journal of Big Data, 6, 28. https://doi.org/10.1186/s40537-019-0191-6

[6] **Agarwal, V., Taware, S., Yadav, S. A., et al.** (2022). Customer Churn Prediction Using Machine Learning. In 2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS) (pp. 893–899). IEEE. https://doi.org/10.1109/ICTACS54183.2022.9763987

[7] **Wang, X., Zhang, Y., & Liu, H.** (2024). Customer churn prediction model based on hybrid neural networks. Scientific Reports, 14, 79603. https://doi.org/10.1038/s41598-024-79603-9

[8] **Rudd, D. H., Huo, H., & Xu, G.** (2023). Causal Analysis of Customer Churn Using Deep Learning. arXiv preprint arXiv:2304.10604. https://arxiv.org/abs/2304.10604

[9] **Li, J.** (2024). Customer Churn Prediction using Machine Learning: A Case Study of E-commerce Data. International Journal of Computer Applications, 186(48), 22–28. https://ijcaonline.org/archives/volume186/number48/li-2024-ijca-924140.pdf

[10] **Sana, J. K., Abedin, M. Z., Rahman, M. S., & Rahman, M. S.** (2022). Data transformation based optimized customer churn prediction model for the

telecommunication industry. arXiv preprint arXiv:2201.04088. https://arxiv.org/abs/2201.04088

[11] **Bhattacharjee, S., Thukral, U., & Patil, N.** (2023). Early Churn Prediction from Large Scale User-Product Interaction Time Series. arXiv preprint arXiv:2309.14390. https://arxiv.org/abs/2309.14390

[12] **Rudd, D. H., Huo, H., Islam, M. R., & Xu, G.** (2023). Churn Prediction via Multimodal Fusion Learning: Integrating Customer Financial Literacy, Voice, and Behavioral Data. arXiv preprint arXiv:2312.01301. https://arxiv.org/abs/2312.01301

[13] **Barham, S., Aweisi, N., & Khalifeh, A.** (2023). A Review on Machine Learning-Based Customer Churn Prediction in the Telecom Industry. In 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 10284430). IEEE. https://doi.org/10.1109/CoDIT58514.2023.10284430

[14] **Bhattacharjee, S., Thukral, U., & Patil, N.** (2023). Early Churn Prediction from Large Scale User-Product Interaction Time Series. arXiv preprint arXiv:2309.14390. https://arxiv.org/abs/2309.14390

[15] **Sana, J. K., Abedin, M. Z., Rahman, M. S., & Rahman, M. S.** (2022). Data transformation based optimized customer churn prediction model for the telecommunication industry. arXiv preprint arXiv:2201.04088. https://arxiv.org/abs/2201.04088