A Project

Report on

CUSTOMER CHURN PREDICTION

USING MACHINE LEARNING

Submitted to

Jawaharlal Nehru Technological University Hyderabad

In Partial fulfillment of requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

By

CHINTHAKINDI SRI SAI	19BD1A1213
SHAIK INTHIYAZ	19BD1A1245
KASIREDDY HARSHA VARDHAN REDDY	19BD1A1226
NEELAKANTI AMITH	19BD1A1232

Under the guidance of

Ms. SWARAJYA LAXMI

Assistant Professor

Department of IT

DEPARTMENT OF INFORMATION TECHNOLOGY

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(Accredited by NBA & NAAC, Approved by AICTE,

Affiliated to JNTUH 3-5-1206, Narayanaguda, Hyderabad – 500029)

Academic Year: 2022-20

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(Accredited by NBA & NAAC, Approved By A.I.C.T.E.,

Reg by Govt of Telangana State & Affiliated to JNTUH, Hyderabad)

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that this is a bonafide record of the project report titled **"CUSTOMER CHURN PREDICTION USING MACHINE LEARNING"** which is being presented as the Project report by

1. CHINTHAKINDI SRI SAI	**19BD1A1213**
2. SHAIK INTHIYAZ	**19BD1A1245**
3. KASIREDDY HARSHA VARDHAN REDDY	**19BD1A1226**
4.NEELAKANTI AMITH	**19BD1A1232**

In partial fulfillment for the award of the degree of Bachelor of Technology in InformationTechnology affiliated to the Jawaharlal Nehru Technological University Hyderabad, Hyderabad.

Internal Guide	**Head of Department**
(Ms.SWARAJYA LAXMI)	**(Dr. G . NARENDER)**

Submitted for Viva Voce Examination held on________________________________

External Examiner

Vision of KMIT

- To be the fountainhead in producing highly skilled, globally competent engineers.
- Producing quality graduates trained in the latest software technologies and related tools and striving to make India a world leader in software products and services.

Mission of KMIT

- To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modal platforms and prepares students to become successful professionals.
- To establish an industry institute Interaction to make students ready for the industry.
- To provide exposure to students on the latest hardware and software tools.
- To promote research-based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises.
- To induce a spirit of nationalism which will enable the student to develop, understand India's challenges and to encourage them to develop effective solutions.
- To support the faculty to accelerate their learning curve to deliver excellent service to students.

Vision of the IT

To produce globally competent graduates to meet the modern challenges through contemporary knowledge and moral values committed to build a vibrant nation.

Mission of the IT

- To create an academic environment, which promotes the intellectual and professional development of students and faculty.
- To impart skills beyond university prescribed to transform students into a well-rounded IT professional.
- To nurture the students to be dynamic, industry ready and to have multidisciplinary skills including e-learning, blended learning and remote testing as an individual and as a team.
- To continuously engage in research and projects development, strategic use of emerging technologies to attain self-sustainability.

PROGRAM OUTCOMES (POs)

1. **Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis:** Identify formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/Development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct Investigations of Complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage:** Create select, and, apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The Engineer and Society:** Apply reasoning informed by contextual knowledge to societal, health, safety. Legal und cultural issues and the consequent responsibilities relevant to professional engineering practice.
7. **Environment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-Long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analyze the common business functions to design and develop appropriate Information Technology solutions for social upliftments.

PSO2: Shall have expertise on the evolving technologies like Python, Machine Learning, Deep learning, Data Science, Full stack development, Social Networks, Cyber Security, Mobile Apps, CRM, ERP, Big Data, etc.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

PEO2: Graduates will try to provide solutions to challenging problems in their profession by applying computer engineering principles.

PEO3: Graduates will engage in life-long learning and professional development by rapidly adapting to the changing work environment.

PEO4: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

PROJECT OUTCOMES

P1: Automatic attendance generation of students.

P2: Helps organizations to understand about attendance data

P3: Minimizing the chances of proxy attendance

P4: Contactless attendance generation

.

L – LOW

M--MEDIUM

H – HIGH

PROJECT OUTCOMES MAPPING PROGRAM OUTCOMES

PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
M	M	H	M	L	M	M	L	L	M	L	M	M	L	M

PROJECT OUTCOMES MAPPING PROGRAM SPECIFIC OUTCOMES

PSO	PSO1	PSO2
P1	M	H
P2	H	M

PROJECT OUTCOMES MAPPING
PROGRAM EDUCATIONAL OBJECTIVES

PEO	PEO1	PEO2	PEO3	PEO4
P1	L	M	H	H
P2	M	H	M	H

DECLARATION

We hereby declare that the results embodied in the dissertation entitled "CUSTOMER CHURN PREDICTION USING MACHINE LEARNING" has been carried out by us together during the academic year 2022-23 as a partial fulfillment of the award of the B.Tech degree in Information Technology from JNTUH. We have not submitted this report to any other university or organization for the award of any other degree.

Student Name	**Roll No.**
CHINTHAKINDI SRI SAI	**19BD1A1213**
SHAIK INTHIYAZ	**19BD1A1245**
KASIREDDY HARSHA VARDHAN REDDY	**19BD1A1226**
NEELAKANTI AMITH	**19BD1A1232**

ACKNOWLEDGEMENT

We take this opportunity to thank all the people who have rendered their full support to our project work. We render our thanks to **Dr. Maheshwar Dutta**, Principal who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Director for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Mr. S. Nitin, Director** and **Ms. S. Anuradha**, Director Academic for providing an excellent environment in the college.

We are also thankful to **Dr. G. Narender**, Head of the Department for providing us with both time and amenities to make this project a success within the given schedule.

We are also thankful to our guide **Ms. Swarajya Laxmi**, for her valuable guidance and encouragement given to us throughout the project work.

We would like to thank the entire IT Department faculty, who helped us directly and indirectly in the completion of the project.

We sincerely thank our friends and family for their constant motivation during the project work.

Student Name	**RollNo.**
Chinthakindi Sri Sai	**19BD1A1213**
Shaik Inthiyaz	**19BD1A1245**
Kasireddy Harsha	**19BD1A1226**
Neelakanti Amith	**19BD1A1232**

ABSTRACT

Churn prediction is the process of identifying which consumers are most likely to abandon a service or cancel their membership. For many firms, this is a crucial assumption because getting new customers is generally more expensive than keeping existing ones. You should know exactly what marketing activity to do for each individual consumer once you've identified those who are on the verge of canceling. This will increase the possibilities that the customer will stay with you.

Customers' behaviors and preferences vary; therefore they cancel their memberships for a variety of reasons. It's vital, then, to connect with each of them on a regular basis in order to keep them on your client list. You must know which marketing action is most effective for each and every customer, as well as when it is most effective.

Customer churn is a problem that affects firms in a variety of industries. You must invest in recruiting new clients if you want to expand as a business. Every time a client departs, a large amount of money is lost. It is necessary to devote both time and effort to their replacement. Knowing when a client is likely to depart and offering them incentives to stay can save a company a lot of money. As a result, knowing what keeps consumers engaged is incredibly useful information, as it may aid in the development of retention strategies and the implementation of operational practices targeted at preventing customers from leaving.

This project focuses on exploratory data analysis, which results in determining which attributes from customer data have the greatest impact on customer churning, and building a machine learning model that can predict which customers are most likely to churn, which can then be used to take steps to keep the customer.

