



## **CU6051NI - Artificial Intelligence**

### **Assessment Weightage & Type**

**75% Individual Coursework**

### **“Customer Churn Prediction”**

**Student Name: Sandesh Prasad Paudel**

**London Met ID: 22015762**

**College ID: NP01CP4S220035**

**Assignment Due Date: 17<sup>th</sup> January, 2024**

**Assignment Submission Date: 17<sup>th</sup> January, 2024**

I confirm that I understand my coursework needs to be submitted online via MST Portal under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

## **Abstract**

Churning the data, saving customers: can ML be the lifeboat of telecommunication industry? The telecom industry navigates and cope up with the constantly changing customer market. This report will explore the intersection of machine learning and artificial intelligence for the telecommunication industry and their customer base, examine the ways in which these new analytic, predictive, and automating technologies are being used in real world telecom industry, and propose the potential future applications and associated challenges. The actual advantages of ML and AI are emphasized by real-world success stories such as Telecom Provider's 20% reduction in churn rates.

The report examines real-world applications of these technologies in the telecom industry and proposes potential future uses and. Different machine learning algorithms will be used to predict customer churn, with a focus on logistic regression, decision trees, and support vector machines. Here, the dataset of 34 columns is used. The report also includes an analysis of the customer churn prediction problem domain, a review of existing research on the topic, and an explanation of the proposed solution and its implementation. Here, in the report different evaluation metrices with the use of model after hyperparameter tuning is done. Using GridSearchCV and RandomizedSearchCV is found to be best. Thus after tuning, Logistic Regression and RandomForest outstands the other models in terms of accuracy.

## **Acknowledgements**

We would like to express our sincere gratitude to everyone who contributed to the success of customer churn prediction. Special thanks to Mr. Rajesh Mahara sir who guided us throughout the project. Mentors who provided guidance, support, and valuable insight throughout the development process. Additionally, I would like to acknowledge the efforts of mine who worked diligently to make this project a success. This project would not have been possible without the encouragement and support of the academic community.

## Table of Contents

<b>1. INTRODUCTION.....</b>	<b>1</b>
<b>1.1 Artificial Intelligence and Machine Learning .....</b>	<b>1</b>
<b>1.2 Customer Churn Prediction .....</b>	<b>3</b>
1.2.1 Problem domain .....	4
<b>2. BACKGROUND.....</b>	<b>5</b>
<b>2.1 Research work done on Telecom Customer Churn Prediction .....</b>	<b>5</b>
2.1.1 Telecom Customer Churn Prediction .....	5
2.1.2 Preferences to Telecom Companies of Nepal.....	6
2.1.3 Problems of Telecom Customer Churn Prediction in Current Scenario .....	6
2.2 Implementation of ML/AI in Customer Churn Prediction .....	7
<b>2.3 Advantages of using AI/ML in Telecom Customer Churn Prediction .....</b>	<b>8</b>
<b>2.4 Disadvantages of using AI/ML in Telecom Customer Churn Prediction .....</b>	<b>9</b>
<b>2.5 Datasets.....</b>	<b>10</b>
<b>2.6 Review and Analysis of Existing System .....</b>	<b>12</b>
2.6.1 Predicting Customer Churn in Telecom Sector .....	12
2.6.2 Churn Analysis in Telecommunication Industry.....	13
2.6.3 Churn Prediction using Supervised Machine Learning Algorithms .....	14
2.6.4 Prediction on Customer Churn in the Telecommunications Sector .....	15
2.6.5 Review on Customer Churn Prediction Using Machine Learning Techniques .....	16
<b>2.7 Review and Analysis of 5 existing work summaries.....</b>	<b>17</b>

<b>3. SOLUTION .....</b>	<b>18</b>
<b>3.1 Explanation of the proposed solution .....</b>	<b>18</b>
<b>3.2 Explanation of Machine Learning algorithms .....</b>	<b>18</b>
3.2.1 Logistic Regression .....	18
3.2.1.1 Sigmoid Function .....	18
3.2.2 Decision Tree .....	19
3.2.2.1. Entropy .....	19
3.2.2.2 Information Gain .....	19
3.2.3 Support Vector Machine (SVM) .....	20
3.2.3.1 Kernel in SVM .....	20
<b>3.3 Pseudocode of the proposed solution .....</b>	<b>21</b>
<b>3.4 Diagrammatic representations of the solution (Flowchart) .....</b>	<b>23</b>
<b>3.5 Development Process .....</b>	<b>24</b>
3.5.1 Tools Used .....	24
3.5.1.1 Ananconda .....	24
3.5.2 Libraries Used .....	25
3.5.2.1 Pandas .....	25
3.5.3 Development Process .....	26
3.5.3.1 Execute Notebook .....	26
3.5.3.2 Data Preprocessing .....	27
3.5.3.3 Feature Selection .....	31
3.5.3.4 Data Modelling with pipeline building for final preprocessing .....	33
3.5.4 Achieved Results .....	36
3.5.4.1 Logistic Regression results .....	36

3.5.4.2 Random Forest Classifier .....	37
3.5.4.3 Decision Tree Classifier.....	38
3.5.4.4 Naïve Bayes Classifier .....	39
3.5.4.5 Support Vector Machine (SVM) .....	40
3.5.4.6 Boxplot to show the accuracy of used models .....	41
3.5.4.7 ROC curve of used models.....	42
3.5.4.8 Final Overview of the model performance.....	44
 4. CONCLUSION .....	 45
 4.1 Analysis of Work Done.....	 45
 4.2 How solution addresses real-world problems? .....	 46
 4.3 Limitations of the system.....	 46
 BIBLIOGRAPHY .....	 47
 APPENDIX .....	 51
 A. Other Algorithms.....	 51
1. Random Forest .....	51
2 Naïve Bayes .....	52
2.1 Bayes Theorem .....	52
 B. Evaluation Metrics .....	 53
1. Accuracy.....	53
2. Precision.....	53
3. Recall (Sensitivity).....	53

4. F1 Score.....	54
<b>C. Others Tools Used.....</b>	<b>55</b>
3.5.1.2 Jupyter Notebook.....	55
3.5.1.3 Python 3 .....	56
<b>D. Others Libraries Used: .....</b>	<b>57</b>
3.5.2.2 NumPy.....	57
3.5.2.3 Matplotlib and Seaborn.....	57
3.5.2.4 Scikit-learn.....	58

## Table of Figures

Figure 1 Machine Learning types (Source: databasecamp.de).....	2
Figure 2 Customer churn prediction (Source: graphite-note.com) .....	3
Figure 3 Customer retention overview on data analysis .....	4
Figure 4 Telecom company churn data analysis (Source: slideteam.net) .....	5
Figure 5 ML/AI in Customer Churn Prediction (Source: pi.exchange) .....	7
Figure 6 Logistic Regression Sigmoid Function (Source: spiceworks.com) .....	18
Figure 7 Entropy Formula .....	19
Figure 8 SVM showing margin with hyperplane (Source: nl.mathworks.com) .....	20
Figure 9 Flowchart for telecom customer churn prediction .....	23
Figure 10 Logo of Anaconda.....	24
Figure 11 Anaconda Prompt .....	24
Figure 12 Importing and working with pandas .....	25
Figure 13 Opening Jupyterlab from command prompt .....	26
Figure 14 Opening Jupyter IDE from Web Browser .....	26
Figure 15 Importing required libraries.....	27
Figure 16 Read Datasets .....	27
Figure 17 Checking for null values in datasets .....	28
Figure 18 dropping the null values .....	28
Figure 19 Replacing similar data under ordinal 'No' in categorical columns.....	29
Figure 20 Visualize the overall target variable data .....	29
Figure 21 Checking for outliers in the column .....	30
Figure 22 Replacing outliers by mean value of particular columns .....	30
Figure 23 Code to generate heatmap.....	31
Figure 24 Heatmap to show correlated values .....	31
Figure 25 Drop the unnecessary features.....	32



Figure 26 Save modelling datasets into csv .....	32
Figure 27 Modelling Dataset in desktop .....	32
Figure 28 Independent and Target Variable .....	33
Figure 29 Dividing categorical features into ordinal and one hot .....	33
Figure 30 Checking the columns of ordinal and one hot .....	33
Figure 31 Pipeline building for converting categorical columns into numerical one.....	34
Figure 32 Final pipeline extension in numerical columns .....	34
Figure 33 Train Test Split .....	35
Figure 34 Logistic Regression .....	35
Figure 35 RandomForestClassifier .....	35
Figure 36 DecisionTreeClassifier .....	35
Figure 37 NaiveBayes .....	35
Figure 38 Support Vector Machine.....	35
Figure 39 Logistic Regression classification reports before tuning.....	36
Figure 40 Using GridSearchCV for tuning logistic regression .....	36
Figure 41 Logistic Regression classification reports after tuning with above parameters.....	36
Figure 42 Random Forest classification reports before tuning .....	37
Figure 43 Using RandomizedSearchCV for tuning random forest classifier .....	37
Figure 44 Random Forest classification reports after tuning with above parameters .....	37
Figure 45 Decision Tree Classification Reports before Tuning .....	38
Figure 46 Using GridSearchCV for tuning decision tree classifier.....	38
Figure 47 Decision Tree Classification Reports after Tuning.....	38
Figure 48 Naïve Bayes Classification Reports before Tuning .....	39
Figure 49 Using GridSearchCV for tuning Naïve Bayes .....	39
Figure 50 Naïve Bayes Classification Reports after Tuning .....	39
Figure 51 SVM Classification Reports before Tuning.....	40
Figure 52 Using GridSearchCV for tuning SVM.....	40

Figure 53 SVM Classification Reports after Tuning.....	40
Figure 54 Boxplot for model accuracy .....	41
Figure 55 Logistic Regression ROC Curve .....	42
Figure 56 SVM ROC Curve.....	42
Figure 57 RandomForest ROC Curve .....	43
Figure 58 NaiveBayes ROC Curve .....	43
Figure 59 Decision Tree ROC Curve .....	44
Figure 60 Random Forest Classification Problem.....	51
Figure 61 Probability Formula for Bayes Theorem .....	52
Figure 62 Accuracy Formula.....	53
Figure 63 Precision Formula.....	53
Figure 64 Recall Formula .....	53
Figure 65 F1-Score Formula .....	54
Figure 66 Jupyter Notebook Logo .....	55
Figure 67 Jupyter Notebook Interface .....	55
Figure 68 Logo of Python .....	56
Figure 69 Use of Python code in Jupyter IDE .....	56
Figure 70 Import and use of NumPy array.....	57
Figure 71 Importing seaborn and matplotlib .....	57
Figure 72 Use of Scikit Learn for Machine Learning .....	58

# 1. Introduction

## 1.1 Artificial Intelligence and Machine Learning

Artificial Intelligence (AI) is the ability of a machine to display human-like capabilities such as reasoning, learning, planning and creativity. It enables technical systems to perceive their environment, deal with what they perceive, solve problems and act to achieve a specific goal. The computer receives data which is already prepared or gathered through its own sensors such as a camera that processes it and responds. AI systems are capable of adapting their behavior to a certain degree by analyzing the effects of previous actions and working autonomously (Parliament, 2023).

Machine learning is a subset of AI that involves the development of algorithms and statistical models that enable computers to learn and make predictions or decisions without being explicitly programmed. Machine learning algorithms can be trained on data to identify patterns and make predictions about future events (Geeks for Geeks, 2023). Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. ML will help and identify the most relevant business questions and the data to answer them (IBM, 2023).

Under Machine Learning, we have different types which are supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning.

### 1. Supervised Learning

In supervised learning, the computer is taught by example. It learns from past data and applies the learning to present data to predict future events. In this case, both input and desired output data provide help to the prediction of future events (Sethi, 14th April 2020).

### 2. Unsupervised Learning

Unsupervised learning is the method that trains machines to use data that is neither classified nor labeled. It means no training data can be provided and the machine is made to learn by itself. The machine must be able to classify the data without any prior information about the data (Sethi, 14th April 2020).

### 3. Semi-supervised Learning

Semi-supervised learning is a machine learning technique that uses a small portion of labeled data and lots of unlabeled data to train a predictive model (Altex Soft, 2022).

### 4. Reinforcement Learning

Reinforcement Learning (RL) is a type of machine learning technique that enables an agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences (Bhatt, 2018).

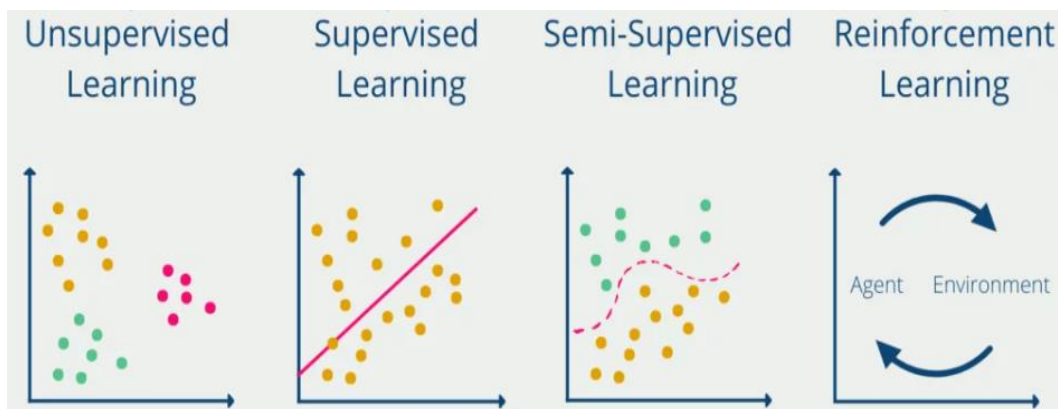


Figure 1 Machine Learning types (Source: databasecamp.de)

The development of AI and ML has the potential to transform various industries and improve people's lives in many ways. AI systems can be used to diagnose diseases, detect fraud, analyze financial data, and optimize manufacturing processes. ML algorithms can help to personalize content and services, improve customer experiences, and even help to solve some of the world's most pressing environmental challenges (Geeks for Geeks, 2023)

## 1.2 Customer Churn Prediction

Churn customer refers to the number of existing customers who may leave the service provider over a given period. These customers can be called as churners. The main aim of churn is to predict the churnable customers at the earliest, to identify the reason for leaving. This is because acquiring new customers often costs more than retaining existing ones. Once you've identified customers at risk of churn, you need to know exactly what marketing efforts you should make with each customer to maximize their likelihood of staying (Wiryaseputra, 2022). Customer churn prediction helps industries by proactively identifying services that customers are leaving of. This enables businesses to implement targeted retention strategies, optimizing resource allocation and reducing customer acquisition costs. Customer churn prediction can be done on different fields ranging from finance and banking, insurance companies, ecommerce, subscription-based business and telecommunication industries.

