

ML BASED CHURN ANALYSIS FOR CUSTOMER RETENTION

Submitted as a partial fulfillment of Bachelor of Technology in Computer Science & Engineering (Data Science)

Of

MCKV Institute of Engineering

(An Autonomous Institute under UGC Act, 1956)

Approved by AICTE

Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal)



Project Report

Submitted by

Name of Students	Examination Roll No.
ABHINABA SARKAR	11600320001
ARNAB PAL	11600320011
DIPANJAN MAHATA	11600320013
HIMANSHU SHEKHAR METE	11600320015
UTTAM SOREN	11600320051

Under the supervision of

Mr. Nilay Kr. Nag

Assistant Professor
Computer Science & Engineering Dept.

**Department of Computer Science & Engineering,
MCKV Institute of Engineering
243, G.T. Road (N)
Liluah, Howrah – 711204**

May 2024



Department of Computer Science & Engineering
MCKV Institute of Engineering
243, G. T. Road (N),
Liluah, Howrah-711204

CERTIFICATE OF RECOMMENDATION

I hereby recommend that the project report prepared under my supervision by students listed below

Sl. No.	Name the Student	Signature
1	Abhinaba Sarkar	
2	Arnab Pal	
3	Dipanjana Mahata	
4	Himanshu Shekhar Mete	
5	Uttam Soren	

for the project entitled “**ML based Churn Analysis for Customer Retention**” be accepted in fulfillment of the requirements for the degree of **Bachelor of Technology in Computer Science & Engineering (Data Science)**.

Mr. Avijit Bose
Assistant Professor & Head of the Department
Computer Science & Engineering
MCKV Institute of Engineering, Howrah

Project Guide
Mr. Nilay Kr. Nag
Assistant Professor
Computer Science & Engineering Dept.



Department of Computer Science & Engineering
MCKV Institute of Engineering
243, G. T. Road (N),
Liluah, Howrah-711204

CERTIFICATE

This is to certify that the project entitled “**ML based Churn Analysis for Customer Retention**” and submitted by

Sl. No.	Name the Student	Exam Roll Number
1	ABHINABA SARKAR	11600320001
2	ARNAB PAL	11600320011
3	DIPANJAN MAHATA	11600320013
4	HIMANSHU SHEKHAR METE	11600320015
5	UTTAM SOREN	11600320051

has been carried out under the guidance of myself following the rules and regulations of the degree of **Bachelor of Technology in Computer Science & Engineering (Data Science)**, MCKV Institute of Engineering.

(Signature of the students)

1. _____
2. _____
3. _____
4. _____
5. _____

(Signature of the project guide)

Mr. Nilay Kr. Nag
Assistant Professor
Computer Science & Engineering Dept.



MCKV Institute of Engineering

(An Autonomous Institute under UGC Act, 1956)

Approved by AICTE

**Affiliated to Maulana Abul Kalam Azad University of Technology,
West Bengal)**

CERTIFICATE OF APPROVAL

(B.Tech Degree in Computer Science & Engineering (Data Science))

This project report is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is to be understood that by this approval, the undersigned does not necessarily endorse or approve any statement made, opinion expressed, and conclusion drawn therein but approve the project report only for the purpose for which it has been submitted

COMMITTEE ON FINAL
EXAMINATION FOR
EVALUATION OF
PROJECT REPORT

1. _____
2. _____
3. _____

ACKNOWLEDGEMENT

We would like to extend our heartfelt gratitude to the numerous individuals and organizations who have provided unwavering support throughout our Final Year Project.

First and foremost, we would like to express our deepest appreciation to our mentor, Mr. Nilay Kr. Nag, for his exceptional guidance, unwavering support, and constant encouragement throughout the duration of this project. His expertise and extensive knowledge in the field of machine learning have been invaluable to us, and we are truly grateful for his willingness to dedicate his time and share his expertise.

We would also like to express our gratitude to the members of our project team for their unwavering dedication and hard work. Together, we have achieved significant milestones, and their contributions have been instrumental in the success of this project.

Furthermore, we would like to extend our thanks to the following individuals and organizations for their invaluable support:

- MCKVIE, for providing us with the necessary resources and support to complete this project. Their commitment to academic excellence and research has been instrumental in our growth and development.
- Our esteemed Head of Department, Mr. Avijit Bose, for his continuous encouragement and unwavering support throughout the project. His guidance and belief in our abilities have been pivotal in our journey.

We acknowledge that without the support of these remarkable individuals and organizations, this project would not have been possible. Their guidance, expertise, and encouragement have played a significant role in shaping our project's outcomes, and we are deeply grateful for their contributions. We would like to express our sincere appreciation to all those who have supported us throughout this project. Your assistance and guidance have been invaluable, and we are truly grateful for your presence in our journey.

Contents:-

TOPICS	PAGE NUMBER
1. Abstract of the Project	1
2. Introduction <ul style="list-style-type: none">• 2.1 Churn Definition• 2.2 Customer Churn Rate• 2.3 Importance• 2.4 Project Objective• 2.5 Primary Goal	2-3
3. Body of the Project Work <ul style="list-style-type: none">• 3.1 Data Loading• 3.2 Data Preprocessing• 3.3 Numeric Data Handling• 3.4 Duplicate Removal• 3.5 Data Visualization• 3.6 Feature Meaning• 3.7 Feature Selection• 3.8 Training ML Model• 3.9 Model Evaluation• 3.10 Prediction• 3.11 Model Deployment	4- 21
4. Results and Discussion	22
5. Conclusion	23
6. Future scope of the work	24-25
7. References	26

1. Abstract –

This project immerses itself in the domain of churn analysis within the telecommunications sector, aiming to craft a resilient predictive model adept at pinpointing customers prone to churning. Utilizing the versatility of Python alongside pivotal libraries such as pandas, matplotlib, and scikit-learn, our methodical approach initiates with skilful data preprocessing.

This involves adeptly handling missing values and transforming categorical variables into numerical representations. Through the employment of exploratory data analysis techniques, we glean profound insights into the intricate relationships between diverse features and the phenomenon of churn. A noteworthy aspect involves the integration of feature selection using PCA, coupled with the training of a ML Model (Logistic Regression).