LIST OF ABBREVIATIONS

ML	Machine Learning
UI	User Interface
VS	Visual Studio
GUI	Graphical User Interface

LIST OF DIAGRAMS

LIST OF SCREENSHOTS

LIST OF SCREENSHOTS

CONTENTS

DESCRIPTION **PAGE**

CHAPTER-1

INTRODUCTION

1.1. Introduction

Most common issue that any company faces is to retain their existing customers. In every industry, one of the challenges faced by the companies is customer churning. Customer churn is the percentage of customers that have stopped using a product or a service provided by a particular company within a timeframe. It can be calculated by dividing the number of customers lost during that time frame by the number of customers the company had at the start of that time frame. That means, the lower the churn rate, the better it is for the company

There are numerous telecom companies in the market today. Every company is providing their customers various schemes, plans and discounts to retain them and reduce the threat from their competitors. Churning in the telecom industry can be of various types. A customer is considered to have churned if they port out to a competitor. A customer can also be considered to have churned within a company by switching from a higher priced plan to a lower priced plan which can be considered as tariff plan churning, or switching from a postpaid subscription to a prepaid subscription which can be considered as product churning, or changing the plan from a yearly subscription to a weekly or a monthly subscription, or reduce the usage of a subscribed service or a product.

1.2. Scope

Every minute, a huge amount of data is being generated in the telecom industry. The companies can make use of this data by extracting valuable insights from it, like predicting the customers who are likely to churn, the reasons why the customers are churning and the measures to be taken to retain them. This can be made possible by using various machine learning and data science techniques.

This project can be upgraded to recommend the suitable measures that could be taken to reduce customer churning.

1.3. Project Overview

This project can be used as a tool to predict if a customer would possibly churn in the future by analyzing the customers data and checking if the customer has characteristics similar to most of the customers who have churned in the past.

The tool being developed in this project is capable of predicting the churn status for a single customer through online processing and also a group of customers whose data can be presented in a csv file through batch processing.

1.4. Objective

The objective of this project is to help companies understand their customers and their satisfaction level with the company. This project helps the organizations in identifying potential churners and formulate certain ideas to retain the customers.

1.5. Architecture Diagram

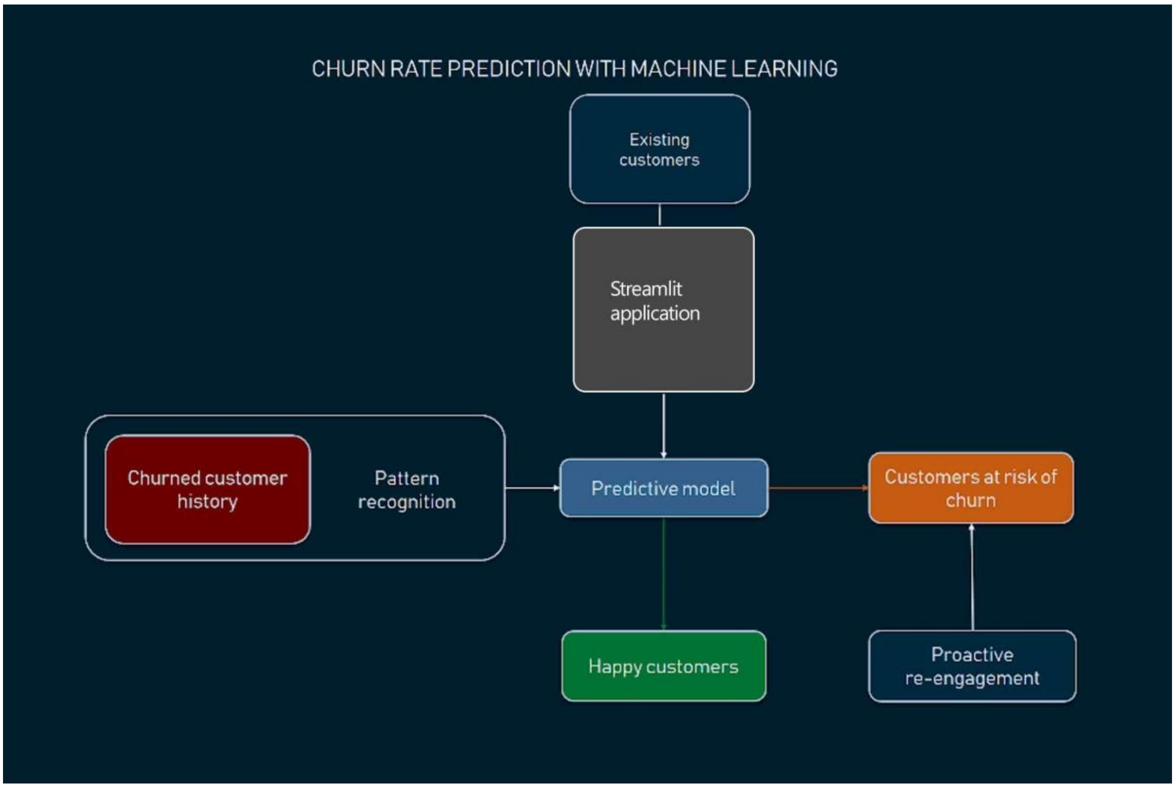


Figure 1: Architecture Diagram of the Project

CHAPTER-2

LITERATURE SURVEY

2.1 Existing System :

Despite the existence of many systems which can analyze the customer data and derive valuable insights from it like the factors affecting the customer churn, it becomes difficult to predict whether a particular customer would churn in the near future based on his characteristics in real time. The existing systems use exploratory data analysis to understand the data and the companies have to make decisions based on these insights.

2.2 Proposed System :

Though we can act upon the insights provided by the exploratory data analysis alone to take measures to retain the customers, much more effective measures can be formulated if the system is able to predict the particular customer who would potentially churn.

The predictive model needs to have a suitable machine learning algorithm to predict the churn value accurately. So it is important to check the compatibility of the algorithm being used by comparing the accuracy of the model by using different algorithms.

Thus, the solution is to train various machine learning algorithms with the customer data and compare the results and choose the best algorithm for the customer churn prediction.

Here are the some related work :

1) Xin Hu [1] proposed a system which combines the decision tree and neural network customer churn prediction models to obtain higher accuracy and better prediction of customer churn which has an accuracy of 98.87%.

2) Mohammad A. Hassonah [2] conducted comparative study on the effectiveness of Decision Tree and K-Nearest Neighbor algorithms in predicting Customer Churn rates which resulted in 86.8% accuracy and 61.5% precision for KNN algorithm and 92.6% accuracy and 77.5% precision for decision tree model.

3) Mykola Malyar [3] used the data from Prozorro system and proposed a system to evaluate customer churn rate using ensemble tree methods (Random Forest, XGBoost, LightGBM). The best results were obtained using XGBoost with an AUC score of 0.8322 whereas worst results were obtained when the decision tree model was used with an AUC score of 0.65.

4) Abinash Mishra [4] used Ensemble based Classifiers namely Bagging, Boosting and Random Forest and performed a comparative study with the known classifiers namely Decision Tree, Naïve Bayes Classifier and Support Vector Machine (SVM) in predicting Customer Churn rates and found less error rate (8.34%), low specificity (53.54%), high sensitivity (98.89%) , greater accuracy of 91.66% and 83.11% precision as compared to other methods.

5) Pushkar Bhuse [5] proposed a system for Telecom-Customer Churn Prediction using various techniques like Random Forest Classifiers and Support Vector Machine (SVM) that are compared with deep learning techniques like XGBoost and Deep Neural networks. The best results were obtained using the Random Forest with 90.06% accuracy and after grid search the accuracy increased to 91.26%

6) S. Stehani [6] proposed a system for the future churn probability by analyzing the churn customer details using the 6 popular Machine Learning techniques and finally found the Random Forest Classifier as the best with 89.0% accuracy.