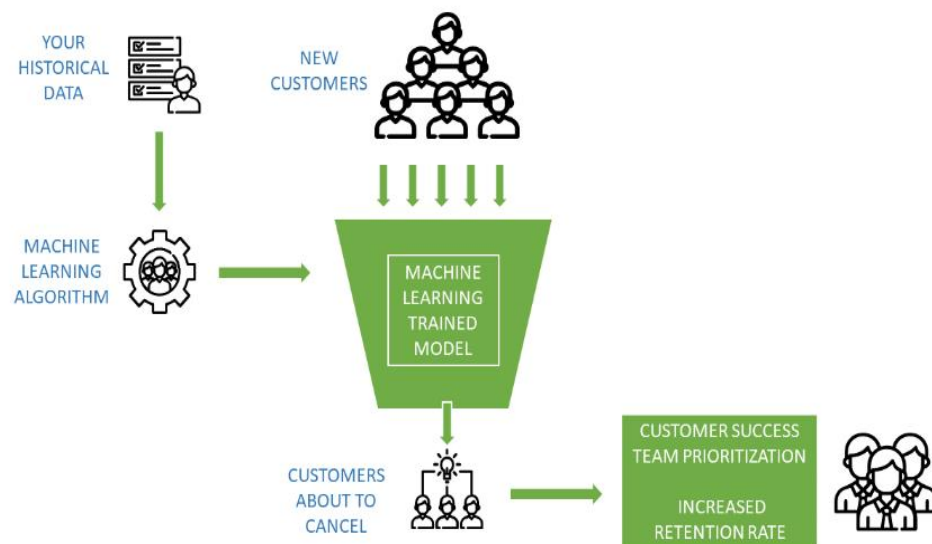


Figure 2 Customer churn prediction (Source: graphite-note.com)

There are basically two types of churners with voluntary churn and involuntary churn. Involuntary churn happens when a customer can't continue using a service for reasons that are partially or entirely out of their control, most of the time related to payment processing issues. Voluntary churn is when customers leave because they are no longer happy with the product, its features, or its pricing.

### 1.2.1 Problem domain

With the terrific growth of digital data and associated technologies, there is an emerging trend, where industries become rapidly digitized. These technologies are providing great opportunities to identify and resolve different problems. In particular, the telecommunication industry is facing a serious problem of customer churn relating to, the customers who are going to abandon their established relation with the business/network in the near future. The telecom sector is dynamic, with rapidly changing technologies with changes in customer preferences, and intense competition in the market. For these companies to stay competitive, it is important to plan for and handle loss of customers.

I find telecom companies more feasible to choose for my coursework since it provides a practical advantage for maximizing my main problem domain which is customer retention. I can predict customer churn to analyze large datasets of customer relationships usage trends, and service data. This choice aligns with the goal of not only minimizing revenue loss but also fostering long-term customer loyalty and maintaining a competitive edge in the market.



Figure 3 Customer retention overview on data analysis

## 2. Background

### 2.1 Research work done on Telecom Customer Churn Prediction

#### 2.1.1 Telecom Customer Churn Prediction

Companies must reduce customer churn because it weakens the company. A survey showed that the annual churn rate in the telecom industry ranges from 20% to 40%, and the cost of retaining existing customers is 5–10 times lower than the cost of obtaining new customers. The cost of predicting churn customers is 16 times lower than that for obtaining new customers. Decreasing the churn rate by 5% increases the profit from 25% to 85%. This shows that customer churn prediction is important for the telecom sector. Telecom companies consider customer relationship management as an important factor in retaining existing customers and preventing customer churn (Tianpei Xu, 2021). Recent studies show that the main objective of these telecom industries are to identify the valuable churn customers using a huge amount of data received from the telecommunication industry. The industry then targets those customers or clients and provides them with special incentives, offerings, and plans except normal customers (Md. Ashraful Haque, December 2023).

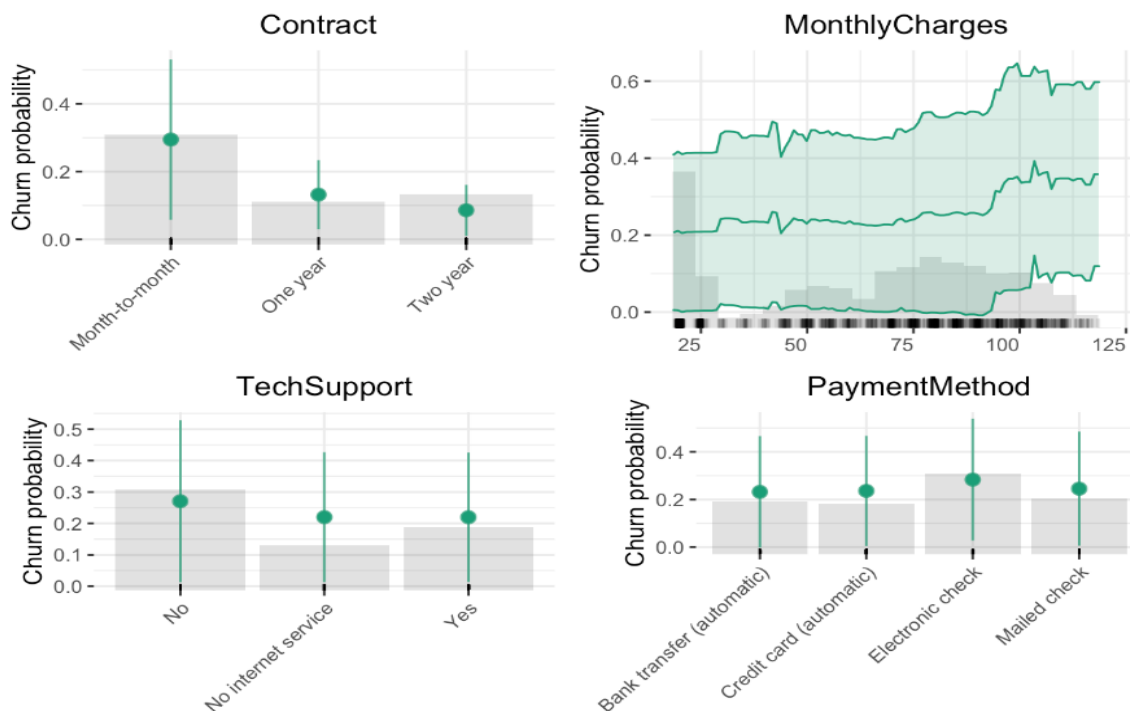


Figure 4 Telecom company churn data analysis (Source: slideteam.net)

### **2.1.2 Preferences to Telecom Companies of Nepal**

In the competitive telecom industries of Nepal, customer satisfaction is important. Four key factors which affected NTC are price, network quality, service experience, and innovative offerings. From that study, I acknowledged that expensive plans, missed calls, unsatisfactory customer support, and a lack of new features can cause the customers to switch providers. Telecom companies can improve their approaches by having a better understanding of these causes. Reliable networks, devoted customer service, specific pricing, and intriguing, customer-focused services can all help retain valuable customers and result the flow of loss (Basnet, 2022). Similarly, there have been rumors these days regarding the sale of Ncell. This topic can indirectly relate to customer churn as after the sale, Ncell will have different management with new strategies. During this transition phase, many customers may not like new offerings or services. So, analyzing customer behavior and fields of deflation will cause Ncell to have predictor system like customer churn prediction.

### **2.1.3 Problems of Telecom Customer Churn Prediction in Current Scenario**

Model training is difficult when there are few churners and imbalanced data. Regardless of their advantages, different algorithms are tough to understand, which makes it more difficult to figure out why customers leave. Furthermore, models find it challenging to stay up to date with changing market trends and consumer preferences. The challenge is to strike a balance between ROI maximization as well as affordable retention strategies (Saran Kumar A., 2023). While techniques like neural networks and gradient boosting are powerful, they also have drawbacks. Training and evaluation are made difficult by imbalanced data, feature engineering is tough, and data accessibility is restricted. In order to facilitate targeted retention strategies and make it difficult to understand why customers churn, complex models lack transparency. Static forecasts also have problems being customized because they can't offer personalized insights or retention strategies that are suited to the particular requirements and circumstances of each individual customer (Karamollaoğlu, et al., 2021). So, the accuracy of the model depends upon the scenario or the datasets one have.



## 2.2 Implementation of ML/AI in Customer Churn Prediction

AI, and ML are good at processing and analyzing large amounts of structured and unstructured data. AI and ML algorithms can easily identify relevant features or variables that contribute to customer churn. By carefully selecting the most impactful predictors, the churn prediction models can concentrate on the key factors influencing churn, while streamlining the churn prediction process. Telecom customer churn prediction is the classification problem under supervised learning. These techniques are frequently used to categorize customers into two categories: churners or non-churners based on previous data (LeewayHertz, 2023).

AI driven systems can recognize early indicators of future customer churn, causing timely notifications and beforehand measures to stop it. By categorizing customers based on a range of certain features or preferences, ML-powered churn prediction models can produce specific strategies to keep customers. Thus, in this way AI/ ML can contribute this telecommunication and other subscriber based companies to keep up with their business profitably. Because of a highly competitive market and a wide range of products/services (Internet, television, mobile networks, etc.), such giants as AT&T, Sprint, Vodafone, and T-Mobile have already utilized machine learning for reducing churn rate. Today even smaller companies and startups try AI applications right after their services enter the market (Fayrix, 2022).

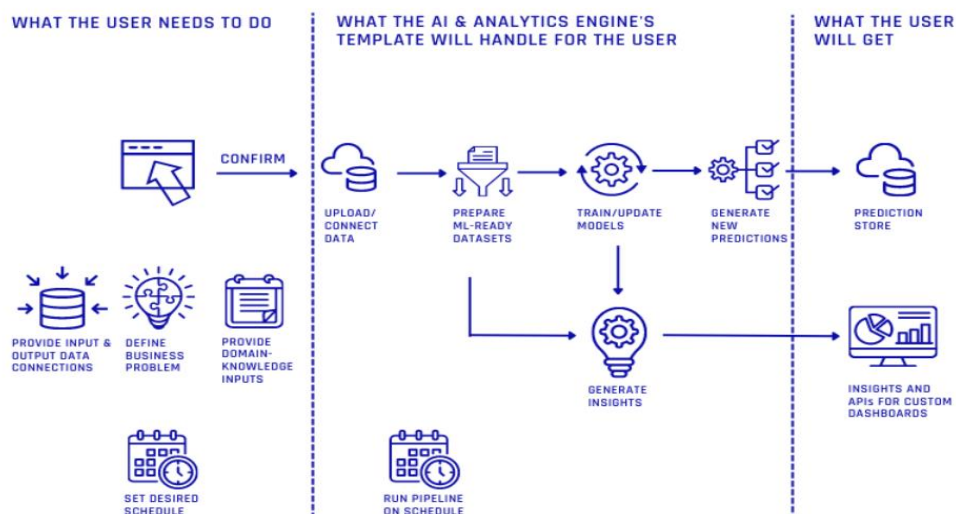


Figure 5 ML/AI in Customer Churn Prediction (Source: pi.exchange)

## 2.3 Advantages of using AI/ML in Telecom Customer Churn Prediction

Knowing in front when and who is most likely to leave can make all the difference. This is where artificial intelligence (AI) and machine learning (ML) come into play, providing a powerful set of resources for accurate churn prediction. The following are some of the main advantages of telco customer churn prediction using ML/AI:

### 1. Increase in accuracy

Machine learning algorithms possess the ability to examine huge amounts of data and identify similarities that traditional methods could fail to recognize. As a result, predictions of customer retention could become more accurate.

### 2. Data-driven insights

Various data sources, such as customer usage patterns, billing data, and customer interactions, can be used by ML models. This provides a more accurate and data-driven understanding of customer behavior.

### 3. Continuous learning

Since, ML models are capable of gathering up latest data, the churn prediction system is going to evolve over time. This adaptation is important in changing environment of telecom industries.

### 4. Cost reduction

Telecom companies can improve the productivity of their retention strategies while lowering overall costs by focusing their efforts on high-risk customers with accurate prediction of churn rate.

### 5. Enhanced customer experience

Telecom companies can modify the services they provide to customers according to their expectations, through the use of machine learning (ML) for churn prediction. Understanding the preferences of customers allows the offer of customized services, which in return improves the customer experiences with the company in a long run.

## **2.4 Disadvantages of using AI/ML in Telecom Customer Churn Prediction**

As AI/ML is a technology too. So, it is obvious that there are some disadvantages of them during customer churn prediction. Thus, below are the main disadvantages of it and they are:

### **1. Data and privacy concerns**

Privacy issues arise when customer data is used in machine learning models. To stay away from issues with the law and ethics, companies must handle customer data responsibly and maintain data protection regulations.

### **2. Initial companies costs**

The initial expenses of infrastructure, training, and hiring skilled workers come along with the use of ML/AI systems. Thus, the initial cost may be expensive for smaller telecom companies.

### **3. Dependence over quality data**

High-quality and relevant training data is essential for successful machine learning models. An error in predicting and an unreliable insights can result from unbalanced or incomplete input data.

### **4. Limited human understandings**

It becomes more difficult to understand the decision-making process as machine learning (ML) models get more complex. It may become difficult for telecom companies to defend their predictions to customers, which might reduce trust and transparency.