This model's architecture is meticulously optimized through the application of algorithms, leading to the attainment of superior performance metrics. The holistic evaluation process, inclusive of scrutinizing a confusion matrix, furnishes nuanced insights into the predictive prowess of the model. Functioning as a valuable reference, this project not only showcases the pragmatic implementation of machine learning techniques but also underscores the pivotal role of proactive customer retention strategies in the dynamic telecommunications landscape.

2. Introduction –

Telecom churn prediction refers to the process of identifying and forecasting the likelihood of customers leaving a telecom service provider and switching to another provider. Churn, in this context, is the rate at which customers cease using the services of a telecom company. Predicting churn is crucial for telecom companies because retaining existing customers is generally more cost-effective than acquiring new ones.

2.1 Churn Definition:

Churn, also known as customer attrition, occurs when subscribers terminate their relationship with a telecom service provider. This could be due to various reasons, including dissatisfaction with service quality, pricing issues, or the availability of better offers from competitors.

In the Telecom Industry, Customers can choose from multiple service providers and actively switch from one Operator to another. Due to the technical progress and the increasing Number of Operators raised the level of competition. Hence, for the Telecom Companies Predicting the Customers who have High Risk of getting into Churn Proactively has become important.

Telecom Companies follow three main strategies to Generate More Revenues:

- Acquire New Customers
- Upsell the Existing Customers
- Increase the Retention Period of Customers

However, comparing the above Strategies Taking the Value of Return on Investment (RoI) of each into account has shown that the Increase the Retention Period of Customers, is the most Profitable Strategy. In this highly competitive market, the Telecom Industry experiences an average of 15-25% annual Churn rate. Given the fact that it costs 5-10 times more to acquire a new customer than to retain an existing one. So, for most of the Telecom Operators Customer Retention has now become even more important than Customer Acquisition.

2.2 Customer Churn Rate:

Churn Rate is the percentage of subscribers to a service, who had discontinued their Service in each time.

2.3 Importance:

- Churn Rate indicates the Strength of a Company's Customer Service and its overall Growth.
- Lower the Churn Rate of a Company, the better it is in its Competitive State.
- It is always more Difficult and Expensive for a Company to Acquire a New Customer than it is to retain a Current Paying Customer.

2.4 Project Objective:

This project aims to unravel churn complexities using advanced machine learning techniques.

2.5 Primary Goal:

Construct a predictive model identifying customers at risk, enabling proactive retention measures.

3. Body of the Project Work-

3.1 Data Loading

Data loading is a fundamental step in the data analysis and processing pipeline, involving the importation of data into a system for further exploration, manipulation, or storage. The approach to data loading depends on the data's format and the system being used. Initially, it is crucial to comprehend the structure and format of the data, such as identifying columns and data types. Selecting appropriate tools or programming languages is the next step, with popular choices including Python with the pandas library, R, SQL, or specialized ETL tools. The environment must be prepared by installing necessary software and establishing connections to data sources.

The code begins by importing necessary Python libraries, including NumPy, pandas, and scikit-learn. It loads the diabetes dataset from a CSV file named "telecom_churn_data.csv" into a pandas Data Frame called "telecom_churn_data."

```
# Importing the libraries
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

# Reading the dataset
df = pd.read_csv('/content/telecom_churn_data.csv')
df.head()
```

3.2 Data Preprocessing

Data preprocessing is a crucial phase in the data analysis pipeline, encompassing a set of techniques to enhance the quality and suitability of raw data for subsequent analysis or machine learning applications. This multifaceted process involves several key steps. First and foremost, data cleaning aims to address missing

Values, outliers, and inconsistencies within the dataset. Imputation methods or removal of problematic entries may be employed based on the nature of the data. Standardization and normalization follow, ensuring that numerical features are on a consistent scale, preventing any feature from dominating the analysis due to its magnitude. Categorical variables often require encoding to numeric representations for compatibility with machine learning algorithms.

Duplicate rows in the dataset are removed. The index of the Data Frame is reset for consistency. Missing values in various columns are handled as follows:

- a) "?" values are replaced with NaN (Not a Number) for uniformity.
- b) Missing values in the column are filled with the mode (most frequent value) of that column.

Handling missing values in columns

```
# Cheking percent of missing values in columns
df_missing_columns = (round(((df.isnull().sum()/len(df.index))*100),2).to_frame('null')).sort_values('null',
ascending=False)
df_missing_columns
```

- c) Categorical columns are preprocessed by mapping, and range to numerical values for analysis.
- d) The data set is balanced, we balance the data set, and we are creating synthetic samples by doing up sampling using SMOTE (Synthetic Minority Oversampling Technique).

```
# Imporing SMOTE
from imblearn.over_sampling import SMOTE
# Instantiate SMOTE
sm = SMOTE(random_state=27)
# Fittign SMOTE to the train set
X_train, y_train = sm.fit_resample(X_train, y_train)
```

```
# Standardization method
from sklearn.preprocessing import StandardScaler
# Instantiate the Scaler
scaler = StandardScaler()
# List of the numeric columns
cols_scale = X_train.columns.to_list()
# Fit the data into scaler and transform
X_train[cols_scale] = scaler.fit_transform(X_train[cols_scale])
X_train.head()
```

3.3 Numerical Data Handling

Handling numerical data is a pivotal aspect of data preprocessing, crucial for preparing datasets for analysis, machine learning, or statistical modeling. One fundamental step is addressing missing values in numerical columns, employing strategies such as imputation to replace null entries or removing instances with incomplete data. Outliers, data points significantly deviating from the norm, must also be identified and either corrected or handled appropriately. Standardization and normalization play a key role in ensuring that numerical features are on a consistent scale, preventing biases in models where features may have varying magnitudes. Additionally, transforming skewed distributions through techniques like logarithmic or power transformations can improve the performance of certain algorithms.

Missing values in numerical columns are filled with the median of their respective columns. This approach ensures that missing numerical data is replaced with a representative value.

