**CHIKKANNA GOVERNMENT ARTS COLLEGE**

**DEPARTMENT OF BACHELOR OF COMPUTER APPLICATION**

**TIRUPUR-641602**

**(AFFILIATED TO BHARATHIAR UNIVERSITY)**



**TEAM MEMBERS NAME :**

**MANIKANDAN N (2022J0041)**

**KALYANASUNDHARAM M (2022J0035)**

**ABINESHKUMAR M(2022J0010)**

**ILANKUMARAN T(2022J0766)**

## **Intelligent Customer Retention:Using Machine Learning for Enhanced Prediction of Telecom Customer Churn**

**1.INTRODUCTION**

**1.1 OVERVIEW :**

Customer churn is often referred to as customer attrition, or customer defection which is the rate at which the customers are lost. Customer churn is a major problem and one of the most important concerns for large companies. Due to the direct effect on the revenues of the companies, especially in the telecom field, companies are seeking to develop means to predict potential customer to churn.Looking at churn, different reasons trigger customers to terminate their contracts, for example better price offers, more interesting packages, bad service experiences or change of customers’ personal situations.

Customer churn has become highly important for companies because ofincreasing competition among companies, increased importance of marketing strategies and consciousbehaviour of customers in the recent years. Customers can easily trend toward alternative services. Companies must develop various strategies to prevent these possible trends, depending on the services they provide. During the estimation of possible churns, data from the previous churns might be used. An efficient churn predictive model benefits companies in many ways. Early identification of customers likely to leave may help to build cost effective ways in marketing strategies.

**1.2. PURPOSE :**

Intelligent customer retention using machine learning for enhanced prediction of telecom customer churn is to help telecom companies to identify and retain customers who are likely to churn (i.e., cancel their subscription or switch to a competitor) before they actually do so. By using machine learning algorithms to analyze customer data, such as call logs, usage patterns, demographic information, and customer feedback, telecom companies can gain insights into the factors that influence customer churn.

With this knowledge, they can develop targeted retention strategies, such as personalized offers, improved customer service, and proactive outreach, to reduce the likelihood of customer churn. By using machine learning to predict customer churn, telecom companies can also prioritize their retention efforts and allocate resources more effectively.

Intelligent customer retention using machine learning can help telecom companies to achieve several benefits, including:

Improved customer satisfaction: By addressing the factors that drive customer churn, telecom companies can improve the overall customer experience, which can lead to higher customer satisfaction and loyalty.

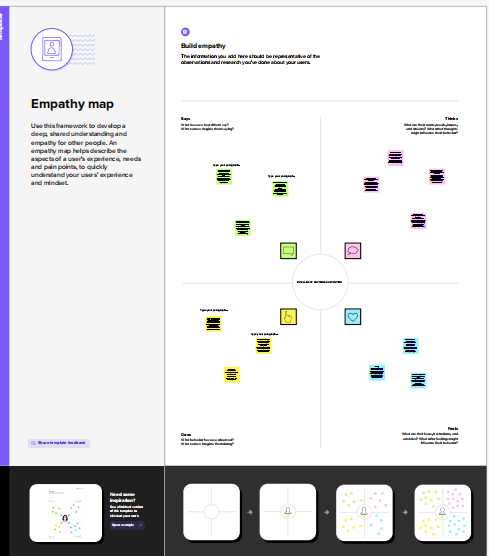
Increased revenue: Retaining existing customers is typically less costly than acquiring new ones. By reducing customer churn, telecom companies can increase revenue and profitability.

Competitive advantage: Telecom companies that can predict and prevent customer churn have a competitive advantage over those that cannot, as they can offer better services and retain more customers over the long term.