7) V. Geetha [7] proposed a system with efficient algorithms like Random Forest Classifier and Support Vector Machine which increases the performance of the system and also helps in retaining the customers. By comparing both the algorithms Support Vector Machine gives 85% efficiency and the best results observed in Random Forest Classifier with 96% efficiency.

8) Qiu Yihui [8] proposed a feature selection method based on orientation ordering pruning Method (OOPM) which is more advantage than the feature selection based on Random Forest and the proposed system improves the performance of all models in predicting the Customer Churn rates which resulted in 83.5% accuracy before and 91.2% accuracy after projection for Random Forest Classifier.

9) Pan Tang [9] proposed a customer churn prediction model combining K-means and XGBoost algorithm. By comparing (k-means and logistic), (K-means and Decision tree) and (k-means and XGBoost) the best results of 81.0% accuracy, 75.0% precision, 88.0% percent F1-Score observed in combination of K-means and XGBoost

CHAPTER-3

SYSTEM REQUIREMENT SPECIFICATION

3.1. Introduction to SRS

Software Requirement Specification (SRS) is the starting point of the software developing activity. As the system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

The SRS phase consists of two basic activities:

Problem/Requirement Analysis:

The process is order and more nebulous of the two, deals with understanding the problem, the goal and constraints.

Requirement Specification:

Here, the focus is on specifying what has been found, giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity. The Requirement phase terminates with the production of the validated SRS document. Producing the SRS document is the basic goal of this phase.

3.2. Role of SRS

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

3.3. Requirements Specification Document

A Software Requirements Specification (SRS) is a document or a manual of a project provided it is prepared before you kick-start a project/application. This document is also known by the names SRS report, software document. A software document is primarily prepared for a project, software. There are a set of guidelines to be followed while preparing the software requirement specification document. This includes the purpose, scope, functional and non-functional requirements, software and hardware requirements of the project. In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.

The purpose of the SRS (Software Requirement Specification) document is to describe the external behavior of the application developed or software. It defines the operations, performance and interfaces and quality assurance requirements of the application or software. The complete software requirements for the system are captured by the SRS. This section introduces the requirement specification document for AR Shopping which lists functional as well as non-functional requirements.

3.4. Functional Requirements

For documenting the functional requirements, the set of functionalities supported by the system are to be specified. A function can be specified by identifying the state at which data is to be input to the system, its input data domain, the output domain, and the type of processing to be carried on the input data to obtain the output data.
Functional requirements define specific behavior or function of the application.
Following are the functional requirements:

1.**Website Module:** The interface for entering the data of the customer whose churn is to be predicted is designed as a web page using a streamlit module in python which is generally used to build data apps.

2. **Pre-processing Module:** In this module, we use the various python modules like pandas, numpy to perform various operations like data cleaning i.e dealing with missing values and inconsistent data, data transformation and data reduction.

In this module, exploratory data analysis is performed and the results are visualized using various data visualization modules in python like matplotlib, seaborn etc.

3**.Prediction Module**: In this module, We use the algorithm which has presented the best results out of all the algorithms taken into consideration. The Random forest algorithm has shown better accuracy compared to the other algorithms like decision tree classifier, Naive bias classifier, logistic regression, Support vector classifier. The random forest model is trained and then pickling is performed on the Model.

3.5. Non-Functional Requirements:

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. These are the constraints the System must work within. Following are the non-functional requirements:

NFR 1) Should be available at all times.

NFR 2) Should not get stuck at anypoint.

NFR 3) Hardware devices with the internet are essential for the functioning of the module.

NFR 4) Should provide security for all information collected and stored.

NFR 5) The site should load within 3 seconds.

3.6. Performance Requirements:

The capability of the device depends on the performance of the software. The app shall be able to handle any quality input ranging from 360p to 1080p Full HD of video input provided the RAM and other device specs are sufficient like space as insufficient space may create problem in installing the app or after installing it may create problem for loading or detecting the image because it will create logs but in that case the history logs can be paused. This would depend on the available memory space on the device.

3.7. Software Requirements

Technologies	:	Python, libraries : sklearn ,ML
Operating System	:	Windows 7, 10, Linux
Tools	:	VS code,Chrome

3.8. Hardware Requirements

RAM	:	4GB
Processor	:	Dual core 2.4 GHz
Architecture	:	64-bit x86, 32-bit x86 with windows/ linux
System	:	Desktop/ Laptop

3.9 Feasibility study

Technical Feasibility:

Since the technologies used are mostly open source and the hardware requirements for the system can be satisfied by any modern computer, the setup can be established with limited technical resources. This technology (machine learning) can be very much useful in the present scenario where everything is digitalized

Operational Feasibility:

As the web page provides a clear user interface and the model has to be trained single time and can be used by pickling it, the operation of this system is very user friendly. The setup required for the proposed system is secure and reliable so that there would be no data leakages.

CHAPTER-4

SYSTEM DESIGN

4.1. Introduction to UML

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from a distinctly different perspective. Each view is defined by a set of diagrams, which is as follows:

- User Model View This view represents the system from the users' perspective. The analysis representation describes a usage scenario from the end-user's perspective.
- Structural Model View In this model, the data and functionality are arrived from inside the system. This model view models the static structures. 3.
- Behavioral Model View It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.
- Implementation Model View In this view, the structural and behavioral as parts of the system are represented as they are to be built.
- Environmental Model View In this view, the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

4.2. UML Diagrams

4.2.1 Use Case Diagram

To model a system, the most important aspect is to capture the dynamic behavior. To clarify a bit in details, dynamic behavior means the behavior of the system when it is running/operating. So only static behavior is not sufficient to model a system; rather dynamic behavior is more important than static behavior. In UML there are five diagrams available to model dynamic nature and use case diagrams are one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors.So use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application.

A single use case diagram captures a particular functionality of a system. So, to model the entire system, numbers of use case diagrams are used. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So, when a system is analyzed together its functionalities, use cases are prepared and actors are identified. In brief, the purposes of use case diagrams can be as follows:

1. Used to gather requirements of a system.
2. Used to get an outside view of a system.
3. Identify external and internal factors influencing the system.
4. Showing the interacting among the requirements are actors.