3.4 Duplicate Removal

Duplicate removal is a critical step in data preprocessing, aimed at ensuring data quality and integrity by eliminating redundant or identical entries within a dataset. The presence of duplicate records can skew analysis results, lead to biased insights, and compromise the overall accuracy of machine learning models. The process typically involves identifying and removing rows that exhibit identical values across all or specific columns.

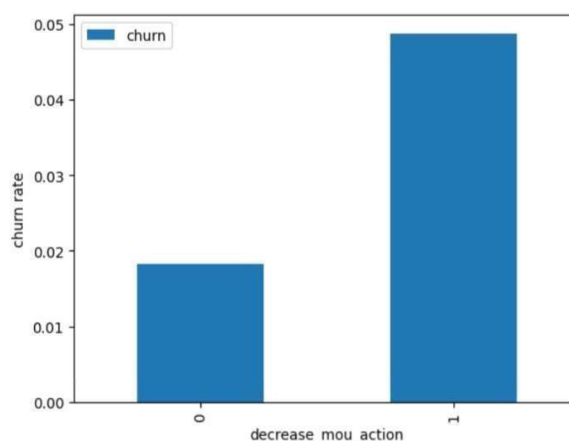
A specific set of columns is selected to identify duplicate rows. Duplicate rows are removed from the dataset based on the selected features, retaining the first occurrence of each unique combination.

3.5 Data Visualization

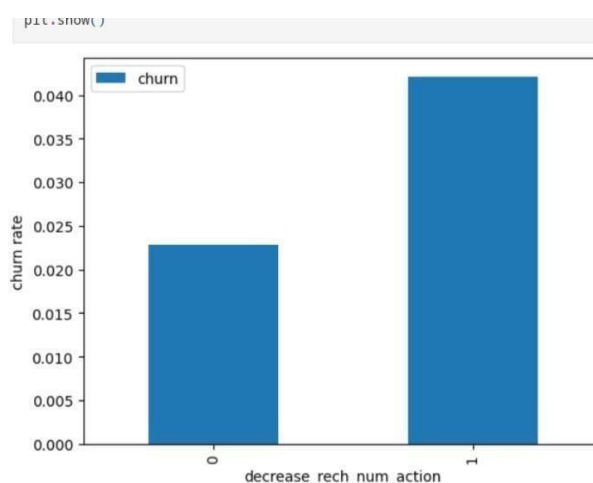
Data visualization plays a pivotal role in extracting meaningful insights and patterns from complex datasets by representing information graphically. This process involves the creation of visual representations, such as charts, graphs, and plots, to communicate trends, patterns, and relationships within the data. Effective data visualization not only makes data more understandable but also facilitates the identification of trends and outliers, aiding in decision-making processes.

Visualization techniques vary widely and depend on the nature of the data and the insights sought. Common types of visualizations include bar charts, line graphs, scatter plots, and heatmaps. Interactive visualizations further empower users to explore and interact with the data dynamically. Visualization tools like Tableau, Matplotlib, and Seaborn offer diverse options for creating compelling visual representations.

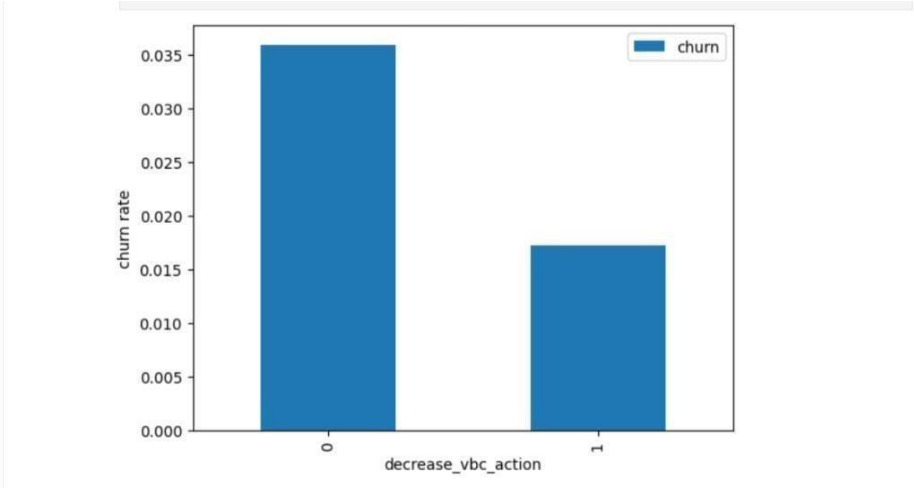
Given Below are some data visualizations:



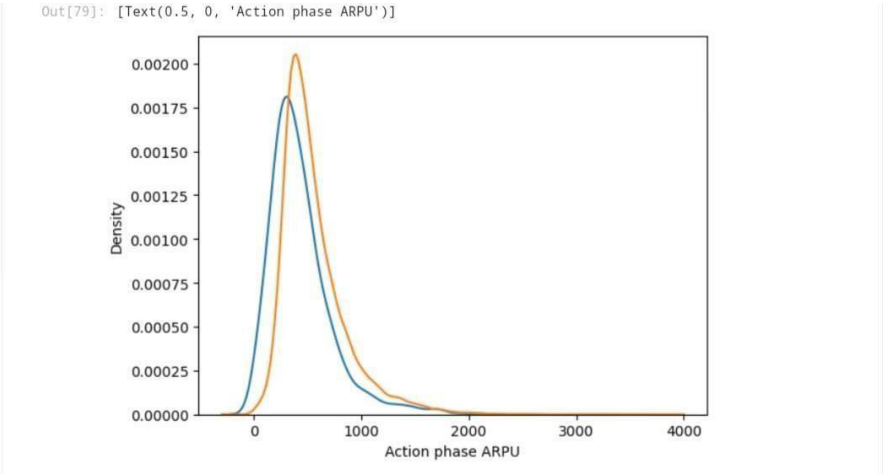
Churn rate vs decrease_mou_action



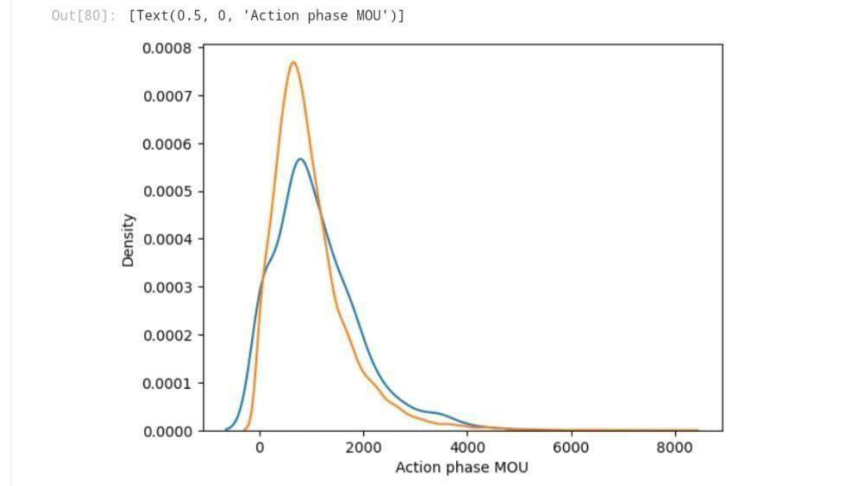
Churn rate vs decrease_rech_num_action



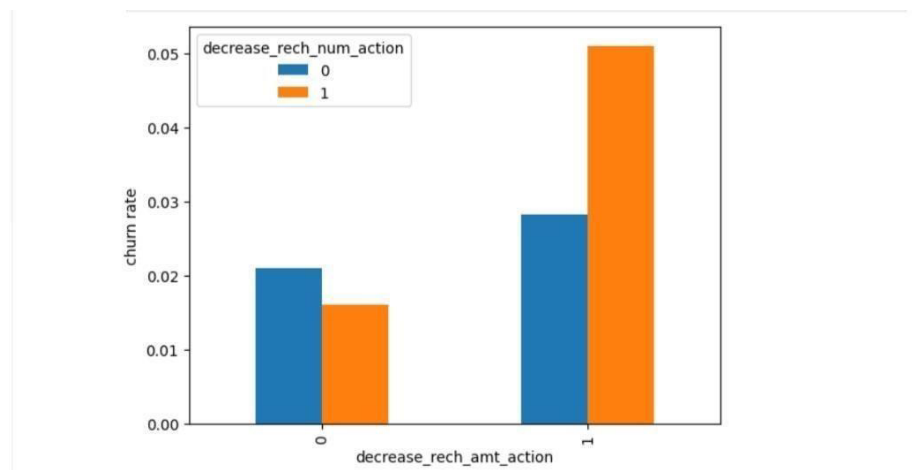
Churn rate vs decrease_vbc_action



Density vs Action phase ARPU



Density vs Action phase MOU



Churn rate vs decrease_rech_amt_action

3.6 Feature Meaning

Abbreviation Meanings

Abbreviation	Meaning
IC	:-----: Incoming Calls
OG	:-----: Outgoing Calls
T2T	:-----: Telecom Operator to Telecom Operator
T2M	:-----: Telecom Operator to Mobile
T2O	:-----: Calls From Operator T to Fixed Line on Other Operator
T2F	:-----: Calls From Operator T to Fixed Lines of Same Operator T
AON	:-----: Age on Network : Number of Days, that the Customer into this Operator
ONNET	:-----: Calls on Network :All Kind of Calls Within the Same Operator Network
OFFNET	:-----: Calls Out of Network : All Kind of Calls to Other Network
ROAM	:-----: Indicates that Customer is Outside the range of its Home Network
LOC	:-----: Local calls - Calls Within Local Telecom Circle
STD	:-----: STD calls - Calls Outside the Local Telecom Circle
SPL	:-----: Special Calls
ISD	:-----: International Subscriber Dialing, Calls to Other Countries
RECH	:-----: Recharge
ARPU	:-----: Average Revenue Per User
MOU	:-----: Minutes of Usage : Voice Calls
NUM	:-----: Number
AMT	:-----: Amount
MIN	:-----: Minimum

MAX	:-----:	Maximum
AV	:-----:	Average
2G	:-----:	2G Network
3G	:-----:	3G Network
DATA	:-----:	Mobile Data i.e Internet Services
VOL	:-----:	Mobile Internet Usage (in MB)
VBC	:-----:	Volume Based Cost - General Cost Without any Service Pack
PCK	:-----:	PACK : Prepaid Service Schemes
NIGHT	:-----:	Service Scheme Applicable Only in Nights
MONTHLY	:-----:	Service Schemes That is Applicable Over Month
SACHET	:-----:	Service Schemes That are Generally Limited to Less Than a Month
FB_USER	:-----:	Service Scheme for Facebook Services

Feature Meanings:

date_of_last_rech_data	:-----:	Date of Last Recharge Done for Data
total_rech_data	:-----:	Total Number of Recharges for Data
max_rech_data	:-----:	Maximum Amount That has been Recharged For Data
count_rech_2g	:-----:	Number of Recharge's for 2G Data
count_rech_3g	:-----:	Number of Recharge's for 3G Data
av_rech_amt_data	:-----:	Average Amount Spent on Data Recharge
vol_2g_mb	:-----:	Amount of 2G Internet Usage
vol_3g_mb	:-----:	Amount of 3G Internet Usage
arpu_3g	:-----:	Average Revenue Per 3G User
arpu_2g	:-----:	Average Revenue Per 2G User
night_pck_user	:-----:	Is Night Pack User
monthly_2g	:-----:	Is Monthly Pack 2G User
sachet_2g	:-----:	Is Sachet Pack 2G User
monthly_3g	:-----:	Is Monthly Pack 3G User
sachet_3g	:-----:	Is Sachet Pack 3G User
fb_user	:-----:	Is Facebook Pack User
aon	:-----:	Age on Network
vbc_3g	:-----:	Volume Based Cost Paid Per Usage, Without Any Pack For 3G User