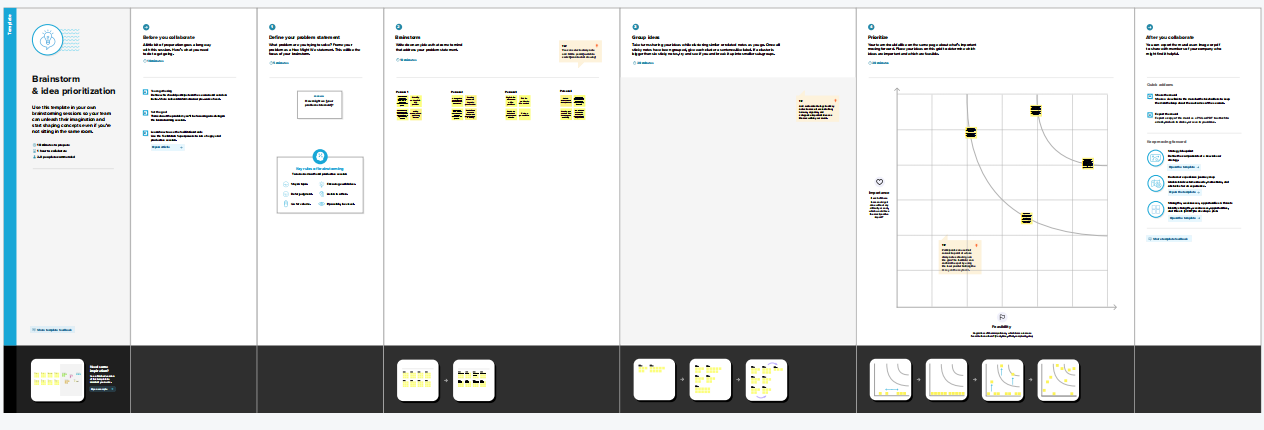
Overall, intelligent customer retention using machine learning is an important strategy for telecom companies looking to improve customer retention, increase revenue, and gain a competitive edge in the market.

**2. PROBLEM DEFINITION & DESIGN THINKING**

**2.1. EMPATHY MAP**

****

**2.2 IDEATION & BRAINSTORMINGS MAP**

****

**3. RESULT**



**4.ADVANTAGES & DISADVANTAGES**

**ADVANTAGES :**

Enhanced prediction accuracy: Machine learning algorithms can analyze large amounts of customer data and identify patterns that humans may miss. This can lead to more accurate predictions of customer churn, allowing telecom companies to take proactive steps to prevent it.

Targeted retention efforts: By using machine learning to predict customer churn, telecom companies can identify specific customer segments that are at risk of leaving and develop targeted retention strategies to address their needs.

Improved customer satisfaction: By identifying the factors that drive customer churn, telecom companies can take steps to address these issues and improve the overall customer experience, which can lead to higher customer satisfaction and loyalty.

Cost-effective: Retaining existing customers is typically less costly than acquiring new ones. By reducing customer churn, telecom companies can increase revenue and profitability while minimizing acquisition costs.

**DISADVANTAGES :**

Cost: Retaining existing customers can be costly for telecom companies. Offering incentives, such as discounts or free services, can reduce customer churn bu**t** also reduce revenue.

Limited growth: Focusing on customer retention may limit a telecom company's ability to acquire new customers and grow its customer base. This may be particularly challenging for companies operating in a competitive market.

Customer satisfaction: In some cases, a focus on customer retention may lead to a lack of focus on improving the overall customer experience. This can lead to lower customer satisfaction and loyalty over the long term.

Difficulty in predicting customer behavior: Despite using machine learning algorithms to predict customer churn, it can still be challenging to accurately predict customer behavior. This may result in the implementation of retention strategies that are not effective, wasting company resources.

Customer acquisition costs: While retaining existing customers is less costly than acquiring new ones, it still incurs costs. Telecom companies must balance the cost of customer retention with the cost of customer acquisition to ensure that they are maximizing their return on investment.

**5. APPLICATION**

Data analysis and feature engineering, The first step in building a churn prediction model is to collect and analyze customer data, such as demographic information, usage patterns, and customer service interactions. This data can then be used to create new features that can help predict churn, such as the number of dropped calls, the average duration of calls, and the number of times a customer has contacted customer service.Classification models Once the data has been analyzed and the features have been engineered, classification models can be trained to predict whether a customer is likely to churn. Some common classification models used in churn prediction include logistic regression, decision trees, random forests, and support vector machines.Neural networks are a type of machine learning model that can be used to predict churn. They are particularly effective for processing large amounts of data and identifying complex patterns. Some common neural network architectures used in churn prediction include feedforward networks, recurrent networks, and convolutional networks. Deep learning is a subset of neural networks that uses multiple layers to learn complex features from raw data. Deep learning models have been shown to be highly effective in churn prediction, particularly when dealing with unstructured data such as call recordings or text messages.Natural Language Processing (NLP) can be used to analyze unstructured data such as customer feedback, social media posts, and call center recordings. This can help companies identify patterns and trends in customer sentiment, which can be used to improve customer satisfaction and reduce churn

**6. CONCLUSION**

By analyzing customer data and identifying patterns that are predictive of churn, machine learning algorithms can help telecom companies take proactive steps to retain customers, ultimately leading to higher revenue and profitability.