## 2.5 Datasets

Dataset is a collection of various types of data stored in a digital format. Data is the key component of any machine learning project. Datasets primarily consist of images, texts, audio, videos, numerical data points, etc. for solving various artificial intelligence related problems such as face recognition, sentiment analysis, stock market prediction, and customer churn rate prediction (Khan, 2022).

Since most of the data columns that were used in Kaggle were till churn with value yes/1 or no/0, I have also used the columns up to their, for easiness in prediction of my model.

1. CustomerID: Unique identifier for each customer in the dataset.
2. Gender: Male or Female.
3. Name: Name of a customer.
4. Partner: Whether the customer has a partner (Yes, No).
5. Dependents: Whether the customer has dependents (children or elderly parents) (Yes, No).
6. Tenure: Number of months the customer has been with the company.
7. Phone Service: Whether the customer has a phone service plan (Yes, No).
8. Multiple Lines: Whether the customer has multiple phone lines on their plan (Yes, No).
9. Internet Service: Type of internet service the customer has (DSL, Fiber, None).
10. Device Protection: Whether the customer has device protection insurance (Yes, No).
11. Tech Support: Whether the customer has opted for additional technical support (Yes, No).
12. Streaming TV: Whether the customer has a streaming TV subscription (Yes, No).
13. Streaming Movies: Whether the customer has a streaming movie subscription (Yes, No).
14. Contract: Type of contract the customer has (Month-to-month, One-year, Two-year and more).
15. Paperless Billing: Whether the customer receives paperless billing (Yes, No).

16. Payment Method: Primary method of payment for the customer's account (eSewa, Cash, Mobile Banking, Credit Card)
17. Monthly Charges: Monthly service charges for the customer's plan (Numerical Data).
18. Total Charges: Total amount billed to the customer over the entire contract period (Numerical Data)
19. Churn: Whether the customer has churned (cancelled their service) in the past month (Yes, No).
20. Age: The age of the customer (Numerical Data).
21. Married: Indicates whether the customer is married.
22. Avg Monthly Long Distance Charges: The average monthly charges incurred by the customer for long-distance services (Numerical Data).
23. Avg Monthly GB Download: The average monthly data consumption in gigabytes (Numerical Data).
24. Streaming Music: Indicates whether the customer uses a streaming music service.
25. Premium Tech Support: Indicates whether the customer subscribes to premium technical support.
26. Unlimited Data: Indicates whether the customer has an unlimited data plan.
27. Total Refunds: The total amount refunded to the customer (Numerical Data).
28. Total Extra Data Charges: The total charges for extra data usage (Numerical Data).
29. Total Long Distance Charges: The overall charges for long-distance services (Numerical Data).
30. Total Revenue: The total revenue generated from the customer (Numerical Data).
31. Under 30: Age categorized as below 30.
32. Senior Citizen: Whether the customer is above 65 years old (1, 0).
33. OnlineSecurity: Indicates whether the customer has access to online security services (Yes or No)
34. OnlineBackups: Indicates whether the customer uses or has access to online backup services (Yes or No).

## **2.6 Review and Analysis of Existing System**

### **2.6.1 Predicting Customer Churn in Telecom Sector**

Title: Churn Prediction of Customer in Telecom Industry using Machine Learning Algorithms

Authors: V.Kavitha, G.Hemanth Kumar, SV. Mohan Kumar, M. Harish

Publisher: International Journal of Engineering Research & Technology (IJERT)

Year. 2020

Pages. 181-184

Vol. 09

Issues 5

#### **Summary**

The author addresses the issues regarding the churn prediction in telecom industry using machine learning algorithms. Further, there has been loss due to competitor offers or network issues. They got the datasets from Kaggle and by using different data processing and feature selection, they divided the data into test and train sets with 80% and 20% respectively. Here, the proposed model that is employed are Logistic Regression, Random Forest, and XGBoost to predict the potential churn. The paper shows the importance of accurate datasets with their improved data preprocessing techniques. While, the confusion matrix of their study gives Random Forest, Logistic Regression and XGBoost with the precision, f1-score, recall and support value, simultaneously. Three tree-based algorithms were chosen because of their applicability and diversity in this type of application. By using Random Forest, XGBoost, and Logistic regression, they got more accuracy comparing other algorithms. The results indicates that the Random Forest Algorithm produced better accuracy in comparison with other methods (V.Kavitha, 2020).

## 2.6.2 Churn Analysis in Telecommunication Industry

Title: Churn Analysis in Telecommunication Industry using Machine Learning Techniques

Authors: Vibhor Shah, Deepak Harbola, Dr.S.Thenmalar

Publisher: Annals of R.S.C.B

Year: 2021

Vol.25

Pages. 4321-4326

Issues.05

### Summary

The author discusses a study on customer churn in the telecom sector, emphasizing the use of big data techniques and machine learning to predict and reduce churn rates. The proposed work outlines a system architecture for predicting customer churn, involving data collection, pre-processing, preparation, prediction and visualization. The implementation section includes analysis of a dataset obtained from Kaggle where exploratory analysis is performed using pandas profiling and the conversion of categorical value into numerical one. The algorithms used for prediction are Logistic Regression, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Random Forest. The dataset is divided into training and testing sets as 70% and 30% respectively and various Python libraries are used for visualization. The results and discussion section presents the accuracy scores of each algorithm with Logistic Regression as highest by having 81.14%, while SVM, Random Forest, KNN and Decision Tree with 80.66%, 79.38%, 76.87% and 73.27% respectively. Here, I find Support Vector and Random Forest also in the range of highest accuracy. The study concludes that retaining existing customers is economically beneficial, and the proposed system can help organizations identify customers likely to churn, enabling targeted retention efforts (Vibhor Shah, 2021).

### 2.6.3 Churn Prediction using Supervised Machine Learning Algorithms

Title: Churn Prediction using Supervised Machine Learning Algorithms- Impact of Over Sampling

Authors: Malika Naresh Panchal, Dr. Anala A Pandit

Publisher: International Research Journal of Engineering and Technology (IRJET)

Year: 2020

Vol. 7

Pages. 1014-1019

Issues. 11

#### Summary

The paper discusses the importance of customer retention in telco industry. The focus is on predicting customer churn using supervised machine learning algorithms. Two customer types, post-paid and prepaid, are considered, with prepaid customers having a higher probability of switching to competitors. Various machine learning models, including Random Forest, C5.0, KNN Classifier, Logistic Regression, XGBoost, and LightGBM, are used and evaluation metrics like accuracy, recall, precision, and F-measure. The paper addresses challenges in churn prediction where the imbalanced data needs to be handled so oversampling techniques like SMOTE is to be used. The study compares the performance of models before and after oversampling, with LightGBM identified as the best-performing model, considering a low false negative ratio and high accuracy on the oversampled data. Thus, Logistic Regression stands high in accuracy with 78% before and 79% after oversampling for predicting churn prediction which withstand LightGBM. Supervised learning like Random Forest also joins the race with an accuracy of 78% before and after oversampling of data. While KNN is listed low with 74% before and 73% after oversampling in data (Malika Naresh Panchal, 2020).



### **2.6.4 Prediction on Customer Churn in the Telecommunications Sector**

Title: Prediction on Customer Churn in the Telecommunications Sector Using Discretization and Naïve Bayes Classifier

Authors: Tan Yi Fei, Lam Hai Shuan, Lai Jie Yan Guo Xiaoning, Soo Wooi King

Publisher: International Journal of Advances in Soft Computing and its Applications (IJSTAC)

Year: 2017

Vol. 9

Pages. 24-35

Issues. 03

#### **Summary**

The paper address the issue of customer churn with the use of Naïve Bayes classifier algorithm for predicting customer churn rate. Methodology including data processing, feature selection, data analysis using Equal-Width Discretization (EWD) and K-Means clustering, and data validation through model evaluation. The study compares the performance of EWD and K-Means in combination with the Naïve Bayes classifier, emphasizing the importance of data preprocessing and feature selection. The results indicate that eliminating highly correlated attributes can significantly reduce computational time. The true-false analysis and overall accuracy of different models (A, B, C, D, and E) are presented, with Model D showing the highest accuracy in predicting true positive values for customer churn. The models' performances are compared with sensitivity and specificity in predicting churn. The paper emphasizes to choose the appropriate model for prediction in telco industry by considering various factors ranging from computational efficiency to prediction accuracy (Tan Yi Fei, 2017).

### **2.6.5 Review on Customer Churn Prediction Using Machine Learning Techniques**

Title: Review on Customer Churn Prediction Using Machine Learning Techniques

Authors: Kiran Dhangar, Prateek Anand

Publisher: International Journal of Innovations in Engineering Research and Technology

Year: 2021

Vol. 8

Pages. 193-201

Issues. 5

#### **Summary**

This paper shows the problem of customer churn in different industries with the inclusion of telecommunication too. Along with that, a methodology is used which have six phases for churn prediction and they are data pre-processing, exploratory data analysis, feature selection, data splitting into training and test sets, application of prediction models, ensemble approaches, hyperparameter tuning, k-fold cross-validation and performance analysis through AUC/RUC curves. The introduction includes the financial impacts of customer churn and the significance of ML in prediction. Some of the algorithms that were noted enlists decision trees, random forests, XGBoost, SVM, Naive Bayes and logistic regression. Finally, the AUC/RUC curve was used to analyze the findings obtained on the test set. Random Forest and SVM were shown to have the highest accuracy of 87 percent and 84 percent, respectively. Random Forest achieves the greatest AUC score of 94.5 percent, while SVM classifiers obtain 92.1 percent, outperforming others (Kiran Dhangar, 2021).

## 2.7 Review and Analysis of 5 existing work summaries

Above studies states that during a customer churn prediction in telecommunication industry, the importance of using ML algorithms such as Logistic Regression, Random Forest, SVM, Naïve Bayes and Decision Tree should be used. In the work by V.Kavitha (2020), Random Forest have been the most accurate algorithm with ability of handling different datasets. Similarly, in the study by Vibhor Shah (2021), Logistic Regression gives the highest accuracy with other evaluation metrices. That's why I find it effective to solve my problem domain too. While Malika Naresh Panchal's work (2020) introduces the concepts of oversampling and the use of LightGBM along with Logistic Regression and Random Forest showing Logistic Regression with highest accuracy too. Tan Yi Fei (2017) employs a Naïve Bayes classifier with its importance in feature selection. In Kiran Dhangar's study (2021), Random Forest and SVM showcases high accuracy involving AUC/RUC curves.

From the above 5 existing work I found that Logistic Regression, Random Forest, SVM, Decision Tree and Naïve Bayes algorithms are best suited for handling the customer churn prediction of telecommunication industry. From the accuracy and other evaluation metrics of Logistic Regression, Random Forest and SVM in the above existing work, I analyzed that I will be using these algorithms at any cost since the features of these datasets are somewhat similar to the datasets that I'm using. Therefore, I'm expecting a better results from these supervised classification model.

### 3. Solution

#### 3.1 Explanation of the proposed solution

During the telecom customer churn predictions, there will be the use of a particular datasets containing different information to build a predictor system. The main goal of developing this model is to predict churn rate of customer based on features and parameters. To build this predictor system, different machine learning algorithms and evaluation metrics are used. Machine learning algorithms like Logistic Regression, Decision Tree, Random Forest, Naïve Bayes and Support Vector Machine (SVM) must be deployed for the development of intended system. While evaluation metrics like accuracy, precision, recall and F-Measure are used to check the overall performance of the model that is built.

#### 3.2 Explanation of Machine Learning algorithms

##### 3.2.1 Logistic Regression

Logistic regression is a supervised machine learning algorithm that accomplishes binary classification tasks by predicting the probability of an outcome, event, or observation. The model delivers a binary or dichotomous outcome limited to two possible outcomes as yes or no, 0 or 1, or true or false. For example, 0 represents a negative class; 1 represents a positive class. In logistic regression, the y variable takes only two values. Hence, one can effectively classify data into two separate classes if linearly separable data is used (Kanade, 2022).

##### 3.2.1.1 Sigmoid Function

In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

$$S(z) = \frac{1}{1 + e^{-z}}$$

##### Note

- $s(z)$  = output between 0 and 1 (probability estimate)
- $z$  = input to the function (your algorithm's prediction e.g.  $mx + b$ )
- $e$  = base of natural log

Figure 6 Logistic Regression Sigmoid Function (Source: spiceworks.com)

### 3.2.2 Decision Tree

Decision Tree algorithm belongs to supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree can be used for solving both regression and classification problems. The goal of using a decision tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data or training data. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes (Chauhan, 2022).

Some variable selection criteria in decision tree are:

#### 3.2.2.1. Entropy

In the context of Decision Trees, entropy is a measure of disorder or impurity in a node. Thus, a node with more variable composition, such as 2Pass and 2 Fail would be considered to have higher entropy than a node which has only pass or only fail. The maximum level of entropy or disorder is given by 1 and minimum entropy is given by a value 0 (Dash, 2022).

Entropy is measured by:

$$E = - \sum_{i=1}^n p_i \log_2(p_i)$$

Figure 7 Entropy Formula

Here,  $p_i$  is the probability of randomly selecting an example in class.

#### 3.2.2.2 Information Gain

Now, calculate the average entropy across both or all the nodes and then the change in entropy of parent node and child node. This change in entropy is termed Information Gain. The formula for information gain:

Information Gain= Entropy (Parent) - Average Entropy (Child)

### 3.2.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning machine learning algorithm that can be used for both classification and regression challenges. However, it is mostly used in classification problems, such as text classification. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have), with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the optimal hyper-plane that differentiates the two classes very well (Ray, 2023).