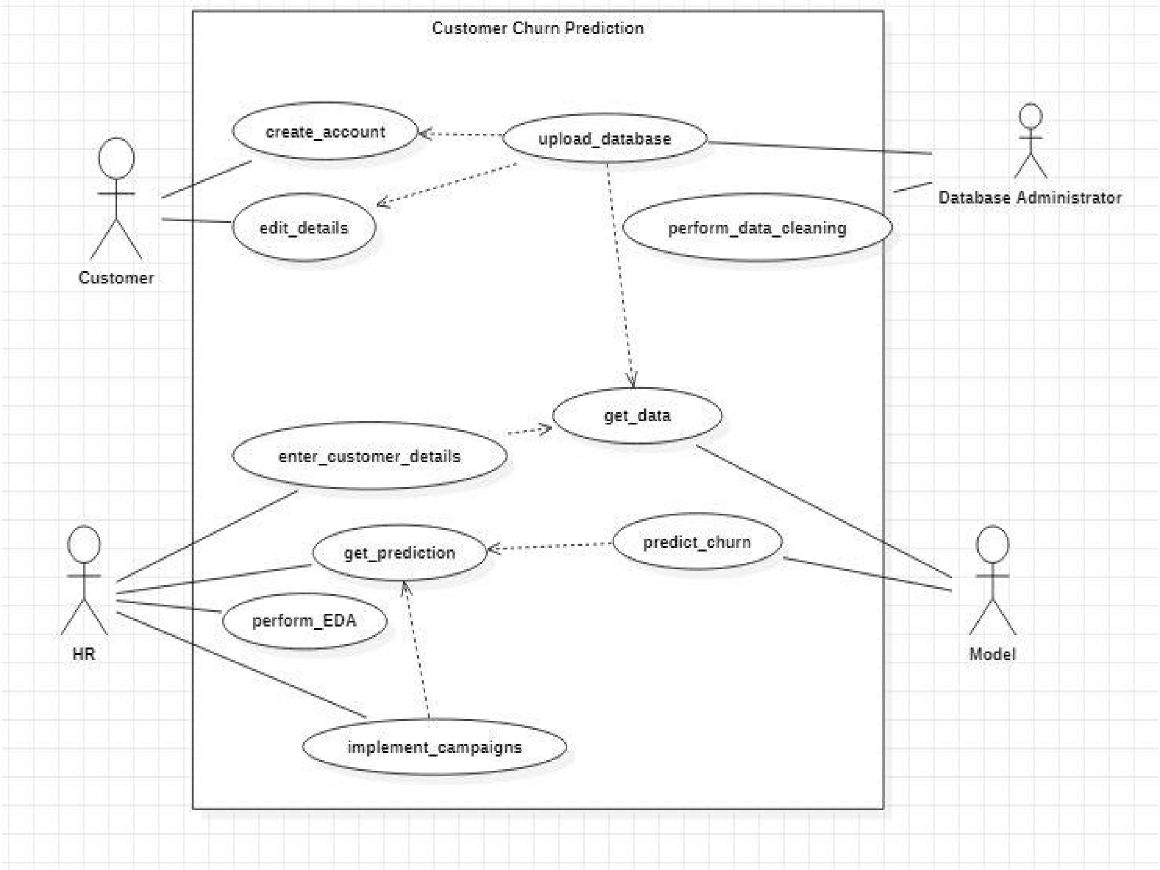


Figure 2. Use Case Diagram

4.2.2. Sequence Diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system. The aim of a sequence diagram is to define event sequences, which would have a desired outcome. The focus is more on the order in which messages occur than on the message per se. However, the majority of sequence diagrams will communicate what messages are sent and the order in which they tend to occur.

Basic Sequence Diagram Notations Class Roles or Participants

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.

Activation or Execution Occurrence

Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.

Messages

Messages are arrows that represent communication between objects. Use half arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.

Lifelines

Lifelines are vertical dashed lines that indicate the object's presence over time.

Destroying Objects

Objects can be terminated early using an arrow labeled "<< destroy >>" that points to an

X. This object is removed from memory. When that object's lifeline ends, you can place an X at the end of its lifeline to denote a destruction occurrence.

Loops :

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [].

When modeling object interactions, there will be times when a condition must be met for a message to be sent to an object. Guards are conditions that need to be used throughout UML diagrams to control flow.

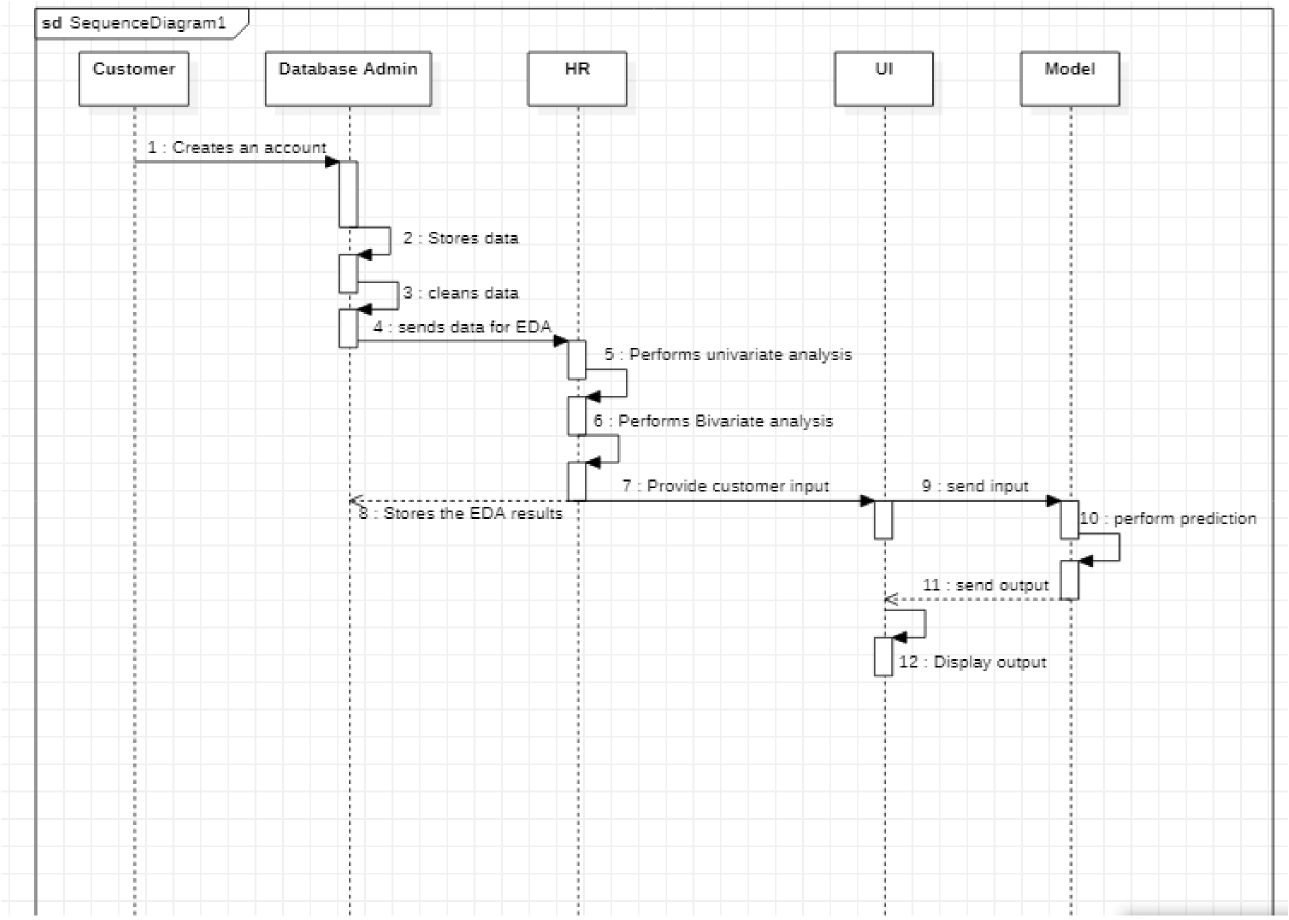


Figure 3. Sequence Diagram

4.2.3 Class Diagram

Class diagrams are the main building blocks of every object-oriented method. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. UML is standardized in class diagrams. Since classes are the building block of an application that is based on OOPs, so as the class diagram has appropriate structure to represent the classes, inheritance, relationships

The main purpose of using class diagrams are:

1. This is the only UML which can appropriately depict various aspects of OOPs.
2. Proper design and analysis of applications can be faster and efficient.
3. It is the base for deployment and component diagrams.
4. Each class is represented by a rectangle having a subdivision of three compartments:name, attributes and operation.