Through the use of advanced analytics techniques such as data pre-processing, feature engineering, and model training, machine learning algorithms can accurately predict which customers are likely to churn. This enables telecom companies to take proactive steps to retain customers, such as offering targeted promotions, improving customer service, or addressing service quality issues.

**7. FUTURE SCOPE**

As customers become more accustomed to personalized experiences in other industries, telecom companies will need to offer personalized services and promotions to retain customers. This may involve using machine learning algorithms to analyze customer data and offer customized promotions and services that are tailored to each customer's needs and preferences.Real-time monitoring and intervention: As telecom services become increasingly complex, real-time monitoring and intervention will become more important for retaining customers. This may involve using machine learning algorithms to monitor network performance and intervene in real-time to address service quality issues and prevent customer churn.Predictive analytics: Predictive analytics will continue to play a key role in telecom customer retention. As machine learning algorithms become more sophisticated, they will be able to identify subtle patterns and trends in customer behavior that may indicate an increased likelihood of churn. This will enable telecom companies to take proactive steps to retain customers before they churn.

Enhanced customer service: Telecom companies will need to invest in enhanced customer service capabilities to retain customers. This may involve using machine learning algorithms to analyze customer interactions and identify areas where customer service can be improved. It may also involve using chatbots and other automated systems to offer 24/7 customer support.5G technology: The rollout of 5G technology will provide new opportunities for telecom companies to offer innovative services and retain customers. For example, 5G technology will enable faster and more reliable network connections, which could be used to offer new services such as augmented reality and virtual reality experiences.

**8. APPENDIX**

**A.SOURCE CODE :**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')

from IPython.core.interactiveshell import InteractiveShell

InteractiveShell.ast\_node\_interactivity = "all"

%matplotlib inline

pd.set\_option("display.max\_columns", 300)

pd.set\_option("display.max\_rows", 300)

churn = pd.read\_csv("/content/Churn\_Modelling.csv")

recharge\_cols = ['total\_rech\_data\_6', 'total\_rech\_data\_7', 'total\_rech\_data\_8', 'total\_rech\_data\_9',

                 'count\_rech\_2g\_6', 'count\_rech\_2g\_7', 'count\_rech\_2g\_8', 'count\_rech\_2g\_9',

                 'count\_rech\_3g\_6', 'count\_rech\_3g\_7', 'count\_rech\_3g\_8', 'count\_rech\_3g\_9',

                 'max\_rech\_data\_6', 'max\_rech\_data\_7', 'max\_rech\_data\_8', 'max\_rech\_data\_9',

                 'av\_rech\_amt\_data\_6', 'av\_rech\_amt\_data\_7', 'av\_rech\_amt\_data\_8', 'av\_rech\_amt\_data\_9',

                 ]

churn[recharge\_cols].describe(include='all')

zero\_impute = ['total\_rech\_data\_6', 'total\_rech\_data\_7', 'total\_rech\_data\_8', 'total\_rech\_data\_9',

        'av\_rech\_amt\_data\_6', 'av\_rech\_amt\_data\_7', 'av\_rech\_amt\_data\_8', 'av\_rech\_amt\_data\_9',

        'max\_rech\_data\_6', 'max\_rech\_data\_7', 'max\_rech\_data\_8', 'max\_rech\_data\_9'

       ]

# impute missing values with 0

churn[zero\_impute] = churn[zero\_impute].apply(lambda x: x.fillna(0))

# now, let's make sure values are imputed correctly

print("Missing value ratio:\n")

print(churn[zero\_impute].isnull().sum()\*100/churn.shape[1])

# summary

print("\n\nSummary statistics\n")

print(churn[zero\_impute].describe(include='all'))

# drop id and date columns

print("Shape before dropping: ", churn.shape)

churn = churn.drop(id\_cols + date\_cols, axis=1)

print("Shape after dropping: ", churn.shape)

# replace missing values with '-1' in categorical columns

churn[cat\_cols] = churn[cat\_cols].apply(lambda x: x.fillna(-1))

# missing value ratio

print("Missing value ratio:\n")

print(churn[cat\_cols].isnull().sum()\*100/churn.shape[0])

initial\_cols = churn.shape[1]

MISSING\_THRESHOLD = 0.7

include\_cols = list(churn.apply(lambda column: True if column.isnull().sum()/churn.shape[0] < MISSING\_THRESHOLD else False))

drop\_missing = pd.DataFrame({'features':churn.columns , 'include': include\_cols})

drop\_missing.loc[drop\_missing.include == True,:]