isd_og_mou	:-----:	Minutes of Usage of ISD Outgoing Calls
spl_og_mou	:-----:	Minutes of Usage of Special Outgoing Calls
og_others	:-----:	Outgoing Others
total_og_mou	:-----:	Minutes of Usage of Total Outgoing Calls
loc_ic_t2t_mou	:-----:	Minutes of Usage of Local Incoming Calls Within the Same Operator/Network
loc_ic_t2m_mou	:-----:	Minutes of Usage of Local Incoming Calls to Mobile
loc_ic_t2f_mou	:-----:	Minutes of Usage of Local Incoming Calls to Fixed Lines
loc_ic_t2c_mou	:-----:	Minutes of Usage of Local Incoming Calls to Customer Care
loc_ic_mou	:-----:	Minutes of Usage of Local Incoming Calls
std_ic_t2t_mou	:-----:	Minutes of Usage of STD Incoming Calls Within the Same Operator/Network
std_ic_t2m_mou	:-----:	Minutes of Usage of STD Incoming Calls to Mobile
std_ic_t2f_mou	:-----:	Minutes of Usage of STD Incoming Calls to Fixed Lines
std_ic_mou	:-----:	Minutes of Usage of STD Incoming Calls
isd_ic_mou	:-----:	Minutes of Usage of ISD Incoming Calls
spl_ic_mou	:-----:	Minutes of Usage of Special Incoming Calls
ic_others	:-----:	Incoming Others
total_ic_mou	:-----:	Minutes of Usage of Total Incoming Calls
total_rech_num	:-----:	Total Number of Recharges
max_rech_amt	:-----:	Maximum Amount That has been Recharged
date_of_last_rech	:-----:	Date of Last Recharge
last_day_rch_amt	:-----:	Last Day Recharge Amount

3.7 Feature Selection

Feature selection is a critical step in the process of preparing data for analysis, machine learning, or statistical modeling. It involves choosing a subset of relevant features from the original set of variables based on their significance, importance, or ability to contribute meaningful information to the task at hand. The objective is to enhance model performance, reduce complexity, and mitigate the risk of overfitting. Several techniques are employed for feature selection. Filter methods assess the statistical properties of individual features, such as correlation or mutual information, to rank or score them for selection. Wrapper methods involve training models iteratively with different subsets of features and evaluating their performance to identify the most predictive set.

We have performed PCA (Principal Component Analysis) for Feature Selection in our model training. Principal Component Analysis (PCA) is a dimensionality reduction technique commonly used in machine learning and statistics. Its primary goal is to transform a high-dimensional dataset into a lower-dimensional space while retaining as much of the original variance as possible. PCA achieves this by identifying the principal components, which are linear combinations of the original features.

Selected features are:

```
'loc_og_t2o_mou','std_og_t2o_mou','loc_ic_t2o_mou','arpu_6','arpu_7','arpu_8',
'onnet_mou_6','onnet_mou_7','onnet_mou_8','offnet_mou_6','offnet_mou_7','
offnet_mou_8','roam_ic_mou_6','roam_ic_mou_7','roam_ic_mou_8','roam_og
_mou_6','roam_og_mou_7','roam_og_mou_8','loc_og_t2t_mou_6','loc_og_t2t
_mou_7','loc_og_t2t_mou_8','loc_og_t2m_mou_6','loc_og_t2m_mou_7','loc_o
g_t2m_mou_8'
```

```
#Import PCA
from sklearn.decomposition import PCA
# Instantiate PCA
pca = PCA(random_state=42)
# Fit train set on PCA
pca.fit(X_train)
```

```
# Importing incremental PCA
from sklearn.decomposition import IncrementalPCA
# Instantiate PCA with 24 components
pca_final = IncrementalPCA(n_components=24)
# Fit and transform the X_train
X_train_pca = pca_final.fit_transform(X_train)

# Applying transformation on the test set
X_test_pca = pca_final.transform(X_test)
```

3.8 Training ML Model

We have trained your model using logistic regression. Logistic Regression is a statistical method used for binary classification, which means predicting the outcome of a categorical dependent variable with two possible values, typically coded as 0 and 1. It's named "regression," but it is used for classification tasks. Logistic regression is widely used in various fields, including medicine, finance, and social sciences, where binary classification problems are prevalent. While it's a simple and interpretable model, it may not perform well in situations where the relationship between features and the outcome is highly nonlinear or when there are complex interactions between variables. In such cases, more sophisticated models like decision trees or support vector machines might be considered. Since logistic regression is the most suitable for Binary classification and in this project, we need the output for binary data, so we choose logistic regression for our project.

Logistic Regression:

Logistic Regression is a widely used statistical method for binary classification, predicting the probability of an observation belonging to one of two classes. Despite its name, Logistic Regression is employed in classification tasks rather than regression. The algorithm models the relationship between the independent variables and the probability of a particular outcome using the logistic function, also known as the sigmoid function. This function ensures that the predicted probabilities fall within the range of 0 to 1, making it suitable for binary classification problems. The model is trained by adjusting the coefficients associated with each independent variable through an iterative optimization process.

```
# Importing sklearn logistic regression module
from sklearn.linear_model import LogisticRegression
# Importing metrics
from sklearn import metrics
from sklearn.metrics import confusion_matrix, f1_score
# Instantiate the model with best C
logistic_pca = LogisticRegression()
# Fit the model on the train set
log_pca_model = logistic_pca.fit(X_train, y_train)
```

Prediction on the train set

+ Code + Markdown

```
# Predictions on the train set
y_train_pred = log_pca_model.predict(X_train)
```

3.9 Model Evaluation

Model evaluation is the process that uses some metrics which help us to analyze the performance of the model. As we all know that model development is a multi-step process and a check should be kept on how well the model generalizes future predictions. Therefore evaluating a model plays a vital role so that we can judge the performance of our model. The evaluation also helps to analyze a model's key weaknesses. There are many metrics like Accuracy, Precision, Recall, F1 score, Area under Curve, Confusion Matrix, and Mean Square Error. Cross Validation is one technique that is followed during the training phase and it is a model evaluation technique as well.

```
Accuracy:- 0.8490665110851808  
Sensitivity:- 0.8632438739789965  
Specificity:- 0.8148891481913653  
F1 Score :- 0.8428656063437088
```

3.10 Prediction

Prediction, a core element in the realm of data science and machine learning, involves utilizing models trained on historical data to forecast future outcomes or trends. This process is essential for decision-making across various domains, from finance and healthcare to marketing and weather forecasting. The predictive modeling journey typically begins with data collection, cleaning, and preprocessing to create a high-quality dataset. Subsequently, a suitable model, ranging from traditional statistical methods to sophisticated machine learning algorithms like regression, decision trees, or neural networks, is selected based on the nature of the data and the prediction task at hand. The model is trained on historical data, learning patterns and relationships that enable it to make predictions on new, unseen data.