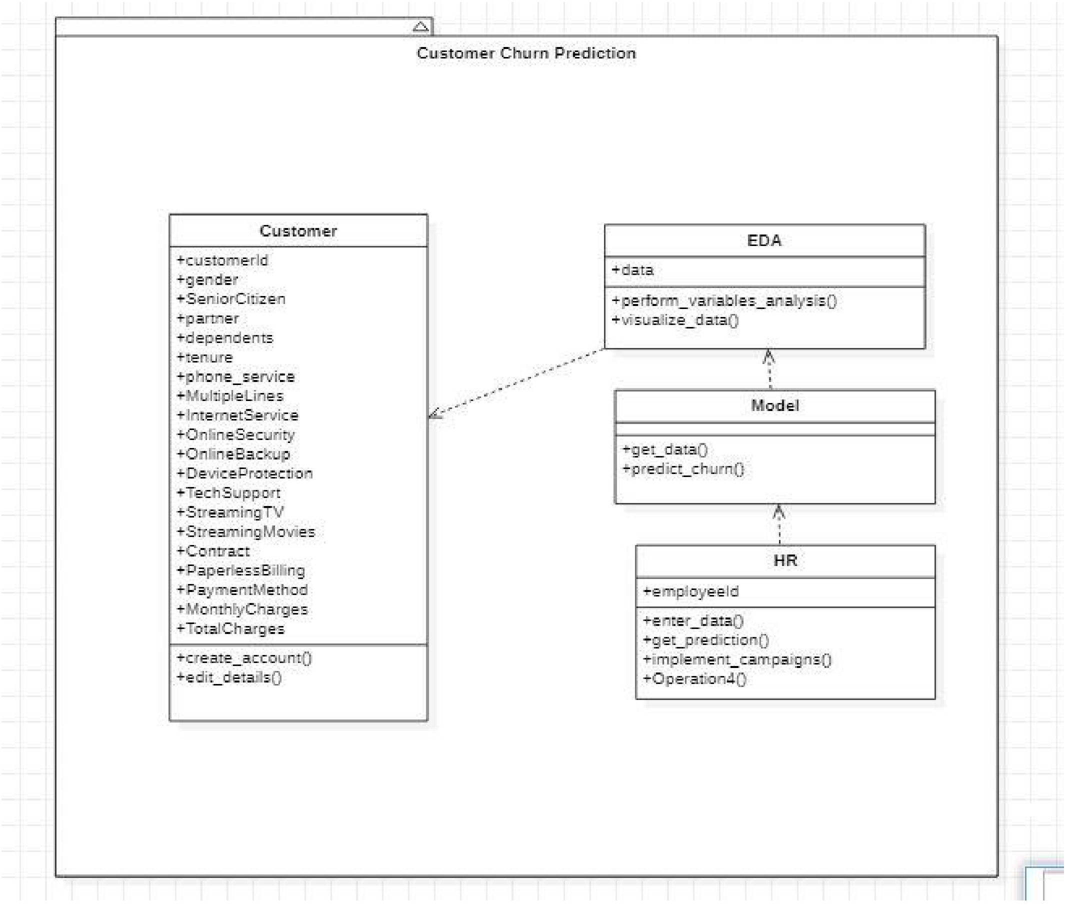


Figure 4. Class Diagram

4.2.4 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all types of flow control by using different elements such as fork, join, etc.

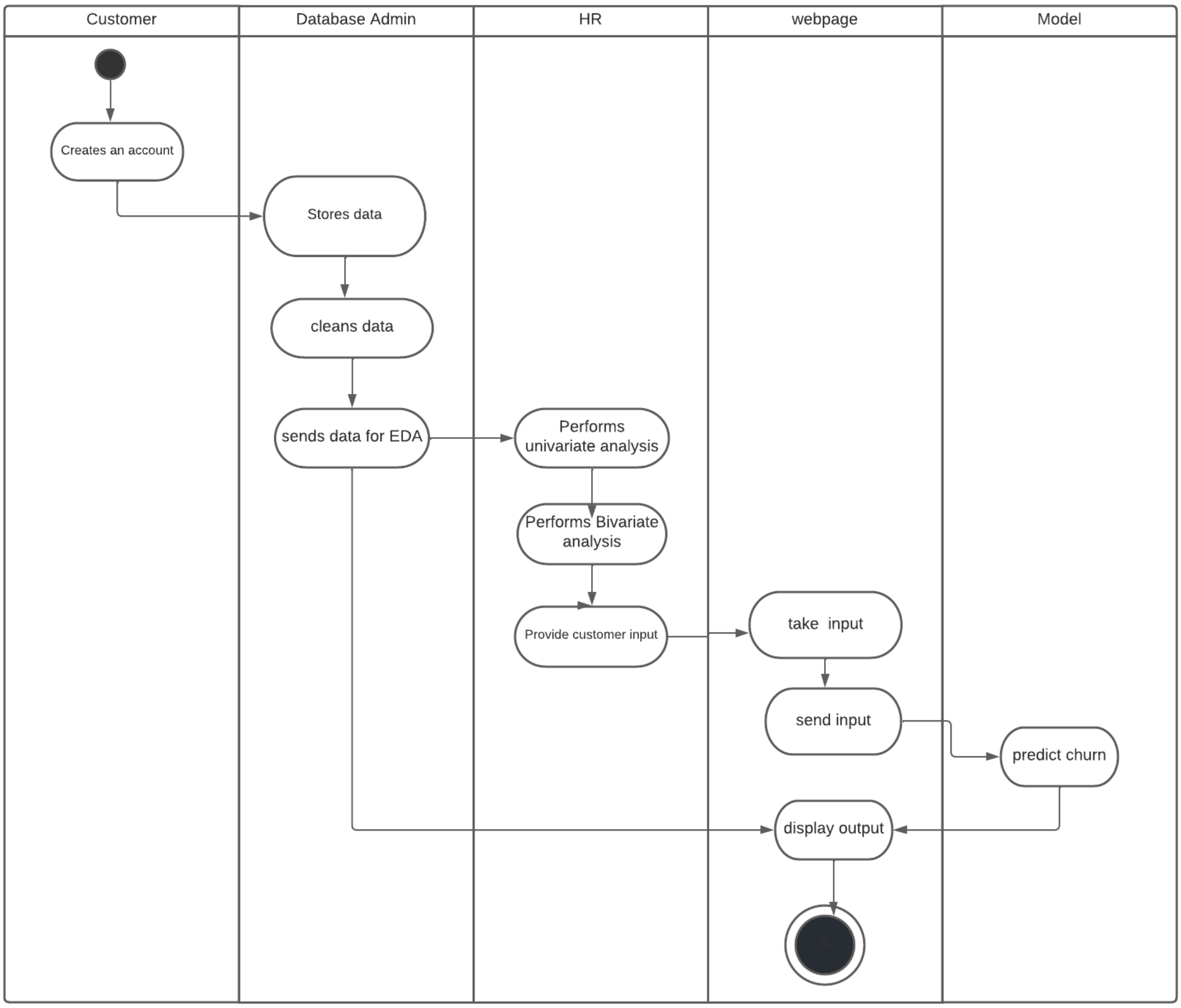


Figure 5. Activity Diagram

4.3. Technologies used:

4.3.1 Python

- Python is a high-level, interpreted, interactive and object-oriented scripting language.
- Python is designed to be highly readable.
- It uses English words frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages.