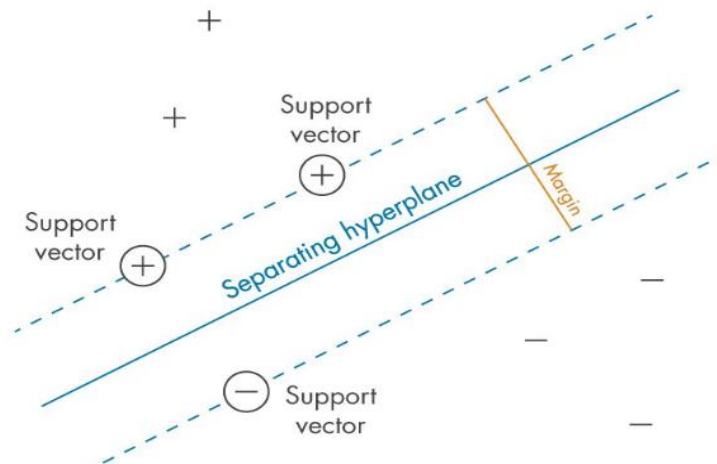


Figure 8 SVM showing margin with hyperplane (Source: nl.mathworks.com)

#### 3.2.3.1 Kernel in SVM

SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions (Data Flair, 2022). These functions can be different types. And they are:

- Polynomial Kernel
- Gaussian Kernel
- Laplace RBF Kernel
- Gaussian RBF Kernel

Note: Explanation of other two algorithms are explained below in appendix section [Other Algorithms](#).

While evaluation metrics are explained here [Evaluation Metrics](#).

### 3.3 Pseudocode of the proposed solution

**START**

**IMPORT** required libraries

**IMPORT** telecom customer dataset

**READ** the dataset

**IF** dataset contains duplicate rows

**ELIMINATE** duplicate rows

**END IF**

**IF** any dataset value have outliers or null values

**CALCULATE** mean (Column values)

**REPLACE** null values by mean

**END IF**

**IF** found highly correlated columns

**REMOVE** correlated column

**END IF**

**SET** columns for modelling

**SAVE** datasets into csv file

**DIVIDE** target and dependent variable

**CONVERT** categorical columns into numerical using pipeline

**OBTAIN** all the numerical columns for model training

**PROCESS** new dataset with numerical columns

**SPLIT** dataset into train and test

**TRAIN** dataset using Logistic regression, Decision tree, Random Forest, Naïve Bayes and SVM

**TEST** model using test dataset

**PERFORM** evaluation metrics

**VISUALIZE** the prediction

**STOP**



### 3.4 Diagrammatic representations of the solution (Flowchart)

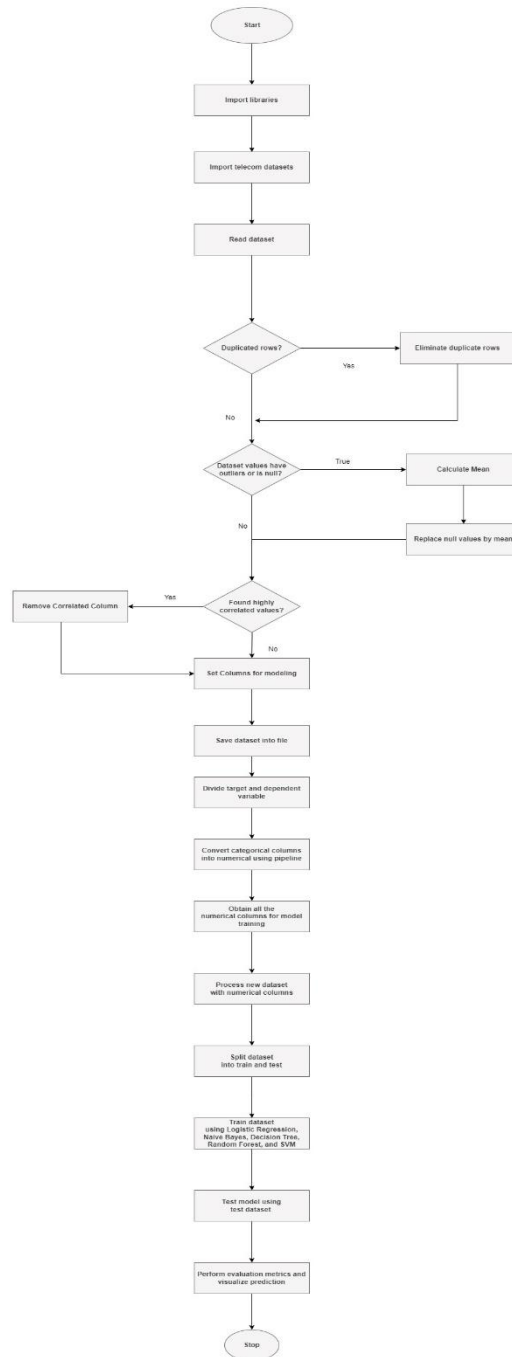


Figure 9 Flowchart for telecom customer churn prediction

## 3.5 Development Process

### 3.5.1 Tools Used

#### 3.5.1.1 Ananconda



Figure 10 Logo of Anaconda

Anaconda Python is a free, open-source platform that allows you to write and execute code in the programming language Python. It is by continuum.io, a company that specializes in Python development. The Anaconda platform is the most popular way to learn and use Python for scientific computing, data science, and machine learning. People like using Anaconda Python because it simplifies package deployment and management. It also comes with a large number of libraries/packages that you can use for your projects. Since Anaconda Python is free and open-source, anyone can contribute to its development (Ellis, 2023).

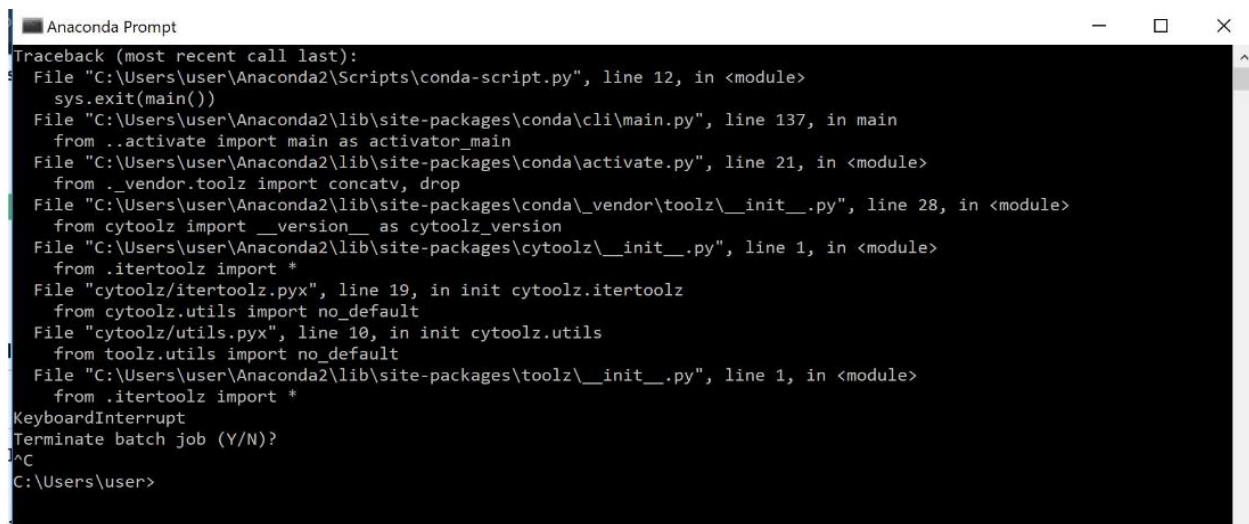
A screenshot of the Anaconda Prompt terminal window. The window title is "Anaconda Prompt". The terminal shows a traceback error starting with "Traceback (most recent call last):". The error path goes through several files: "C:\Users\user\Anaconda2\Scripts\conda-script.py", "C:\Users\user\Anaconda2\lib\site-packages\conda\cli\main.py", "C:\Users\user\Anaconda2\lib\site-packages\conda\activate.py", "C:\Users\user\Anaconda2\lib\site-packages\conda\\_vendor\toolz\\_init\_.py", "C:\Users\user\Anaconda2\lib\site-packages\cytoolz\\_init\_.py", "C:\Users\user\Anaconda2\lib\site-packages\cytoolz\iterutils.py", "C:\Users\user\Anaconda2\lib\site-packages\cytoolz\utils.py", and "C:\Users\user\Anaconda2\lib\site-packages\toolz\\_init\_.py". The error ends with "KeyboardInterrupt". Below the traceback, it asks "Terminate batch job (Y/N)?" and the user has entered "N". The prompt "C:\Users\user>" is visible at the bottom.

Figure 11 Anaconda Prompt

Note: Other Tools Used is in appendix section. [C. Others Tools Used](#)

### 3.5.2 Libraries Used

#### 3.5.2.1 Pandas

Pandas is a data manipulation package in Python for tabular data. That is, data in the form of rows and columns, also known as DataFrame. Intuitively, you can think of a DataFrame as an Excel sheet. Pandas' functionality includes data transformations, like sorting rows and taking subsets, to calculating summary statistics such as the mean, reshaping DataFrame, and joining DataFrame together. It works well with other popular Python data science packages, often called the PyData ecosystem, including NumPy (Chugh, 2023).

```
[2]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
import random
warnings.filterwarnings("ignore")

from sklearn.metrics import confusion_matrix, classification_report

[3]: df = pd.read_csv('Dataset_new.csv')
df.head(10)
```

Figure 12 Importing and working with pandas

Note: Remaining libraries used is in appendix section. [D. Others Libraries Used:](#)

## 3.5.3 Development Process

### 3.5.3.1 Execute Notebook

```
C:\Users\ASUS>jupyter notebook
[I 2024-01-16 21:32:37.992 ServerApp] Package notebook took 0.0000s to import
[I 2024-01-16 21:32:38.034 ServerApp] Package jupyter_lsp took 0.0424s to import
[W 2024-01-16 21:32:38.034 ServerApp] A '_jupyter_server_extension_points' function was not found in jupyter_lsp. Instead, a '_jupyter_server_extension_paths' function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2024-01-16 21:32:38.060 ServerApp] Package jupyter_server_terminals took 0.0241s to import
[I 2024-01-16 21:32:38.061 ServerApp] Package jupyterlab took 0.0000s to import
[I 2024-01-16 21:32:38.569 ServerApp] Package notebook_shim took 0.0000s to import
[W 2024-01-16 21:32:38.569 ServerApp] A '_jupyter_server_extension_points' function was not found in notebook_shim. Instead, a '_jupyter_server_extension_paths' function was found and will be used for now. This function name will be deprecated in future releases of Jupyter Server.
[I 2024-01-16 21:32:38.571 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2024-01-16 21:32:38.576 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2024-01-16 21:32:38.583 ServerApp] jupyterlab | extension was successfully linked.
[I 2024-01-16 21:32:38.590 ServerApp] notebook | extension was successfully linked.
[I 2024-01-16 21:32:39.207 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-01-16 21:32:39.543 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-01-16 21:32:39.546 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-01-16 21:32:39.546 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-01-16 21:32:39.551 LabApp] JupyterLab extension loaded from C:\Users\ASUS\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11.qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\jupyterlab
[I 2024-01-16 21:32:39.551 LabApp] JupyterLab application is running at http://localhost:8888
```

Figure 13 Opening Jupyterlab from command prompt

1. Using the command prompt, type the Jupyter notebook. Jupyter notebook will open in desktop where our actual file is located. Then, open CustomerChurnPrediction.ipynb file which will direct us to the Jupyter IDE.

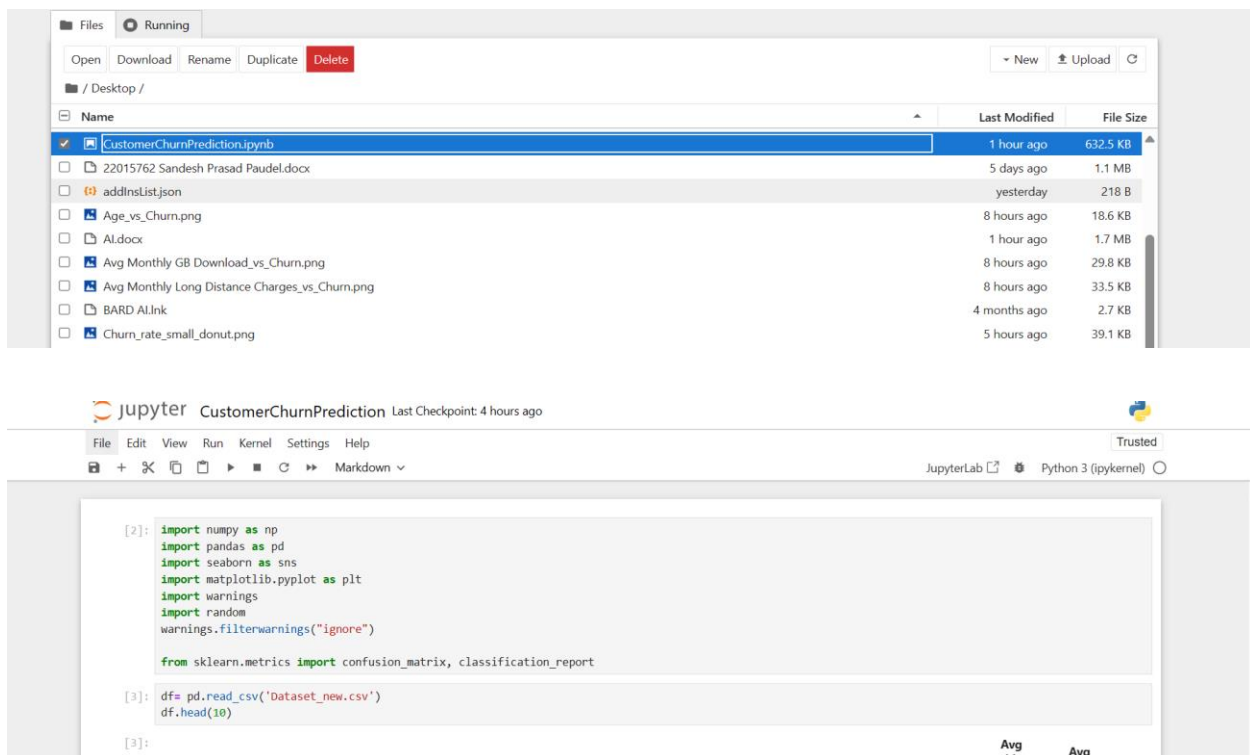


Figure 14 Opening Jupyter IDE from Web Browser

### 3.5.3.2 Data Preprocessing

1. Import the necessary libraries to be used in the project which are numpy, pandas, seaborn, and matplotlib.pyplot.

```
In [2]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
import random
warnings.filterwarnings("ignore")
```