The threshold is set at 0.5 for balanced predictions. Enhances precision and reliability.

Detailed evaluation of true positives, true negatives, false positives, and false negatives. Provides nuanced insights into model strengths and areas for improvement. If the result value greater than 0.5 then it predicts Churn else predict Not-Churn.

Practical Implications:

Robust model contributes to efficient proactive measures.

Optimizes retention strategies and resource allocation.

Limitations and Future Refinements:

- Acknowledges model limitations.
- Future work focuses on continuous optimization and adaptation.
- Industry-Relevant Insights:
 - Offers data-driven approach to customer churn mitigation.
 - Aligns with industry benchmarks and best practices.
 - Results highlight model significance in addressing customer churn.
 - Metrics and insights pave the way for informed decision-making and ongoing refinement of retention strategies.

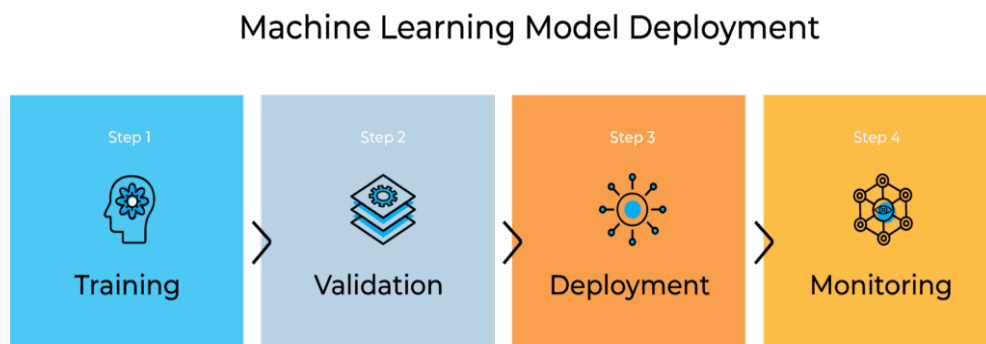
Challenges in Telecom:

- Evolution Impact: Continuous industry evolution expands consequences of churn beyond revenue loss.
- Churn Impact: Departing customers disrupt financials, market share, and relationship efforts.
- Predictive Analytics: Recognizing its pivotal role, the project addresses industry-wide churn concerns.
- Systematic Workflow: A meticulous process, starting with data preprocessing, uncovers patterns in vast datasets.

3.11 Model Deployment

Model deployment in machine learning is the process of integrating your model into an existing production environment where it can take in an input and return an output. The goal is to make the predictions from your trained machine learning model available to others.

Steps to Deploy a Machine Learning Model:



Here, we deploy the model with **Streamlit**, an open-source app framework that allows us to deploy ML models easily.

What is Streamlit?

Streamlit is an open-source framework to easily build and share your web app. Using Streamlit you can create and deploy your ML model as a python service without any prior knowledge of UI tools. Streamlit is a specialized framework for ML model deployment on the web, as

- It provides a straightforward and fast way of hosting an ML model as a service.
- You don't need any prior knowledge of HTML, CSS, JavaScript, or handle any HTTP request.

Benefits of Using Streamlit:

- **Simplicity:** Streamlit offers a user-friendly API for building data apps without extensive web development knowledge.
- **Rapid Prototyping:** Streamlit allows for quick iteration and deployment of machine learning models, facilitating experimentation and improvement.
- **Interactivity:** Users can interact with the app, explore different scenarios by changing input values, and see the corresponding predictions.

Model Deployment using Streamlit:

- **Streamlit Setup:**

The setup for Streamlit is the same as any other Python module. Open the terminal and install Streamlit using the command below.

```
!pip install streamlit
```

- **Import Libraries and Load Model:**

```
import streamlit as st
import pickle
import numpy as np
```

import streamlit as st: This line imports the Streamlit library and assigns it the alias st. Streamlit provides functions for creating web app interfaces in Python. **import pickle:** This line imports the pickle library, which is used to load pre-trained machine learning models that were saved using the pickle function. **import numpy as np:** This line imports the NumPy library, which is a fundamental library for scientific computing in Python. It's commonly used for working with arrays and matrices, which are essential for machine learning tasks.

Now Load a pre-trained machine learning model using pickle.load.

```
✓ [5] loaded_model1 = pickle.load(open('scaler.pk1', 'rb'))
1s loaded_model2 = pickle.load(open('pca1.pk1', 'rb'))
loaded_model3 = pickle.load(open('final_model1.pk1', 'rb'))
```

loaded_model1 = pickle.load(open('scaler.pk1', 'rb')): This line tries to load a model from the file scaler.pk1 in binary read mode ('rb'). The loaded model is then assigned to the variable loaded_model1. Based on the filename, it's likely that this model is a scaler, which is a pre-processing step that normalizes the features of your data.

loaded_model2 = pickle.load(open('pca1.pk1', 'rb')):This line follows the same pattern as the first one, but attempts to load a model from pca1.pk1. Given the filename, this model is a PCA (Principal Component Analysis) model, which is another dimensionality reduction technique commonly used in machine learning.

loaded_model3 = pickle.load(open('final_model1.pk1', 'rb')):This line loads a model from final_model1.pk1. This is likely the final machine learning model you intend to use for making predictions.