History of Python

1. Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.
2. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.
3. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).
4. Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Importance of Python

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner- level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

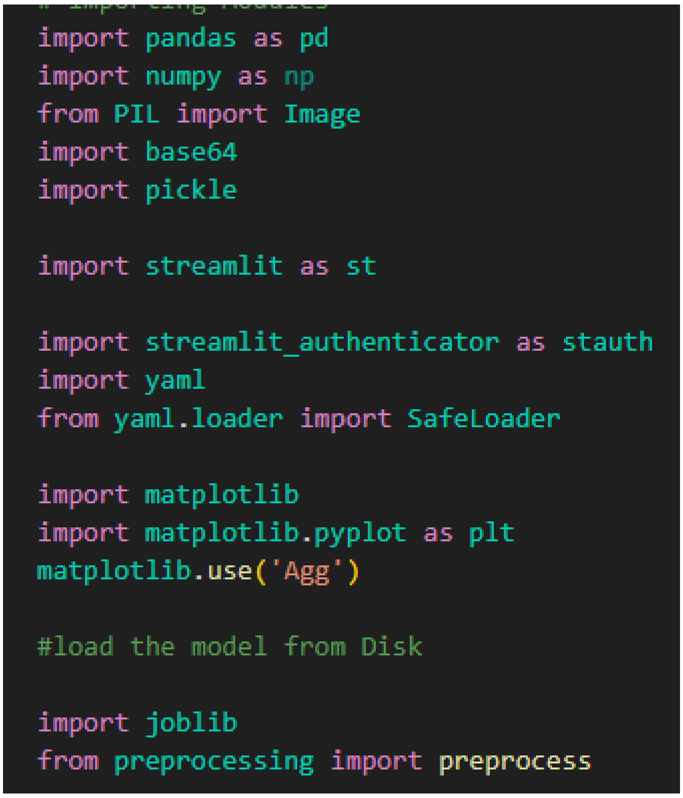
Features of Python

1. **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
2. **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
3. **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
4. **A broad standard library** − Python's bulk of the library is very portable and cross platform compatible on UNIX, Windows, and Macintosh.
5. **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
6. **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
7. **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
8. **Databases** − Python provides interfaces to all major commercial databases.
9. **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
10. **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Libraries used in python:



```
# importing modules
import pandas as pd
import numpy as np
from PIL import Image
import base64
import pickle

import streamlit as st

import streamlit_authenticator as stauth
import yaml
from yaml.loader import SafeLoader

import matplotlib
import matplotlib.pyplot as plt
matplotlib.use('Agg')

#load the model from Disk

import joblib
from preprocessing import preprocess
```

Figure 6. Libraries Used

1. PANDAS:

Pandas is a popular open-source Python library used for data manipulation and analysis. It provides powerful data structures and data analysis tools that simplify working with structured data. Pandas is widely used in data science, machine learning, and data analysis projects.

The key features of Pandas include:

- Data Structures: Pandas introduces two primary data structures: Series and DataFrame. A Series is a one-dimensional labeled array that can hold any data type. A DataFrame is a two-dimensional tabular data structure that consists of columns with labeled data. These data structures allow for efficient handling and manipulation of data, providing a flexible and intuitive way to work with structured data.

- Data Manipulation: Pandas provides a comprehensive set of functions for data manipulation tasks. It allows you to filter, slice, and reshape data, handle missing values, merge and join datasets, group and aggregate data, and perform advanced operations like pivoting and melting. Pandas makes it easy to transform and clean data, enabling efficient data preprocessing for further analysis.

- Data I/O: Pandas supports reading and writing data from various file formats, including CSV, Excel, SQL databases, and more. It simplifies the process of importing and exporting data, making it convenient to work with different data sources and integrate with other libraries or tools.

- Data Alignment: Pandas handles automatic data alignment based on labels, which means you can perform operations on datasets with different indices or columns, and Pandas will align the data correctly. This feature simplifies working with heterogeneous datasets and enables seamless computations and calculations.

- Missing Data Handling: Pandas provides flexible methods for handling missing data, allowing you to detect, filter, fill, or interpolate missing values in your datasets. It provides powerful mechanisms for data imputation, ensuring that missing data does not hinder your analysis.

- Time Series Analysis: Pandas has extensive support for time series data. It includes specialized data structures and functions for handling time series data, such as date/time indexing, resampling, shifting, rolling window calculations, and frequency conversion. Pandas makes it easy to analyze and manipulate time-based data.

- Integration with NumPy and Matplotlib: Pandas seamlessly integrates with other scientific computing libraries like NumPy and Matplotlib. NumPy arrays can be directly converted to Pandas Series or DataFrame, allowing for efficient computation and analysis.

- Matplotlib can be used to create visualizations from Pandas data structures, enabling rich and interactive data visualizations.

- Performance Optimization: Pandas is designed for efficient data processing and is built on top of highly optimized libraries like NumPy. It provides vectorized operations and uses optimized algorithms, resulting in fast and efficient data manipulation. Additionally, Pandas supports parallel processing and can leverage multicore CPUs to speed up computations on large datasets.

2. NUMPY:

NumPy (Numerical Python) is a fundamental library for scientific computing in Python. It provides a powerful array object, along with a collection of functions for efficient numerical operations. NumPy is widely used in various domains, including mathematics, physics, engineering, and data science.

The key features of NumPy include:

- N-dimensional Array: NumPy introduces the nd array object, which is a multidimensional array capable of holding elements of the same data type. Arrays can have any number of dimensions and are highly efficient for storing and manipulating large datasets. The nd array object allows for element-wise operations, slicing, indexing, and reshaping, providing a powerful tool for numerical computations.

- Mathematical Functions: NumPy provides a comprehensive set of mathematical functions for array manipulation. It includes basic mathematical operations (addition, subtraction, multiplication, division), exponential and logarithmic functions, trigonometric functions, statistical functions, and more. These functions are designed to work efficiently with large arrays, making complex mathematical computations fast and convenient.

- Broadcasting: NumPy enables broadcasting, a powerful mechanism for performing operations on arrays with different shapes. Broadcasting allows arrays of different sizes to be combined or operated upon element-wise without the need for explicit loops. This feature simplifies and accelerates computations by eliminating the need for unnecessary array copies or transformations.

- Array Manipulation: NumPy provides functions for manipulating arrays, such as reshaping, transposing, splitting, and joining. These operations allow for easy reorganization and transformation of data, enabling efficient data preprocessing and formatting.

- Linear Algebra Operations: NumPy offers a rich set of linear algebra functions, including matrix multiplication, matrix decomposition (e.g., LU, QR, SVD), eigenvalue computation, and solving linear equations. These functions facilitate numerical computations in linear algebra and provide a foundation for many machine learning algorithms.

- Random Number Generation: NumPy includes functions for generating random numbers from various probability distributions. These functions are useful for simulation, modeling, and generating random data for testing and analysis purposes.

- Integration with other Libraries: NumPy integrates seamlessly with other scientific computing libraries in Python. It serves as the foundation for many higher-level libraries and tools, such as Pandas (for data manipulation), Matplotlib (for data visualization), and SciPy (for scientific computations). NumPy arrays can be easily passed to and processed by these libraries, enabling a seamless workflow for data analysis and scientific computing.

- Performance Optimization: NumPy is implemented in C and provides highly optimized functions and algorithms, resulting in fast and efficient numerical computations. The array-oriented nature of NumPy allows for vectorized operations, avoiding the need for explicit loops and enhancing performance. Additionally, NumPy provides mechanisms for parallel computing, enabling efficient utilization of multi-core processors.

3. PIL

PIL (from PIL import Image): The Python Imaging Library (PIL) is a library for opening, manipulating, and saving various image file formats. It provides a wide range of functions and methods to perform operations like resizing, cropping, rotating, applying filters, and converting between different image formats.

4. BASE64

base64: The base64 module includes functions to encode and decode binary data using Base64 encoding. Base64 encoding converts binary data into a text format, allowing it to be easily transmitted or stored. It is commonly used for embedding images or binary data within text-based formats like HTML or JSON.