Figure 15 Importing required libraries

2. After importing the datasets, read the datasets using pandas and place them in dataframe. Then, using head () function view top 10 data of the top

```
In [3]: df= pd.read_csv('Dataset_new.csv')
df.head(10)
```

Out[3]:

	customerID	Name	Gender	SeniorCitizen	Partner	Dependents	Tenure	PhoneService	MultipleLines	InternetService	...	Married	Avg Monthly Long Distance Charges	Avg Monthly GB Download	Streaming Music	Pr S
0	3668-QPYBK	Nabin Sapkota	Male	0	No	No	2	Yes	No	DSL	...	No	314.1	21	No	
1	9237-HQITU	Deepika Danwar	Female	0	No	No	2	Yes	No	Fiber optic	...	No	273.6	51	No	
2	9305-CDSKC	Swati Baral	Female	0	No	No	8	Yes	Yes	Fiber optic	...	No	364.5	26	Yes	
3	7892-POOKP	Kalpita Subba	Female	0	Yes	No	28	Yes	Yes	Fiber optic	...	Yes	146.7	47	Yes	
4	0280-XJGEX	Tulsi Duwadi	Male	0	No	No	49	Yes	Yes	Fiber optic	...	No	1329.9	11	Yes	
5	4190-MFLUW	Medhavi Magar	Female	0	Yes	Yes	10	Yes	No	DSL	...	Yes	282.3	69	No	
6	8779-QRDMV	Kanak Pokhrel	Male	1	No	No	1	No	No phone service	DSL	...	No	0.0	8	No	
7	1066-JKSGK	Hina Paneru	Male	0	No	No	1	Yes	No	No	...	No	1007.4	0	No	
8	6467-CHFZW	Dharma Sherpa	Male	0	Yes	Yes	47	Yes	Yes	Fiber optic	...	Yes	940.5	16	Yes	

Figure 16 Read Datasets

### 3. Check if the values in the datasets are null using is null function.

```
In [7]: df.isnull().sum()
```

```
Out[7]: Gender                0
SeniorCitizen                0
Partner                      0
Dependents                   0
Tenure                       11
PhoneService                 0
MultipleLines                0
InternetService              0
OnlineSecurity               0
OnlineBackup                 0
DeviceProtection             0
TechSupport                  0
StreamingTV                  0
StreamingMovies              0
Contract                     0
PaperlessBilling             0
PaymentMethod                0
MonthlyCharges               0
TotalCharges                 11
Churn                        0
Age                          0
Under_30                     0
Married                      0
Avg Monthly Long Distance Charges 0
Avg Monthly GB Download      0
Streaming Music               0
Premium Tech Support         0
Unlimited Data                0
Total Refunds                 0
Total Extra Data Charges     0
Total Long Distance Charges   0
Total Revenue                 0
```

Figure 17 Checking for null values in datasets

### 4. Drop the null values using dropna function.

Since, there are 11 null values in Total charges. As there are total of 7043 columns, dropping only 11 won't make huge impacts

```
n [8]: df.dropna(inplace=True)
```

After, removing null values let us check them back if null values are removed!

```
n [9]: df.isnull().sum()
```

Figure 18 dropping the null values

5. Select columns with an "object" data type, commonly used for categorical variables. The loop iterates through each of these categorical columns, creating a label for each that includes the column name and its unique values. This information is printed to the console, offering a concise overview of the distinct categorical values present in each relevant column.

```
In [11]: categorical_columns = df.select_dtypes(include="object")
```

Selecting each of the unique values from the categorical columns

```
In [306]: for column in categorical_columns.columns:
          label = f'{column} : {categorical_columns[column].unique()}'
          print(label)
```

```
Gender : ['Male' 'Female']
SeniorCitizen : ['No' 'Yes']
Partner : ['No' 'Yes']
Dependents : ['No' 'Yes']
PhoneService : ['Yes' 'No']
MultipleLines : ['No' 'Yes' 'No phone service']
InternetService : ['DSL' 'Fiber optic' 'No']
OnlineSecurity : ['Yes' 'No' 'No internet service']
OnlineBackup : ['Yes' 'No' 'No internet service']
DeviceProtection : ['No' 'Yes' 'No internet service']
TechSupport : ['No' 'Yes' 'No internet service']
StreamingTV : ['No' 'Yes' 'No internet service']
StreamingMovies : ['No' 'Yes' 'No internet service']
Contract : ['Month-to-month' 'Two year' 'One year']
PaperlessBilling : ['Yes' 'No']
PaymentMethod : ['eSewa' 'Cash' 'Mobile Banking' 'Credit card (automatic)']
Churn : ['Yes' 'No']
Under 30 : ['No' 'Yes']
Married : ['No' 'Yes']
Streaming Music : ['No' 'Yes']
Premium Tech Support : ['No' 'Yes']
Unlimited Data : ['Yes' 'No']
```

Since 'No internet service' and 'No phone service' are same as 'No'. Replace them with 'No'

```
In [307]: df.replace('No internet service', 'No', inplace=True)
          df.replace('No phone service', 'No', inplace=True)
```

Figure 19 Replacing similar data under ordinal 'No' in categorical columns

## 6. Create a donut chart to visualize the target variable.

### Target Variable check

```
plt.figure(figsize=(8, 8))
colors = ['#4169e1', '#ff0000']

plt.pie(df["Churn"].value_counts(), shadow=True, startangle=45,
        labels=df["Churn"].value_counts().index, autopct='%0.1f%%',
        explode=(0, 0.08), colors=colors)

centre_circle = plt.Circle((0, 0), 0.5, color='white', fc='white')
plt.gca().add_artist(centre_circle)

plt.title('Churn Rate - Small Donut Chart')
plt.savefig('Churn_rate_small_donut.png')
plt.show()
```

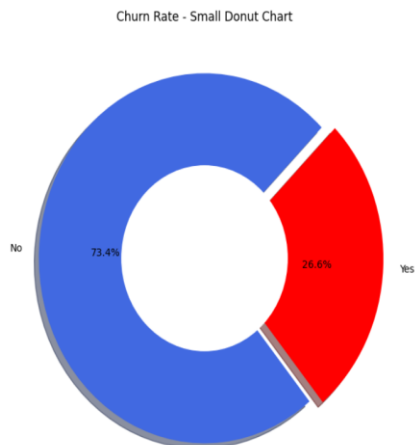


Figure 20 Visualize the overall target variable data

7. Outliers in a numerical columns are viewed using the Z-score method. For each numerical column, it calculates the Z-score for each value based on the column's mean and standard deviation. If a value's Z-score exceeds a threshold of 3, indicating it is an outlier, the value is replaced with np.nan.

```
def replace_outliers(value, mean, std):  
    z_score = np.abs(value - mean) / std  
    return value if z_score <= 3 else np.nan  
  
for column in numerical_columns:  
    mean = df[column].mean()  
    std = df[column].std()  
    df[column] = df[column].apply(replace_outliers, args=(mean, std))
```

Figure 21 Checking for outliers in the column

8. Subsequently, missing values in the DataFrame are filled with the mean of their respective columns. This approach aims to mitigate the impact of outliers on statistical measures and analyses.

### Replacing the outliers with null values by applying mean methods

```
df = df.apply(lambda column: column.fillna(column.mean()) if column.isnull().any() else column, axis=0)  
  
df.isnull().sum()
```

Figure 22 Replacing outliers by mean value of particular columns



### 3.5.3.3 Feature Selection

1. Generate a heatmap visualizing these correlation coefficients, where values closer to 1 indicate a stronger positive correlation, values closer to -1 indicate a stronger negative correlation, and values around 0 indicate a weaker or no correlation. The heatmap is annotated with the actual correlation values, and the plot is displayed with a color map ('PRGn') to represent positive and negative correlations.

```
numeric_df = df.select_dtypes(include=['number'])
corr = numeric_df.corr()

fig, ax = plt.subplots(figsize=(25, 15))
sns.heatmap(corr, vmax=1, vmin=-1, annot=True, cmap='PRGn', square=True, ax=ax)

plt.title('Correlation Matrix')
plt.show()
```

Figure 23 Code to generate heatmap



Figure 24 Heatmap to show correlated values

2. Drop the unnecessary numerical features which are not required during prediction using `df.drop` function with `inplace` as `true` for making permanent changes in the datasets as they have highly correlated values which is viewed from heatmap.

```
329... df.drop(['TotalCharges', 'Total Revenue', 'Under 30', 'Total Refunds',  
          'Total Extra Data Charges', 'Avg Monthly Long Distance Charges', 'Married', 'Total Long Distance Charges', 'Gender', 'SeniorCitizen'], axis=1, inplace=True)
```

Figure 25 Drop the unnecessary features

3. After selecting the features for prediction, a new dataset is saved in a desktop with a name `Modelling_Dataset.csv`.

#### Save the datasets

```
in [336... df.to_csv('Modelling_Dataset.csv', index=False)
```

Figure 26 Save modelling datasets into csv



Figure 27 Modelling Dataset in desktop

### 3.5.3.4 Data Modelling with pipeline building for final preprocessing

1. View the overall columns of the modelling datasets. Then, divide the target variable and independent variable.

```
[337...] df.columns
[337...] Index(['Partner', 'Dependents', 'Tenure', 'PhoneService', 'MultipleLines',
            'InternetService', 'StreamingServices', 'OnlineSecurity',
            'OnlineBackup', 'DeviceProtection', 'TechSupport', 'Contract',
            'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 'Age',
            'Avg Monthly GB Download', 'Premium Tech Support', 'Unlimited Data',
            'Churn'],
            dtype='object')
[338...] df.shape
[338...] (7032, 20)
X as dependent variable and Y as target variable.
[339...] X = df.drop('Churn', axis = 1)
y = df.Churn
```

Figure 28 Independent and Target Variable

2. Here, the independent variable which are categorical is divided into ordinal with 2 unique values in the columns or onehot with more than 2 unique values.

```
... is_ordinal, is_onehot = list(), list()
categorical_column = X.select_dtypes(include = 'object').columns.tolist()
numerical_column = X.select_dtypes(exclude = 'object').columns.tolist()
for col in categorical_column:
    unique_values = df[col].nunique()
    if unique_values == 2:
        is_ordinal.append(col)
    else:
        is_onehot.append(col)
```

Figure 29 Dividing categorical features into ordinal and one hot

```
In [341...] is_ordinal
Out[341...] ['Partner',
            'Dependents',
            'PhoneService',
            'MultipleLines',
            'OnlineSecurity',
            'OnlineBackup',
            'DeviceProtection',
            'TechSupport',
            'PaperlessBilling',
            'Premium Tech Support',
            'Unlimited Data']
In [342...] is_onehot
Out[342...] ['InternetService', 'Contract', 'PaymentMethod']
In [343...] numerical_column
Out[343...] ['Tenure',
            'StreamingServices',
            'MonthlyCharges',
            'Age',
            'Avg Monthly GB Download']
```

Figure 30 Checking the columns of ordinal and one hot

3. Three sub-pipelines within a larger Column Transformer is built. The numerical pipeline handles numerical features by imputing missing values with the mean and scaling the features to a specified range. The oneHotEncoding\_pipeline and ordinal\_pipeline handle categorical features, with the former performing one-hot encoding after imputing missing values with the most frequent value and the latter performing ordinal encoding using the same imputation strategy. The ColumnTransformer combines these pipelines and applies them to specific subsets of features based on their type (numerical, ordinal, or one-hot encoded). Finally, the overall dummy model pipeline fits the data through this preprocessing sequence.

```
In [346.. from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, OrdinalEncoder, MinMaxScaler
from sklearn.impute import SimpleImputer

In [347.. numerical_pipeline = Pipeline(steps = [
    ('imputer', SimpleImputer(missing_values=np.nan, strategy='mean')),
    ('scaler', MinMaxScaler(feature_range=(0, 1)))
])

In [348.. oneHotEncoding_pipeline = Pipeline(steps = [
    ('imputer', SimpleImputer(missing_values=None, strategy='most_frequent')),
    ('encoder', OneHotEncoder())
])

In [349.. ordinal_pipeline = Pipeline(steps = [
    ('imputer', SimpleImputer(missing_values=None, strategy='most_frequent')),
    ('encoder', OrdinalEncoder())
])

In [350.. transformColumn = ColumnTransformer(transformers = [
    ('numerical', numerical_pipeline, numerical_column),
    ('ordinal', ordinal_pipeline, is_ordinal),
    ('onehot', oneHotEncoding_pipeline, is_onehot)
])

In [351.. dummy_model = Pipeline(steps = [
    ('processor', transformColumn))
_ = dummy_model.fit(X, y)
```

4. After fitting the data with the dummy model pipeline, it retrieves the feature names from the one-hot encoding transformation and appends them to the list is\_onehot2. Additionally, it combines the original numerical columns (numerical column), ordinal columns (is ordinal), and the newly obtained one-hot encoded feature names into a single list named numerical\_columns\_final

Figure 31 Pipeline building for converting categorical columns into numerical one

```
is_onehot2 = list(dummy_model.named_steps['processor'].named_transformers_['onehot'].named_steps['encoder'].get_feature_names_out(input_features=is_onehot))

numerical_columns_final = list(numerical_column)
numerical_columns_final.extend(is_ordinal)
numerical_columns_final.extend(is_onehot2)
numerical_columns_final
```

Figure 32 Final pipeline extension in numerical columns

5. Firstly, all the required scikit-learn machine learning algorithms are import. Then, the transformed numerical columns are divided into train and test split with test size of 0.20 and random state of 23.

```
In [361]: from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier

from sklearn.model_selection import train_test_split, GridSearchCV, RandomizedSearchCV, cross_val_score
import time
from sklearn.metrics import accuracy_score, roc_curve, roc_auc_score
import sys
import os

In [362]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=23)

In [363]: X_train.shape

Out[363]: (5625, 26)

In [364]: X_test.shape

Out[364]: (1407, 26)
```