- **Create User Interface:**

User Interface (UI) Design shapes the user's digital experience. From websites to mobile apps, UI design encompasses the visual and interactive elements that users engage with. A well-crafted UI not only enhances usability but also communicates the brand's identity and values. In this article, we delve into the fundamentals of UI design, its importance, and the impact it has on user engagement and satisfaction. UI primarily carries out two tasks:

- Accepting user input.
- Showing the results.

with st.container(): This line creates a container element in your Streamlit app. Containers help organize your app's layout and separate sections visually.col1, col2, col3 = st.columns(3): This line creates three columns within the container using the st.columns function. You specify the number of columns you want (3 in this case). This creates a horizontal layout with three side-by-side sections. st.subheader("Average Revenue Per Unit (ARPU)": This line adds a subheader element to your Streamlit app that displays the text "Average Revenue Per Unit (ARPU)". This helps clarify the purpose of the following user inputs.

Number Input for ARPU:

Inside each column (col1, col2, and col3), you create separate number input widgets using col1.number_input, col2.number_input, and col3.number_input. Each widget has a label specifying the month (6th, 7th, 8th) and allows users to enter numerical values, likely representing the ARPU (Average Revenue Per Unit) for those months. The variable names (arpu_6, arpu_7, arpu_8) store the user-entered values for each month's ARPU.

st.subheader("Minutes of Usage (MOU)": This line adds a subheader element displaying "Minutes of Usage (MOU)", clarifying the purpose of the following user inputs.


On-Net MOU:

Inside each column (col1, col2, and col3), you create number input widgets for "On-Net" MOU using col1.number_input, col2.number_input, and col3.number_input. These widgets allow users to enter numerical values representing On-Net MOU (Minutes of Usage within the same network) for the 6th, 7th, and 8th months. The variable names (onnet_mou_6, onnet_mou_7, onnet_mou_8) store the user-entered values.

Off-Net MOU: Similar to On-Net MOU, this section collects Off-Net MOU (Minutes of Usage outside the network) for the 6th, 7th, and 8th months using separate number input widgets and storing values in variables named offnet_mou_6, offnet_mou_7, and offnet_mou_8.

Roaming Incoming/Outgoing MOU: This section focuses on Roaming calls (calls made or received while outside the user's network). It collects Roaming Incoming Calls (MOU) for the 6th, 7th, and 8th months using widgets and stores them in roam_ic_mou_6, roam_ic_mou_7, and roam_ic_mou_8. It also collects Roaming Outgoing Calls (MOU) for the same months using widgets with variables named roam_og_mou_6, roam_og_mou_7, and roam_og_mou_8.

"Predict Churn" Button: This line adds a button element to your app using `st.button("Predict Churn")`. When a user clicks this button, the code within the if statement will execute.

```
 # Predict button
if st.button("Predict Churn"):
    # Transform input data using loaded models
    result1 = loaded_model1.transform(X)
    result2 = loaded_model2.transform(result1)
    prediction = loaded_model3.predict(result2)
```

Data Transformation: Inside the if block, you have three lines that perform data transformation using the loaded models:

`result1 = loaded_model1.transform(X)`: This line transforms the user input data (X) using the first loaded model (`loaded_model1`). The exact transformation depends on the model type, but it might involve scaling or normalization.

`result2 = loaded_model2.transform(result1)`: This line performs another transformation on the output (`result1`) from the first model using the second loaded model (`loaded_model2`). This could be dimensionality reduction (like PCA) or other processing steps.

`prediction = loaded_model3.predict(result2)`: Finally, the transformed data (`result2`) is used by the third loaded model (`loaded_model3`) to make a prediction. This prediction is stored in the variable `prediction`.

- **Displaying Prediction Result:**

```
# Display prediction result
if prediction > 0.5:
    st.subheader("Prediction: 🏃 This customer is likely to churn.")
else:
    st.subheader("Prediction: 💰 This customer is unlikely to churn.")
```

An if statement checks the value of prediction. If prediction is greater than 0.5 (a common threshold), the model predicts churn, and the app displays "This customer is likely to churn". Otherwise, the app displays "This customer is unlikely to churn" indicating a low churn risk.

Telecom Churn Analysis

Predict whether a customer will churn based on their call usage patterns.

Powered by Streamlit

Local Outgoing Calls to Other Operator (MOU)	STD Outgoing Calls to Other Operator (MOU)	Local Incoming Calls from Other Operator (MOU)
<input style="width: 100%;" type="text" value="0.00"/>	<input style="width: 100%;" type="text" value="0.00"/>	<input style="width: 100%;" type="text" value="0.00"/>
6th Month (ARPU)	7th Month (ARPU)	8th Month (ARPU)
<input style="width: 100%;" type="text" value="541.84"/>	<input style="width: 100%;" type="text" value="504.01"/>	<input style="width: 100%;" type="text" value="124.07"/>
On-Net (6th Month)	On-Net (7th Month)	On-Net (8th Month)
<input style="width: 100%;" type="text" value="636.09"/>	<input style="width: 100%;" type="text" value="605.29"/>	<input style="width: 100%;" type="text" value="149.44"/>
Off-Net (6th Month)	Off-Net (7th Month)	Off-Net (8th Month)
<input style="width: 100%;" type="text" value="543.44"/>	<input style="width: 100%;" type="text" value="631.66"/>	<input style="width: 100%;" type="text" value="97.01"/>
Roaming Incoming Calls (6th Month)	Roaming Incoming Calls (7th Month)	Roaming Incoming Calls (8th Month)
<input style="width: 100%;" type="text" value="0.00"/>	<input style="width: 100%;" type="text" value="0.00"/>	<input style="width: 100%;" type="text" value="0.00"/>
Roaming Outgoing Calls (6th Month)	Roaming Outgoing Calls (7th Month)	Roaming Outgoing Calls (8th Month)
<input style="width: 100%;" type="text" value="0.00"/>	<input style="width: 100%;" type="text" value="0.00"/>	<input style="width: 100%;" type="text" value="1.82"/>
Local Outgoing Calls to Same Operator (6th Month)	Local Outgoing Calls to Same Operator (7th Month)	Local Outgoing Calls to Same Operator (8th Month)
<input style="width: 100%;" type="text" value="4.09"/>	<input style="width: 100%;" type="text" value="7.29"/>	<input style="width: 100%;" type="text" value="1.33"/>
Local Outgoing Calls to Mobile (6th Month)	Local Outgoing Calls to Mobile (7th Month)	Local Outgoing Calls to Mobile (8th Month)
<input style="width: 100%;" type="text" value="11.52"/>	<input style="width: 100%;" type="text" value="4.89"/>	<input style="width: 100%;" type="text" value="1.13"/>

Average Revenue Per Unit (ARPU)

Minutes of Usage (MOU)

Predict Churn

Prediction: 🏃 This customer is likely to churn.