5. PICKLE

pickle: The pickle module is used for object serialization and deserialization in Python. It allows you to convert Python objects into a stream of bytes, which can be saved to a file or transmitted over a network. The serialized objects can later be reconstructed from the byte stream, enabling the preservation and transfer of complex Python objects.

6. STREAMLIT

Streamlit is an open-source Python framework that simplifies the process of building interactive web applications for data science and machine learning tasks. It provides a clean and intuitive API that allows developers to create web-based data visualizations, dashboards, and machine learning applications without needing expertise in web development.

The key features of Streamlit are -

- simplicity,
- ease of use,
- fast prototyping capabilities.

With just a few lines of code, you can create a basic web application and have it up and running in no time. Streamlit eliminates the need for boilerplate code and complex configurations, making it accessible to both beginners and experienced developers.

- Streamlit follows a declarative programming paradigm, where the application's interface is defined by writing code. The code is executed from top to bottom, and any changes made to the code are immediately reflected in the web application, allowing for real-time updates. This feature makes it easy to iterate and experiment with different visualizations and models.

- To start building a Streamlit application, you typically import the Streamlit library (import streamlit as st) and define the different elements of your application using Streamlit's API. For example, you can add text, images, interactive widgets, and plots to your application by simply calling the corresponding functions provided by Streamlit.

- Streamlit provides a wide range of built-in widgets that allow users to interact with the application. These widgets include sliders, dropdowns, checkboxes, and buttons, among others. By leveraging these widgets, you can create dynamic and interactive components that respond to user inputs, making your application more engaging and user-friendly.

- In addition to its core functionality, Streamlit offers various extensions and integrations that enhance its capabilities. For instance, you can integrate Streamlit with popular data manipulation and analysis libraries like Pandas and NumPy. You can also leverage other visualization libraries such as Matplotlib, Plotly, or Altair to create rich and interactive plots within your Streamlit application.

- Streamlit also supports custom theming and styling options, allowing you to customize the appearance of your application to match your desired design. You can choose from different predefined themes or create your own custom theme using CSS.

- Deployment of Streamlit applications is straightforward. You can deploy Streamlit apps on various platforms, including local servers, cloud platforms, or containerized environments like Docker. Streamlit applications can be easily shared with others by sharing the URL or deploying them on cloud services like Heroku or AWS.

- Streamlit also supports collaboration through sharing and versioning your Streamlit application code using Git or other version control systems. This enables multiple developers to work together on the same application, making it suitable for team projects.

- Overall, Streamlit simplifies the process of creating web applications for data science and machine learning tasks by providing an easy-to-use and intuitive framework. It enables developers to focus on building the core functionality of their applications, saving time and effort on web development intricacies. With its real-time updates, interactive widgets, and seamless deployment options, Streamlit empowers data scientists and developers to create compelling and interactive web applications with ease.

7. STREAMLIT-AUTHENTICATOR

Streamlit Authenticator is a module designed to provide authentication functionality for Streamlit applications. It is a custom module built on top of the Streamlit framework that allows you to secure your Streamlit applications by implementing user management, login/logout functionality, and access control.

Authentication is an essential aspect of many web applications, especially when dealing with sensitive data or providing personalized experiences to users. Streamlit Authenticator helps you add this security layer to your Streamlit applications without the need for complex setup or extensive knowledge of authentication protocols.

With Streamlit Authenticator, you can implement user authentication by defining user accounts and managing user credentials. This module enables you to create user registration forms to allow users to sign up for accounts and securely store their credentials. It also provides a login form that verifies user credentials and grants access to protected sections of your Streamlit application.

Streamlit Authenticator supports various authentication mechanisms, such as username/password authentication or integration with external authentication providers like OAuth or LDAP. This flexibility allows you to choose the authentication method that best suits your application's needs and integrates seamlessly with your existing infrastructure.

Access control is another crucial aspect of securing Streamlit applications. With Streamlit Authenticator, you can define different user roles or groups and assign specific permissions to each role. This allows you to control which parts of your application users can access based on their roles. For example, you can create an admin role with full access to all features and a regular user role with limited access.

Streamlit Authenticator also provides features for managing user sessions and implementing logout functionality. It handles session management, ensuring that users remain authenticated during their session and automatically logging them out after a certain period of inactivity or upon explicit logout.

In addition to authentication and access control, Streamlit Authenticator offers features for user management, such as user profile management, password reset functionality, and account activation. These features enhance the user experience and provide necessary tools for administrators to manage user accounts effectively.

Integrating Streamlit Authenticator into your Streamlit application is straightforward. You can import the module, define the authentication and authorization settings, and utilize the provided authentication and authorization functions within your Streamlit code. Streamlit Authenticator seamlessly integrates with the Streamlit framework, allowing you to focus on building your application's core functionality while ensuring the security of your users and data.

8. YAML

yaml (import yaml): YAML (YAML Ain't Markup Language) is a human-readable data serialization format. The yaml module allows you to load YAML data into Python objects and convert Python objects into YAML data. It is commonly used for configuration files, data storage, and interchanging data between different programming languages.

yaml.loader and SafeLoader: The yaml.loader module provides loaders for reading YAML data, and SafeLoader is a secure loader that constructs Python objects from YAML data. It ensures that only safe and expected data structures are loaded, protecting against arbitrary code execution.

9. MATPLOTLIB

Matplotlib's pyplot module is a powerful component of the Matplotlib library that provides a simple and intuitive interface for creating various types of plots and visualizations in Python. It is widely used for data visualization and is particularly helpful for generating static, interactive, and publication-quality plots.

Pyplot provides a set of functions and methods that allow you to create figures, axes, and plots in a straightforward manner. It abstracts many of the complexities of creating plots and provides a high-level API that makes it easy to create common types of visualizations.

With pyplot, you can create line plots, scatter plots, bar charts, histograms, pie charts, box plots, and many other types of plots. You can customize the appearance of your plots by adding labels, titles, legends, gridlines, and annotations. Pyplot also provides functions for setting axes limits, adjusting the aspect ratio, and configuring plot styles.

One of the key features of pyplot is its ability to work seamlessly with NumPy arrays and Pandas data structures. You can pass arrays or data frames directly to the plotting functions, eliminating the need for manual data manipulation. Pyplot automatically handles the data transformation and visualization, allowing you to focus on analyzing and presenting your data.

Pyplot supports various backends, allowing you to save plots to different file formats (e.g., PNG, PDF, SVG) or display them interactively in a graphical window. It can be integrated with Jupyter notebooks, making it convenient for interactive data exploration and analysis.

In addition to its core functionality, pyplot provides a wide range of customization options. You can control the colors, line styles, markers, and other visual attributes of your plots. Pyplot also supports subplots, allowing you to create multiple plots within a single figure, facilitating the comparison and visualization of multiple datasets.

Another notable feature of pyplot is its support for advanced plot types, such as 3D plots, contour plots, and heatmaps. These plots are useful for visualizing complex relationships or spatial data.

Pyplot is highly extensible, and you can further enhance its capabilities by using other Matplotlib components, such as the Axes module, which provides fine-grained control over plot elements, or the ColorMap module, which allows you to define custom color maps.

Overall, pyplot simplifies the process of creating plots and visualizations in Python. Its intuitive interface, extensive customization options, and compatibility with other scientific computing libraries make it a popular choice for data visualization tasks. Whether you need to create basic line plots or intricate visualizations, pyplot provides a versatile and powerful toolset for effective data presentation and exploration

matplotlib.use('Agg'): This line of code sets the backend of Matplotlib to 'Agg'. The backend determines how Matplotlib renders the plots. 'Agg' is a non-interactive backend that is commonly used in web applications or server environments where interactive plots are not needed. It allows Matplotlib to render plots as static images.