Figure 33 Train Test Split

6. Train the five models and fit the models in training datasets.

```
lr = LogisticRegression()
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)
results, false = display_test_scores(y_test, y_pred)
```

Figure 34 Logistic Regression

```
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
results, false = display_test_scores(y_test, y_pred)
```

Figure 35 RandomForestClassifier

```
dt_classifier = DecisionTreeClassifier(random_state=0, criterion="entropy")
dt_classifier.fit(X_train, y_train)
y_pred = dt_classifier.predict(X_test)
results, false = display_test_scores(y_test, y_pred)
```

Figure 36 DecisionTreeClassifier

```
nb = GaussianNB()
nb.fit(X_train, y_train)
y_pred = nb.predict(X_test)
results, false = display_test_scores(y_test, y_pred)
```

Figure 37 NaiveBayes

```
svm = SVC()
svm.fit(X_train, y_train)
y_pred = svm.predict(X_test)
results, false = display_test_scores(y_test, y_pred)
```

Figure 38 Support Vector Machine

### 3.5.4 Achieved Results

#### 3.5.4.1 Logistic Regression results

```

-----
ACCURACY: 0.7967

AUC: 0.7150

CONFUSION MATRIX:
-----
[[922 120]
 [166 199]]

-----

```

	precision	recall	f1-score	support
0	0.85	0.88	0.87	1042
1	0.62	0.55	0.58	365
accuracy			0.80	1407
macro avg	0.74	0.72	0.72	1407
weighted avg	0.79	0.80	0.79	1407

Figure 39 Logistic Regression classification reports before tuning

```

param_grid2 = {
    'C': [ 0.001, 0.01, 0.1, 1, 10, 100, 1000],
    'penalty': ['l1', 'l2'],
    'solver': [ 'lbfgs', 'liblinear']
}

start_time = time.time()
grid_search2 = GridSearchCV(lr, param_grid2, verbose=1,cv=5, scoring='accuracy')

```

Figure 40 Using GridSearchCV for tuning logistic regression

```

Fitting 5 folds for each of 28 candidates, totalling 140 fits
The best parameters are {'C': 1, 'penalty': 'l1', 'solver': 'liblinear'}

Run time for train & test cv (Logistic Regression): 3.4088404178619385

##### TEST SCORES #####
-----
ACCURACY: 0.8010

AUC: 0.7188

CONFUSION MATRIX:
-----
[[927 115]
 [165 200]]

-----

```

	precision	recall	f1-score	support
0	0.85	0.89	0.87	1042
1	0.63	0.55	0.59	365
accuracy			0.80	1407
macro avg	0.74	0.72	0.73	1407
weighted avg	0.79	0.80	0.80	1407

Figure 41 Logistic Regression classification reports after tuning with above parameters

### 3.5.4.2 Random Forest Classifier

```

-----
ACCURACY: 0.7946

AUC: 0.7020

CONFUSION MATRIX:
-----
[[932 110]
 [179 186]]

-----

```

	precision	recall	f1-score	support
0	0.84	0.89	0.87	1042
1	0.63	0.51	0.56	365
accuracy			0.79	1407
macro avg	0.73	0.70	0.71	1407
weighted avg	0.78	0.79	0.79	1407

Figure 42 Random Forest classification reports before tuning

```

param_dist2 = {
    'n_estimators': [140, 160, 180, 200, 220, 240],
    'max_features': [0.6, 0.8, 1.0],
    'max_depth': [6, 8, 10, 12, 14],
    'max_samples': [0.7, 0.8, 0.9, 1.0]
}

start_time = time.time()
rf_random2 = RandomizedSearchCV(rf, param_distributions=param_dist2, n_iter=50, cv=5, verbose=2, n_jobs=-1, random_state=42)

rf_random2.fit(X_train, y_train)
end_time = time.time()

```

Figure 43 Using RandomizedSearchCV for tuning random forest classifier

```

Fitting 5 folds for each of 50 candidates, totalling 250 fits
The best parameters are {'n_estimators': 180, 'max_samples': 0.8, 'max_features': 1.0, 'max_depth': 6}

Run time for train & test cv (Random Forest Regression): 101.86899423599243

##### TEST SCORES #####
-----
ACCURACY: 0.8017

AUC: 0.7121

CONFUSION MATRIX:
-----
[[936 106]
 [173 192]]

-----

```

	precision	recall	f1-score	support
0	0.84	0.90	0.87	1042
1	0.64	0.53	0.58	365
accuracy			0.80	1407
macro avg	0.74	0.71	0.72	1407
weighted avg	0.79	0.80	0.79	1407

Figure 44 Random Forest classification reports after tuning with above parameters

### 3.5.4.3 Decision Tree Classifier

ACCURACY: 0.7363

AUC: 0.6680

CONFUSION MATRIX:

```
-----
[[844 198]
 [173 192]]
```

```
-----
              precision    recall  f1-score   support

         0       0.83        0.81        0.82       1042
         1       0.49        0.53        0.51        365

   accuracy                0.74       1407
  macro avg              0.66        0.67        0.66       1407
 weighted avg              0.74        0.74        0.74       1407
```

Figure 45 Decision Tree Classification Reports before Tuning

```
: params = {
    'max_depth': [2, 3, 5, 10, 20, 40],
    'min_samples_leaf': [5, 10, 20, 50, 100],
    'criterion': ["gini", "entropy"]
}

: start_time = time.time()
  grid_dt2 = GridSearchCV(dt_classifier, param_grid=params, verbose=1, cv=5, n_jobs=-1)

: grid_dt2.fit(X_train, y_train)
  end_time = time.time()
```

Figure 46 Using GridSearchCV for tuning decision tree classifier

```
Fitting 5 folds for each of 60 candidates, totalling 300 fits
The best parameters are {'criterion': 'entropy', 'max_depth': 10, 'min_samples_leaf': 100}

Run time for train & test cv (Decision Tree): 0.9868083000183105

#### TEST SCORES ####
-----
ACCURACY: 0.7953

AUC: 0.6954

CONFUSION MATRIX:
-----
[[941 101]
 [187 178]]

-----
              precision    recall  f1-score   support

         0       0.83        0.90        0.87       1042
         1       0.64        0.49        0.55        365

   accuracy                0.80       1407
  macro avg              0.74        0.70        0.71       1407
 weighted avg              0.78        0.80        0.79       1407
```

Figure 47 Decision Tree Classification Reports after Tuning



### 3.5.4.4 Naïve Bayes Classifier

```

-----
ACCURACY: 0.7520

AUC: 0.7604

CONFUSION MATRIX:
-----
[[774 268]
 [ 81 284]]

-----

```

	precision	recall	f1-score	support
0	0.91	0.74	0.82	1042
1	0.51	0.78	0.62	365
accuracy			0.75	1407
macro avg	0.71	0.76	0.72	1407
weighted avg	0.80	0.75	0.77	1407

Figure 48 Naïve Bayes Classification Reports before Tuning

```

: params_NB2 = {'var_smoothing': np.logspace(0,-9, num=10)}

: start_time = time.time()
: gnb_grid2 = GridSearchCV(nb, params_NB2, cv=5, verbose=1, n_jobs=-1)

: gnb_grid2.fit(X_test, y_test)

```

Figure 49 Using GridSearchCV for tuning Naïve Bayes

```

Fitting 5 folds for each of 10 candidates, totalling 50 fits
The best parameters are {'var_smoothing': 0.001}

Run time for train & test cv (Naive Bayes using Gaussian NB): -0.17151308059692383

##### TEST SCORES #####
-----
ACCURACY: 0.7555

AUC: 0.7628

CONFUSION MATRIX:
-----
[[779 263]
 [ 81 284]]

-----

```

	precision	recall	f1-score	support
0	0.91	0.75	0.82	1042
1	0.52	0.78	0.62	365
accuracy			0.76	1407
macro avg	0.71	0.76	0.72	1407
weighted avg	0.81	0.76	0.77	1407

Figure 50 Naïve Bayes Classification Reports after Tuning

### 3.5.4.5 Support Vector Machine (SVM)

```

-----
ACCURACY: 0.7967

AUC: 0.7008

CONFUSION MATRIX:
-----
[[938 104]
 [182 183]]
-----

              precision    recall  f1-score   support

     0         0.84        0.90        0.87       1042
     1         0.64        0.50        0.56        365

 accuracy          0.80          0.80       1407
 macro avg         0.74        0.70        0.71       1407
 weighted avg      0.79        0.80        0.79       1407

```

Figure 51 SVM Classification Reports before Tuning

```

parameters = {
    "C": [0.001, 0.01, 0.1],
    "kernel": ["linear", "poly", "rbf", "sigmoid"],
    "gamma": ["scale", "auto"],
}

start_time=time.time()
grid_6 = GridSearchCV(svm, param_grid=parameters, verbose=1, cv=5, n_jobs=-1)

grid_6.fit(X_train, y_train)

```

Figure 52 Using GridSearchCV for tuning SVM

```

Fitting 5 folds for each of 24 candidates, totalling 120 fits
The best parameters are {'C': 0.1, 'gamma': 'scale', 'kernel': 'linear'}

Run time for train&test cv SVM : 21.012845754623413

##### TEST SCORES #####
-----
ACCURACY: 0.7982

AUC: 0.7124

CONFUSION MATRIX:
-----
[[928 114]
 [170 195]]
-----

              precision    recall  f1-score   support

     0         0.85        0.89        0.87       1042
     1         0.63        0.53        0.58        365

 accuracy          0.80          0.80       1407
 macro avg         0.74        0.71        0.72       1407
 weighted avg      0.79        0.80        0.79       1407