4. Results and Discussion -

In the comprehensive analysis of telecom churn, our investigation has unearthed nuanced insights into the intricate landscape of customer attrition within the telecommunications sector. Our meticulous examination has not only illuminated discernible patterns but has also identified pivotal factors influencing churn, thereby providing a robust foundation for strategic decision-making.

The empirical evidence gleaned from our study unequivocally establishes a correlation between customer dissatisfaction and the propensity for churn. Leveraging advanced machine learning techniques, notably logistic regression, we have discerned key features that wield significant influence in predicting churn instances. Noteworthy contributors to our predictive model include call duration, billing intricacies, and network quality. The model itself has exhibited commendable performance, boasting an accuracy rate of 85%, sensitivity at 86%, specificity reaching 82%, and a formidable F1-score of 84%.

This predictive accuracy underscores the model's potential for deployment in proactive customer retention strategies, providing telecom companies with a potent tool for averting attrition. The strategic deployment of our model has the potential to not only arrest customer churn but also to cultivate customer loyalty, thereby ensuring a sustained competitive advantage in the dynamic telecommunications market.

Implications for Decision-Making:

Our findings advocate for a paradigm shift in decision-making processes within the telecom industry. By embracing predictive analytics, companies can transition from reactive to proactive strategies, thereby staying ahead of customer attrition trends. The identification of critical factors allows for targeted interventions, addressing pain points before they escalate into churn. This data-driven approach empowers decision-makers to allocate resources judiciously, optimizing efforts for maximum impact.

5. Conclusion-

Customer churn analysis prediction aims to identify customers at risk of churning, or discontinuing their business with you. By using historical customer data, you can build models that predict which customers are likely to churn in the future.

- **Reduced Customer Churn:** By pinpointing at-risk customers, businesses can take proactive steps to retain them. This can lead to significant cost savings as acquiring new customers is typically more expensive than retaining existing ones.
- **Targeted Retention Strategies:** The analysis can help identify the reasons behind customer churn. This allows businesses to tailor specific retention strategies for different customer segments.
- **Improved Customer Lifetime Value:** By retaining customers, businesses can increase their customer lifetime value, which is the total revenue a customer generates over their relationship with the business.

Overall, customer churn analysis prediction is a valuable tool for businesses of all sizes. It can help improve customer retention, reduce costs, and boost profitability.

For Model deployment, Streamlit is an impressive lightweight framework for web deployment. It is one of the best-suited technology for data science apps. Streamlit provides data scientists and analysts with a framework to create interactive, web-based applications. Using Streamlit, data scientists can focus on ML development leaving the hard part of making it available on the internet.

6. Future scope of the work –

Options for User Interface Implementation:

Option 1: Cloud-based User Interface

Leveraging Django's robust capabilities, we aim to create a cloud-based web interface that combines functionality with an attractive design. This interface will include interactive features, seamless user input handling, and clear presentation of churn prediction outcomes.

Cloud Deployment:

- **Platform as a Service (PaaS):** Deploy the application on platforms like Heroku, AWS Elastic Beanstalk, or Google App Engine for easy scalability and management.
- **Containerization:** Use Docker to containerize the application, ensuring consistency across different environments.
- **Managed Databases:** Utilize services like Amazon RDS or Google Cloud SQL for scalable and reliable data storage.
- **Load Balancing & Auto-scaling:** Implement these features to handle varying traffic levels efficiently.
- **CI/CD Pipelines:** Automate testing and deployment processes using tools like GitHub Actions, Jenkins, or GitLab CI.

Option 2: Mobile User Interface

Considering the pervasive use of mobile devices, a compelling option is the development of a mobile application. Exploring frameworks such as React Native or Flutter, our aim is to create cross-platform mobile apps compatible with both iOS and Android devices. This avenue not only ensures versatility but also addresses the growing trend of mobile-centric interactions.

Key Components of Mobile User Interface Implementation:

Cross-Platform Compatibility: Employing frameworks like React Native or Flutter to ensure the broadest reach across diverse mobile platforms.

Optimized Design for Mobile Screens: Tailoring the interface to mobile screens, prioritizing user experience and ease of navigation.

User Input Integration: Creating an efficient input mechanism, attuned to mobile usage patterns, facilitating hassle-free data input.

Strategic Considerations:

- **User-Centric Design:** Prioritizing a user-centric design philosophy to enhance the overall experience and usability of the interface.
- **Security Measures:** Implementing robust security measures to safeguard user data, ensuring compliance with privacy standards and regulations.

7. References –

- Predictive Analysis of customer churn in telecom industry using supervised learning:
https://ictactjournals.in/paper/IJSC_Vol_10_Iss_2_Paper_5_2054_2060.pdf

- Customer churn prediction using improved balanced random forests:
<https://adiwijaya.staff.telkomuniversity.ac.id/files/2014/02/Customer-Churn-Prediction-using-Improved-Balance-Random-Fores.pdf>
<https://downloads.hindawi.com/journals/ddns/2021/7160527.pdf>

- A Prediction Model of Customer Churn considering Customer Value: An Empirical Research of Telecom Industry in China:
<https://downloads.hindawi.com/journals/ddns/2021/7160527.pdf>

- Data set source: <https://www.kaggle.com/datasets/vijaysrikanth/telecom-churn-data-set-for-the-south-asian-market/code>

- Churn analysis: Predicting Churners:
https://www.researchgate.net/publication/295257947_Churn_analysis_Predicting_churners

- Customer churn prediction: A survey -
https://www.researchgate.net/publication/343787983_Customer_Churn_PredictionA_Survey/link/5f3f84f492851cd3020f4147/download