# drop columns

churn = churn.loc[:, include\_cols]

dropped\_cols = churn.shape[1] - initial\_cols

print("{0} columns dropped.".format(dropped\_cols))

churn\_cols = churn.columns

print('Total Columns :',len(churn\_cols))

# using MICE technique to impute missing values in the rest of the columns

from fancyimpute import MICE

churn\_imputed = MICE(n\_imputations=1).complete(churn)

# convert imputed numpy array to pandas dataframe

churn = pd.DataFrame(churn\_imputed, columns=churn\_cols)

print(churn.isnull().sum()\*100/churn.shape[0])

# calculate the total data recharge amount for June and July --> number of recharges \* average recharge amount

churn['total\_data\_rech\_6'] = churn.total\_rech\_data\_6 \* churn.av\_rech\_amt\_data\_6

churn['total\_data\_rech\_7'] = churn.total\_rech\_data\_7 \* churn.av\_rech\_amt\_data\_7

# calculate total recharge amount for June and July --> call recharge amount + data recharge amount

churn['amt\_data\_6'] = churn.total\_rech\_amt\_6 + churn.total\_data\_rech\_6

churn['amt\_data\_7'] = churn.total\_rech\_amt\_7 + churn.total\_data\_rech\_7

# calculate average recharge done by customer in June and July

churn['av\_amt\_data\_6\_7'] = (churn.amt\_data\_6 + churn.amt\_data\_7)/2

# look at the 70th percentile recharge amount

print("Recharge amount at 70th percentile: {0}".format(churn.av\_amt\_data\_6\_7.quantile(0.7)))

# retain only those customers who have recharged their mobiles with more than or equal to 70th percentile amount

churn\_filtered = churn.loc[churn.av\_amt\_data\_6\_7 >= churn.av\_amt\_data\_6\_7.quantile(0.7), :]

churn\_filtered = churn\_filtered.reset\_index(drop=True)

churn\_filtered.shape

churn\_filtered = churn\_filtered.drop(['total\_data\_rech\_6', 'total\_data\_rech\_7',

                                      'amt\_data\_6', 'amt\_data\_7', 'av\_amt\_data\_6\_7'], axis=1)

churn\_filtered.shape

# calculate total incoming and outgoing minutes of usage

churn\_filtered['total\_calls\_mou\_9'] = churn\_filtered.total\_ic\_mou\_9 + churn\_filtered.total\_og\_mou\_9

# calculate 2g and 3g data consumption

churn\_filtered['total\_internet\_mb\_9'] =  churn\_filtered.vol\_2g\_mb\_9 + churn\_filtered.vol\_3g\_mb\_9

# create churn variable: those who have not used either calls or internet in the month of September are customers who have churned

# 0 - not churn, 1 - churn

churn\_filtered['churn'] = churn\_filtered.apply(lambda row: 1 if (row.total\_calls\_mou\_9 == 0 and row.total\_internet\_mb\_9 == 0) else 0, axis=1)

# delete derived variables

churn\_filtered = churn\_filtered.drop(['total\_calls\_mou\_9', 'total\_internet\_mb\_9'], axis=1)

# change data type to category

churn\_filtered.churn = churn\_filtered.churn.astype("category")

# print churn ratio

print("Churn Ratio:")

print(churn\_filtered.churn.value\_counts()\*100/churn\_filtered.shape[0])

churn\_filtered['arpu\_diff'] = churn\_filtered.arpu\_8 - ((churn\_filtered.arpu\_6 + churn\_filtered.arpu\_7)/2)

churn\_filtered['onnet\_mou\_diff'] = churn\_filtered.onnet\_mou\_8 - ((churn\_filtered.onnet\_mou\_6 + churn\_filtered.onnet\_mou\_7)/2)

churn\_filtered['offnet\_mou\_diff'] = churn\_filtered.offnet\_mou\_8 - ((churn\_filtered.offnet\_mou\_6 + churn\_filtered.offnet\_mou\_7)/2)

churn\_filtered['roam\_ic\_mou\_diff'] = churn\_filtered.roam\_ic\_mou\_8 - ((churn\_filtered.roam\_ic\_mou\_6 + churn\_filtered.roam\_ic\_mou\_7)/2)