```

Figure 53 SVM Classification Reports after Tuning

### 3.5.4.6 Boxplot to show the accuracy of used models

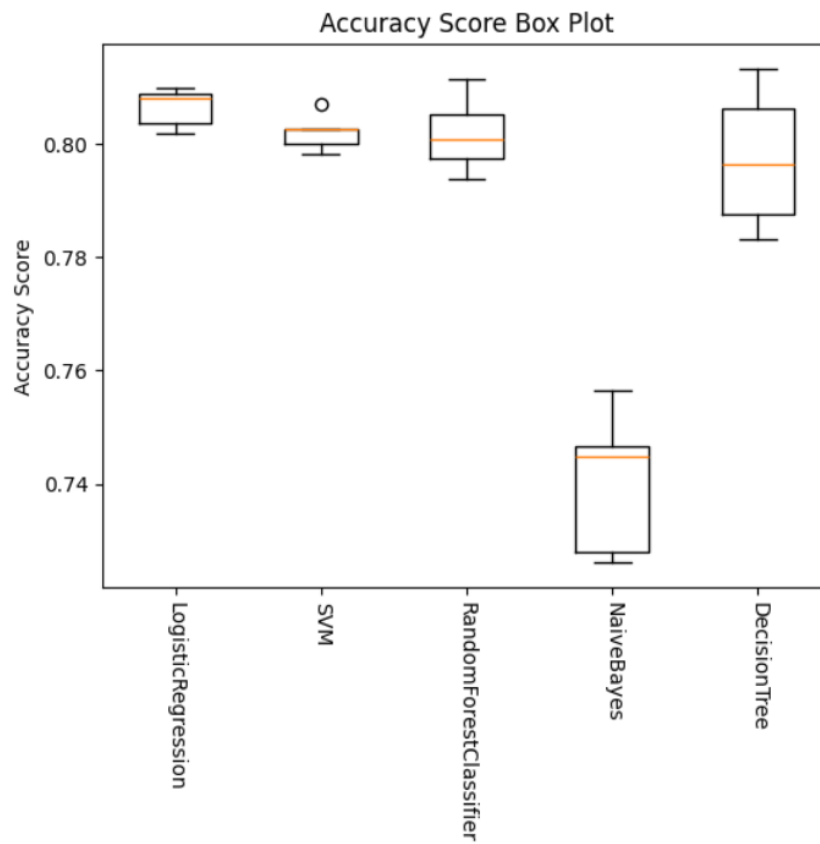


Figure 54 Boxplot for model accuracy

### 3.5.4.7 ROC curve of used models

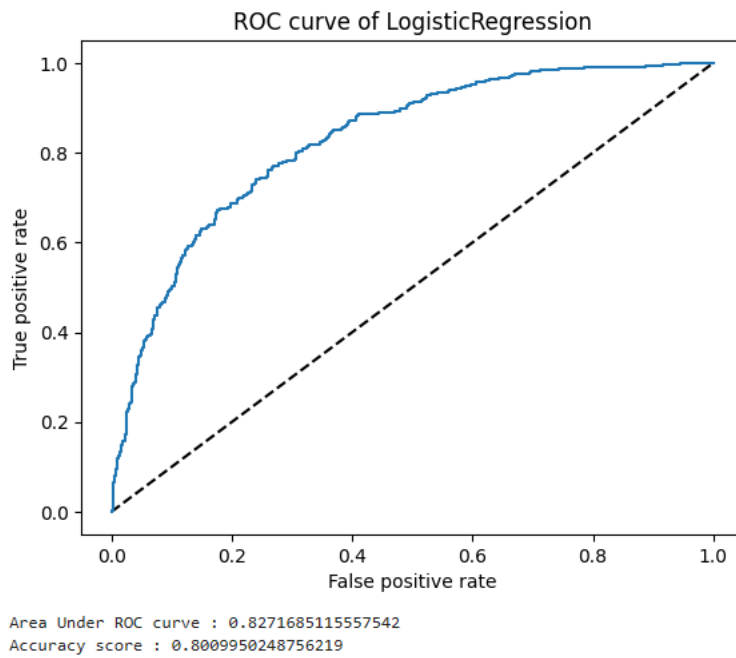


Figure 55 Logistic Regression ROC Curve

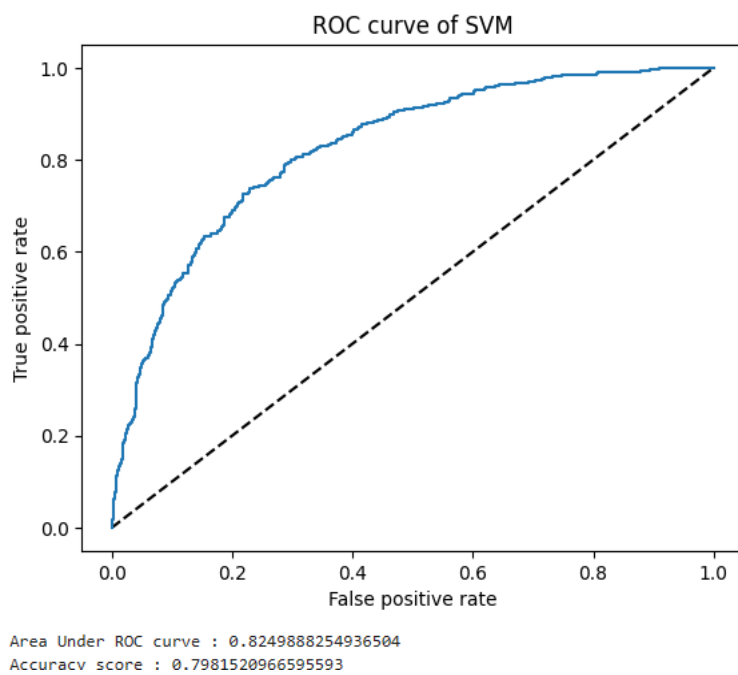


Figure 56 SVM ROC Curve

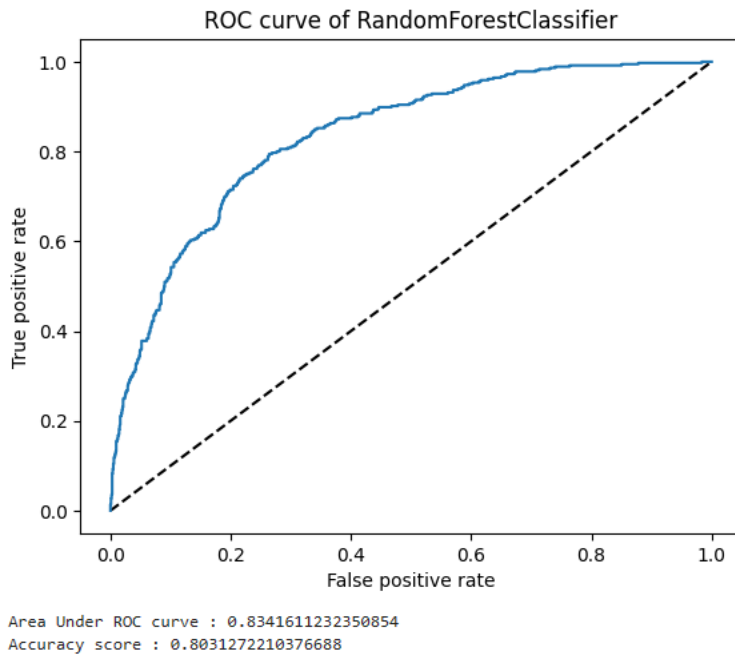


Figure 57 RandomForest ROC Curve

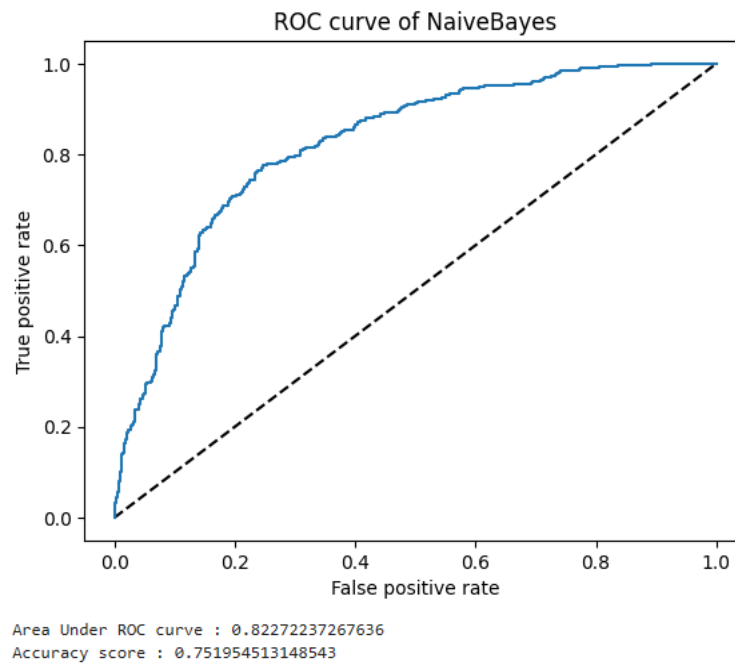


Figure 58 NaiveBayes ROC Curve

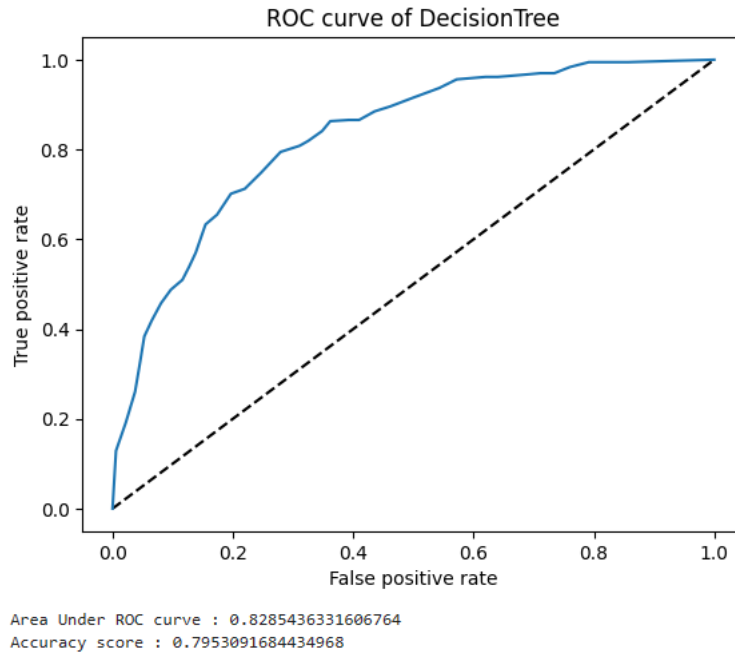


Figure 59 Decision Tree ROC Curve

### 3.5.4.8 Final Overview of the model performance

For customer churn prediction with a new datasets of modelling, I have used altogether of 5 models where the models are tuned either by using GridSearchCV or RandomizedSearchCV. On tuning I have found that Logistic regression, and RandomForest Classifier to have accuracy of 80%. Among them, if we compare other metrics like f1-score, recall and precision Logistic Regression still withstands. While SVM and Decision Tree are likely to have 79% accuracy with Naïve Bayes as the lowest with 75%. Overall, it suggests that Logistic Regression performs the best in terms of accuracy, while Naïve Bayes performs the worst. However, it is important to note that accuracy is just one metric for evaluating classification models, and other factors are also important considerations. If we view the AUC Score Random Forest Classifier stands as the best.

## **4. Conclusion**

### **4.1 Analysis of Work Done**

A clear introduction to AI and ML along with its types is done for the readers who are unfamiliar about the concepts. Since, my prediction is of telecom customer churn, a brief description of it is also presented. Then, the problem domain for choosing this proposed topic is listed out. After this much of introduction, I moved into the background and research for my problem domain or topic. Here, the research on existing work of any 5 research papers are done on telecom customer churn prediction. During this, I mentioned the title, year, journal name, volume, pages and issues of different research papers. These reviews or paper helped in identifying commonly used algorithms such as Logistic Regression, Decision Tree, Random Forest, Naïve Bayes and SVM with their accuracies and other evaluation metrics.

Along with these all, insights of the challenges faced by the industry and the significance of retaining customers using AI/ML is also provided. Inclusion of information about the dataset, its source and the relevant features are also mentioned. Under the proposed solution the details of the algorithms are explained. Then, the pseudocode showing the overall overview is shown along with flowchart. After all of this, the development process is carried out where models for customer churn prediction are build, tested and the performance is visualized using boxplot and ROC Curve.

## 4.2 How solution addresses real-world problems?

In the telecom sector, specific retention plans promote diverse approaches to keep current customers. Telecom companies are able to provide customer-oriented offerings, discounts, and promotions by analyzing customer data and their preferences. These strategies look for enhancing each customer's lifetime value, promote retention of customers, and lower churn rates. If we identify earlier about the customers who might be at risk enables immediate response and elimination of issues by improving the overall needs of customers. Using data driven decision-making encourages companies to make decisions on the best way to retain customers for particular customers groups. The relationship between the telecom provider and its customers continues to strengthen by special deals, improved customer experiences, and possibilities for cross-selling. Customers who are satisfied are not only more likely to remain with you but also will pass on around information regarding their positive experiences. In this way, if we use different ML and AI algorithms to derive data for decisions, it solves the churn problems of overall telecom companies.

## 4.3 Limitations of the system

The project's early phases had been affected by issues that emerged throughout the encoding of categorical data during the development process. Furthermore, the process of tuning hyperparameter was difficult, especially when building complex trees for the Random Forest and Decision Tree models, which resulted in a series of error logs. One major limitation had been the existence of class imbalance in customer churn datasets, where it was significant variance between churners and non-churners, which could introduce bias into the predictive models. In order to improve model accuracy and fairness, this limitation emphasizes the need for further research on datasets with a more balanced representation of churners and non-churners. Additional ensemble techniques and neural network algorithms, like gradient boosting, Artificial Neural Networks (ANN), and LightGBM, can help the system get better in future versions.

An in-depth evaluation of feature importance to improve model understanding and resolving any overfitting addresses are two more areas that could be improved. Furthermore, studying methods such as advanced regularization techniques and data augmentation may help improve generalization. The ultimate objective is to provide predictive insights via an online application so that businesses or individuals can evaluate and understand churn rates with ease.



## Bibliography

Altex Soft, 2022. *Semi-supervised learning*. [Online]

Available at: <https://www.altexsoft.com/blog/semi-supervised-learning/>

[Accessed 17 12 2023].

Analytics Vidhya, 2023. *Matplot and Seaborn*. [Online]

Available at: <https://www.analyticsvidhya.com/blog/2021/10/introduction-to-matplotlib-using-python-for-beginners/>

[Accessed 15 01 2024].

Basnet, S. S. a. D., 2022. Determinant of Customer Churn in the Nepalese Mobile Telephony Market. *Management Dynamics*,, 25(2), pp. 57-73.

Bhatt, S., 2018. *Reinforcement Learning 101*. [Online]

Available at: <https://towardsdatascience.com/reinforcement-learning-101-e24b50e1d292>

[Accessed 16 12 2023].

Chauhan, N. S., 2022. *decision tree algorithm explained*. [Online]

Available at: <https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html>

[Accessed 18 12 2023].

Chugh, V., 2023. *pandas introduction*. [Online]

Available at: <https://www.datacamp.com/tutorial/pandas>

[Accessed 15 01 2024].

Dash, S., 2022. *decision trees ecplained*. [Online]

Available at: <https://towardsdatascience.com/decision-trees-explained-entropy-information-gain-gini-index-ccp-pruning-4d78070db36c>

[Accessed 18 12 2023].

Data Flair, 2022. *svm-kernel-functions*. [Online]

Available at: <https://data-flair.training/blogs/svm-kernel-functions>

[Accessed 18 12 2023].

Ellis, D. R., 2023. *hubspot.com*. [Online]

Available at: <https://blog.hubspot.com/website/anaconda-python>

[Accessed 15 01 2024].

Fayrix, 2022. *Benefits of Customer Churn Prediction Using Machine Learning*. [Online]

Available at: <https://fayrix.com/blog/customer-churn-prediction-benefits>

[Accessed 16 12 2023].

Geeks for Geeks, 2023. *machine learning vs artificial intelligence*. [Online]

Available at: <https://www.geeksforgeeks.org/machine-learning-versus-artificial-intelligence/>

[Accessed 16 12 2023].

IBM, 2022. *Naive Bayes*. [Online]

Available at: <https://www.ibm.com/topics/naive-bayes>

[Accessed 18 12 2023].

IBM, 2023. *machine-learning*. [Online]

Available at: <https://www.ibm.com/topics/machine-learning>

[Accessed 16 12 2023].

Kanade, V., 2022. *What is Logistic Regression*. [Online]

Available at: <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/>

[Accessed 18 12 2023].

Karamollaoğlu, H., Yücedağ, İ. & Doğru, İ. A., 2021. Customer Churn Prediction Using Machine Learning Methods: A Comparative Analysis.

Khan, R., 2022. *datasets in machine learning*. [Online]

Available at: <https://www.datatobiz.com/blog/datasets-in-machine-learning/>

[Accessed 18 12 2023].

Kiran Dhangar, P. A., 2021. Review on Customer Churn Prediction Using Machine Learning Techniques. *International Journal of Innovations in Engineering Research and Technology*, 8(5), pp. 193-201.

LeewayHertz, 2023. *ai and ml in customer churn prediction*. [Online]

Available at: <https://www.leewayhertz.com/ai-and-ml-in-customer-churn-prediction/>

[Accessed 17 12 2023].

Malika Naresh Panchal, D. A. A. P., 2020. Churn Prediction using Supervised Machine Learning Algorithms- Impact of Over Sampling. *International Research Journal of Engineering and Technology (IRJET)*, 7(11), pp. 1014-1019.

Md. Ashraful Haque, M. S. M. A. A. S. M. R.-U.-Z. M. A., December 2023. Customer churn prediction in telecom sector using machine learning techniques. *Engineering Applications of Artificial Intelligence*, 152(December 2023), pp. 107545-107550.

NumPy official website, 2021. *What is NumPy*. [Online]

Available at: <https://numpy.org/doc/stable/user/whatisnumpy.html>

[Accessed 15 01 2024].

Official website of python, 2021. *Python*. [Online]

Available at: <https://www.python.org/doc/essays/blurb/>

[Accessed 15 01 2024].

Parliament, N. E., 2023. *what is artificial intelligence and how it is used?*. [Online]

Available at: <https://www.europarl.europa.eu/news/en/headlines/society/20200827STO85804/what-is-artificial-intelligence-and-how-is-it-used>

[Accessed 16 12 2023].

Ray, S., 2023. *understaing-support-vector-machine-example-code*. [Online]

Available at: [https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/#What\\_Is\\_a\\_Support\\_Vector\\_Machine\\_\(SVM\)?](https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/#What_Is_a_Support_Vector_Machine_(SVM)?)

[Accessed 17 12 2023].

R, S. E., 2023. *Understanding Random Forest*. [Online]

Available at: <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>

[Accessed 18 12 2023].

Saran Kumar A., C. D., 2023. A Survey on Customer Churn Prediction using Machine Learning Techniques. *International Journal of Computer Applications*, 154(10).

Sethi, A., 14th April 2020. *supervised learning vs unsupervised learning*. [Online]

Available at: <https://www.analyticsvidhya.com/blog/2020/04/supervised-learning-unsupervised-learning/>  
[Accessed 16 12 2023].

Sharma, M., 2023. *Scikit Learn*. [Online]

Available at: <https://intellipaat.com/blog/tutorial/python-tutorial/scikit-learn-tutorial/>  
[Accessed 15 01 2024].

Tan Yi Fei, L. H. S. L. J. Y. G. X. S. W. K., 2017. Prediction on Customer Churn in the Telecommunications Sector Using Discretization and Naïve Bayes Classifier. *International Journal of Advances in Soft Computing and its Applications (IJSTAC)*, 09(03), pp. 24-35.

Taskin, N., 2023. Customer churn prediction. *Customer churn prediction model in telecommunication industry*, Volume I, p. 30.

Tianpei Xu, Y. M. a. K. K., 2021. Churn Predictions. *Telecom Churn Prediction System Based on Ensemble Learning Using Feature Grouping*, Volume I, p. 12.

V.Kavitha, G. K. S. K. M., 2020. Churn Prediction of Customer in Telecom Industry using Machine Learning Algorithms. *International Journal of Engineering Research and Technology*, 09(05), pp. 181-184.

Vibhor Shah, D. H. D., 2021. Churn Analysis in Telecommunication Industry using Machine Learning Techniques. *Annals of R.S.C.B*, 25(05), pp. 4321- 4326.

Wiryaseputra, M., 2022. *Bank Customer churn prediction*. [Online]

Available at: <https://www.analyticsvidhya.com/blog/2022/09/bank-customer-churn-prediction-using-machine-learning/>  
[Accessed 16 12 2023].

## Appendix

### A. Other Algorithms

#### 1. Random Forest

Random Forest Algorithm widespread popularity stems from its user-friendly nature and adaptability, enabling it to tackle both classification and regression problems effectively. The algorithm's strength lies in its ability to handle complex datasets and mitigate overfitting, making it a valuable tool for various predictive tasks in machine learning. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables, as in the case of regression, and categorical variables, as in the case of classification. It performs better for classification and regression tasks (R, 2023).

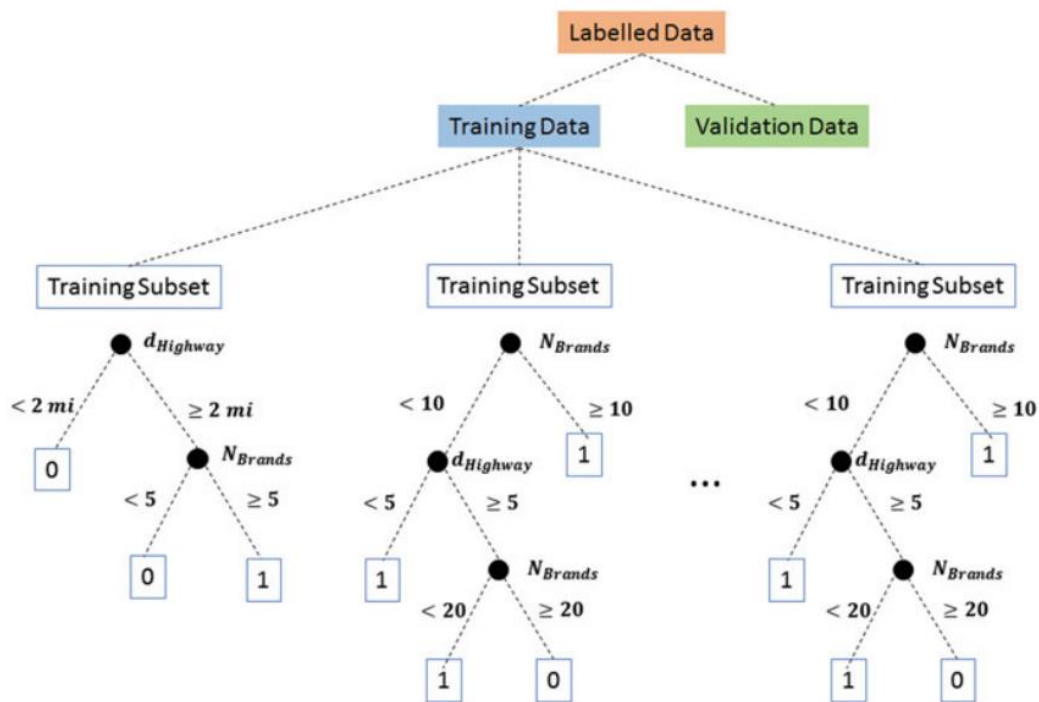


Figure 60 Random Forest Classification Problem

Random forest is under the ensemble learning classifier since it combines decision tree and gives an output.

## 2 Naïve Bayes

Naïve Bayes classifier is a supervised machine learning algorithm, which is used for classification tasks, like text classification. It is also part of a family of generative learning algorithms, meaning that it seeks to model the distribution of inputs of a given class or category. Unlike discriminative classifiers, like logistic regression, it does not learn which features are most important to differentiate between classes (IBM, 2022).

### 2.1 Bayes Theorem

Bayes' Theorem states that the conditional probability of an event, based on the occurrence of another event, is equal to the likelihood of the second event given the first event multiplied by the probability of the first event.

The diagram shows the formula  $P(A|B) = \frac{P(B|A) P(A)}{P(B)}$  with handwritten annotations and arrows explaining each term:

- $P(A|B)$ : THE PROBABILITY OF "A" BEING TRUE GIVEN THAT "B" IS TRUE
- $P(B|A)$ : THE PROBABILITY OF "B" BEING TRUE GIVEN THAT "A" IS TRUE
- $P(A)$ : THE PROBABILITY OF "A" BEING TRUE
- $P(B)$ : THE PROBABILITY OF "B" BEING TRUE

Where,

- $P(A|B)$  is the probability of hypothesis A given the data B. This is called the posterior probability.
- $P(B|A)$  is the probability of data B given that the hypothesis A was true.
- $P(A)$  is the probability of hypothesis A being true (regardless of the data). This is called the prior probability of A.
- $P(B)$  is the probability of the data (regardless of the hypothesis).
- $P(A|B)$  or  $P(B|A)$  are conditional probabilities  $P(B|A) = P(A \text{ and } B)/P(A)$

Figure 61 Probability Formula for Bayes Theorem

Three types of Naïve Bayes Classifier are Multinomial, Bernoulli, and Gaussian Classifier.

## B. Evaluation Metrics

### 1. Accuracy

Accuracy simply measures how often the classifier correctly predicts. We can define accuracy as the ratio of the number of correct predictions and the total number of predictions.

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

Figure 62 Accuracy Formula

### 2. Precision

It explains how many of the correctly predicted cases actually turned out to be positive. Precision is useful in the cases where False Positive is a higher concern than False Negatives.

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}$$

Figure 63 Precision Formula

### 3. Recall (Sensitivity)

It explains how many of the actual positive cases we were able to predict correctly with our model. Recall is a useful metric in cases where False Negative is of higher concern than False Positive.

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}}$$

Figure 64 Recall Formula

#### 4. F1 Score

It gives a combined idea about precision and recall metrics. It is maximum when precision is equal to recall.

$$F1 = 2. \frac{Precision \times Recall}{Precision + Recall}$$

Figure 65 F1-Score Formula

Note: Back to other algorithms [Support Vector Machine \(SVM\)](#).



## C. Others Tools Used

### 3.5.1.2 Jupyter Notebook



Figure 66 Jupyter Notebook Logo

Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. It is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself.

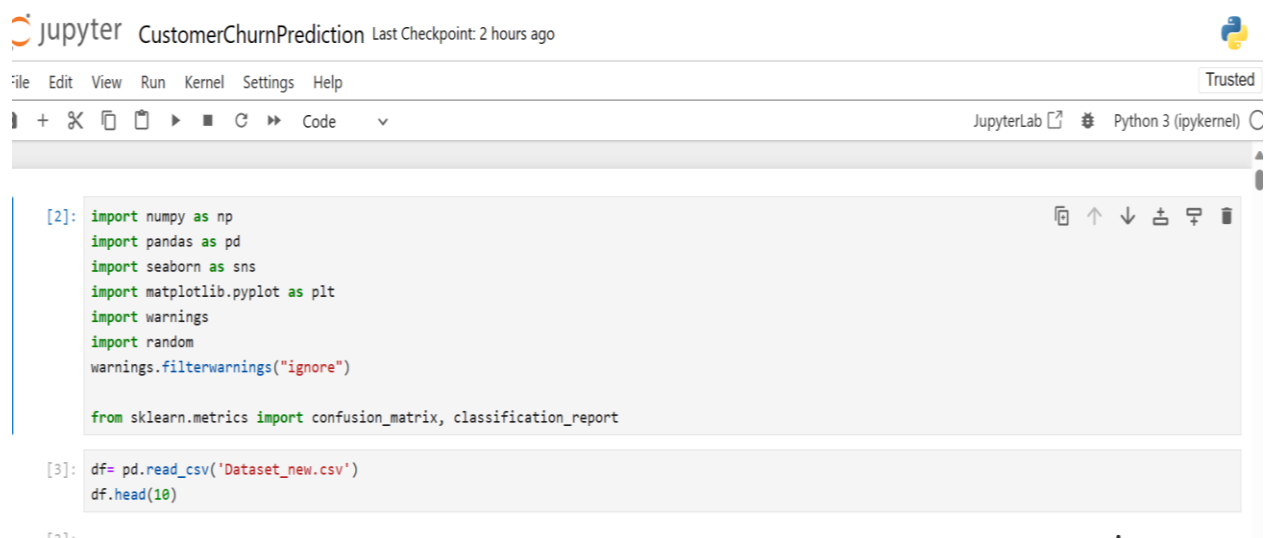


Figure 67 Jupyter Notebook Interface

### 3.5.1.3 Python 3



Figure 68 Logo of Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed (Official website of python, 2021).

The image shows a JupyterLab interface with a file named 'CustomerChurnPrediction'. The code in the cell is as follows:

```
[1]: def display_test_scores(test, pred):
    str_out = "\n"
    str_out += ("##### TEST SCORES #####\n-----")
    str_out += ("\n")

    #print accuracy
    accuracy = accuracy_score(test, pred)
    str_out += ("ACCURACY: {:.4f}\n".format(accuracy))
    str_out += ("\n")

    #print AUC score
    auc = roc_auc_score(test, pred)
    str_out += ("AUC: {:.4f}\n".format(auc))
    str_out += ("\n")

    #print confusion matrix
    str_out += ("CONFUSION MATRIX:\n-----\n")
    conf_mat = confusion_matrix(test, pred)
    str_out += ("{}\n".format(conf_mat))
    str_out += ("\n")
    str_out += ("\n-----\n")
```

Figure 69 Use of Python code in Jupyter IDE

Note: Go Back to Top. [3.5.1 Tools Used](#)

## D. Others Libraries Used:

### 3.5.2.2 NumPy

NumPy is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical algebra, basic statistical operations, random simulation and much more. At the core of the NumPy package, is the ndarray object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance (NumPy official website, 2021).

```
In [48]: import numpy as np

In [49]: a = np.array([[11,12,13],[21,22,23]])

print(a)

-----
TypeError                                Traceback (most recent call last)
<ipython-input-49-65fa3f3609a0> in <module>()
----> 1 a = np.array([[11,12,13],[21,22,23]])
      2
      3 print(a)

TypeError: 'tuple' object is not callable
```

Figure 70 Import and use of NumPy array

### 3.5.2.3 Matplotlib and Seaborn

Matplotlib and seaborn are popular plotting library in Python used for creating high-quality visualizations and graphs. They offer various tools to generate diverse plots, facilitating data analysis, exploration, and presentation. They are flexible, supporting multiple plot types and customization options, making it valuable for scientific research, data analysis, and visual communication. Different types of visualization reports like line plots, scatter plots, histograms, bar charts, pie charts, box plots, and many more different plots can be created using them (Analytics Vidhya, 2023).

```
[2]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Figure 71 Importing seaborn and matplotlib

### 3.5.2.4 Scikit-learn

Scikit-learn is a free machine-learning library for Python. It's a very useful tool for data mining and analysis and can be used for personal as well as commercial purposes. Python Scikit-learn lets users perform various machine learning tasks and provides a means to implement machine learning in Python. It needs to work with Python scientific and numerical libraries, namely, Python SciPy and Python NumPy, respectively. It's basically a SciPy toolkit that features various machine learning algorithms (Sharma, 2023).

```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier

from sklearn.model_selection import train_test_split, GridSearchCV, RandomizedSearchCV, cross_val_score
import time
from sklearn.metrics import accuracy_score, roc_curve, roc_auc_score
import sys
```

Figure 72 Use of Scikit Learn for Machine Learning

Note: Return back to top. [3.5.2 Libraries Used](#)