churn\_filtered['roam\_og\_mou\_diff'] = churn\_filtered.roam\_og\_mou\_8 - ((churn\_filtered.roam\_og\_mou\_6 + churn\_filtered.roam\_og\_mou\_7)/2)

churn\_filtered['loc\_og\_mou\_diff'] = churn\_filtered.loc\_og\_mou\_8 - ((churn\_filtered.loc\_og\_mou\_6 + churn\_filtered.loc\_og\_mou\_7)/2)

churn\_filtered['std\_og\_mou\_diff'] = churn\_filtered.std\_og\_mou\_8 - ((churn\_filtered.std\_og\_mou\_6 + churn\_filtered.std\_og\_mou\_7)/2)

churn\_filtered['isd\_og\_mou\_diff'] = churn\_filtered.isd\_og\_mou\_8- ((churn\_filtered.isd\_og\_mou\_6 + churn\_filtered.isd\_og\_mou\_7)/2)

churn\_filtered['spl\_og\_mou\_diff'] = churn\_filtered.spl\_og\_mou\_8 - ((churn\_filtered.spl\_og\_mou\_6 + churn\_filtered.spl\_og\_mou\_7)/2)

churn\_filtered['total\_og\_mou\_diff'] = churn\_filtered.total\_og\_mou\_8 - ((churn\_filtered.total\_og\_mou\_6 + churn\_filtered.total\_og\_mou\_7)/2)

churn\_filtered['loc\_ic\_mou\_diff'] = churn\_filtered.loc\_ic\_mou\_8 - ((churn\_filtered.loc\_ic\_mou\_6 + churn\_filtered.loc\_ic\_mou\_7)/2)

churn\_filtered['std\_ic\_mou\_diff'] = churn\_filtered.std\_ic\_mou\_8 - ((churn\_filtered.std\_ic\_mou\_6 + churn\_filtered.std\_ic\_mou\_7)/2)

churn\_filtered['isd\_ic\_mou\_diff'] = churn\_filtered.isd\_ic\_mou\_8 - ((churn\_filtered.isd\_ic\_mou\_6 + churn\_filtered.isd\_ic\_mou\_7)/2)

churn\_filtered['spl\_ic\_mou\_diff'] = churn\_filtered.spl\_ic\_mou\_8 - ((churn\_filtered.spl\_ic\_mou\_6 + churn\_filtered.spl\_ic\_mou\_7)/2)

churn\_filtered['total\_ic\_mou\_diff'] = churn\_filtered.total\_ic\_mou\_8 - ((churn\_filtered.total\_ic\_mou\_6 + churn\_filtered.total\_ic\_mou\_7)/2)

churn\_filtered['total\_rech\_num\_diff'] = churn\_filtered.total\_rech\_num\_8 - ((churn\_filtered.total\_rech\_num\_6 + churn\_filtered.total\_rech\_num\_7)/2)

churn\_filtered['total\_rech\_amt\_diff'] = churn\_filtered.total\_rech\_amt\_8 - ((churn\_filtered.total\_rech\_amt\_6 + churn\_filtered.total\_rech\_amt\_7)/2)

churn\_filtered['max\_rech\_amt\_diff'] = churn\_filtered.max\_rech\_amt\_8 - ((churn\_filtered.max\_rech\_amt\_6 + churn\_filtered.max\_rech\_amt\_7)/2)

churn\_filtered['total\_rech\_data\_diff'] = churn\_filtered.total\_rech\_data\_8 - ((churn\_filtered.total\_rech\_data\_6 + churn\_filtered.total\_rech\_data\_7)/2)

churn\_filtered['max\_rech\_data\_diff'] = churn\_filtered.max\_rech\_data\_8 - ((churn\_filtered.max\_rech\_data\_6 + churn\_filtered.max\_rech\_data\_7)/2)

churn\_filtered['av\_rech\_amt\_data\_diff'] = churn\_filtered.av\_rech\_amt\_data\_8 - ((churn\_filtered.av\_rech\_amt\_data\_6 + churn\_filtered.av\_rech\_amt\_data\_7)/2)

churn\_filtered['vol\_2g\_mb\_diff'] = churn\_filtered.vol\_2g\_mb\_8 - ((churn\_filtered.vol\_2g\_mb\_6 + churn\_filtered.vol\_2g\_mb\_7)/2)

churn\_filtered['vol\_3g\_mb\_diff'] = churn\_filtered.vol\_3g\_mb\_8 - ((churn\_filtered.vol\_3g\_mb\_6 + churn\_filtered.vol\_3g\_mb\_7)/2)