10. JOBLIB

joblib (import joblib): Joblib is a library for efficient serialization of Python objects, particularly NumPy arrays. It provides functions to dump and load Python objects to/from disk, allowing you to save trained machine learning models or large data structures efficiently. Joblib leverages the pickle module but optimizes the serialization process for improved performance.

preprocessing (from preprocessing import preprocess): This module is likely a custom module that contains functions or classes for data preprocessing. Data preprocessing involves transforming raw data into a suitable format for analysis or machine learning. It may include tasks like data cleaning, feature scaling, handling missing values, encoding categorical variables, and more. The specific functionality of the preprocess function would depend on its implementation in the module.

4.4. Website Module:

The Website Module serves as the user interface for the app and is developed using the Streamlit framework. It provides a user-friendly interface for collecting customer data. Users can interact with the app by inputting various customer attributes such as demographics, purchase history, and usage patterns. The module ensures a seamless user experience by using interactive forms and widgets to gather the necessary information.

4.5. Pre-Processing Module

The Pre-processing Module is responsible for preparing the collected customer data for analysis and prediction. It utilizes Python modules such as Pandas and NumPy to perform several data operations. These operations include data cleaning, where missing values and inconsistent data are handled. Missing values can be imputed or removed, and inconsistent data is corrected to ensure data consistency.

The module also performs data transformation to convert categorical variables into numerical representations using techniques like one-hot encoding or label encoding. Numeric variables may be scaled or normalized to bring them to a comparable range. Additionally, feature engineering techniques are applied to create new features or modify existing ones. This can involve deriving new features from existing attributes or extracting relevant information from date/time variables.

4.6. Prediction Module

The Prediction Module utilizes a trained prediction model to estimate the likelihood of customer churn based on the pre-processed data. After evaluating various algorithms, such as decision tree classifier, Naive Bayes classifier, logistic regression, and support vector classifier, the Random Forest algorithm is selected for its superior accuracy.

The Random Forest model is trained using the pre-processed customer data. This ensemble learning algorithm combines multiple decision trees to make predictions. The trained model is then serialized using pickling, allowing it to be stored and reused for future prediction.

When a new customer's data is provided through the website module, the pre-processed data is fed into the trained Random Forest model to predict the likelihood of customer churn. The prediction module generates the churn prediction result, which can be displayed to the user through the website module's interface.

The inclusion of the CatBoost Classifier algorithm enhances the preprocessing module. CatBoost is a gradient boosting algorithm that automatically handles categorical variables without explicit encoding. It simplifies the preprocessing stage by effectively handling categorical features and potentially improving the performance of the subsequent prediction model.

By incorporating these modules into the Customer Churn Prediction App, businesses can gather customer data efficiently, preprocess it effectively with feature engineering and CatBoost Classifier, and leverage the power of the Random Forest algorithm for accurate churn predictions.

CHAPTER-5

Installation of VS Code :

This should be the initial step when you are building an end to end project. VS implies Visual Studio Code.Visual Studio Code is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS.Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including C#, Java, JavaScript, Go, Node.js, Python, C++, C

INSTALL VS CODE

Go to the below mentioned site and download vs code. It's an open source which is freely available Windows,MAC,LINUX.
https://code.visualstudio.com/download

1. Download the Visual Studio Code installer for Windows.
2. Once it is downloaded, run the installer (VSCodeUserSetup-{version}.exe). This will only take a minute.
3. By default, VS Code is installed under C:\Users\{Username}\AppData\Local\Programs\Microsoft VS Code .

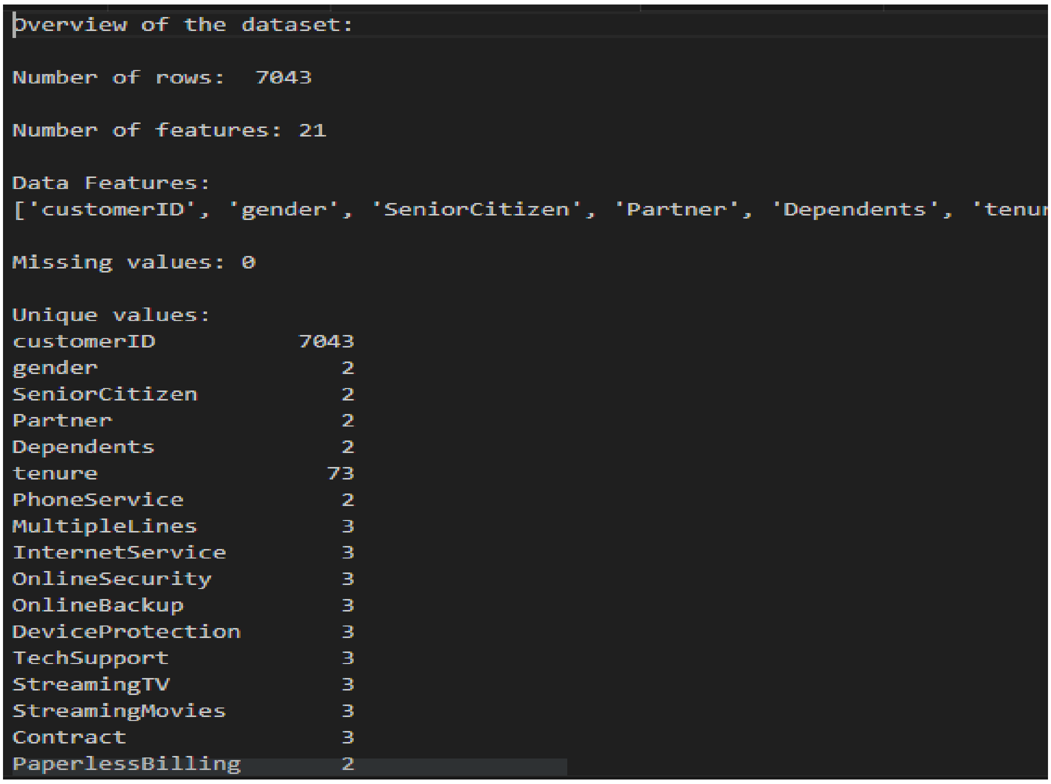
5.1 Module 1 - Data Preprocessing

Import all the essential libraries that will be needed for this project.

```
import numpy as np

import pandas as pd

import Seaborn as sns

import matplotlib.ticker as mtick

import matplotlib.pyplot as plt

%matplotlib inline
```

STEP 2 : Loading the data file Code

```
def data_overiew(df, message):

    print(f'{message}:\n')

    print('Number of rows: ', df.shape[0])

    print("\nNumber of features:", df.shape[1])

    print("\nData Features:")

    print(df.columns.tolist())

    print("\nMissing values:", df.isnull().sum().values.sum())

    print("\nUnique values:")

    print(df.nunique())

data_overiew(data_df, 'Overview of the dataset')
```



```
Overview of the dataset:

Number of rows:  7043

Number of features: 21

Data Features:
['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenur

Missing values: 0

Unique values:
customerID          7043
gender                 2
SeniorCitizen          2
Partner                2
Dependents             2
tenure                73
PhoneService           2
MultipleLines          3
InternetService        3
OnlineSecurity         3
OnlineBackup           3
DeviceProtection       3
TechSupport            3
StreamingTV            3
StreamingMovies        3
Contract               3
PaperlessBilling       2
```

Figure 7. Data Overview

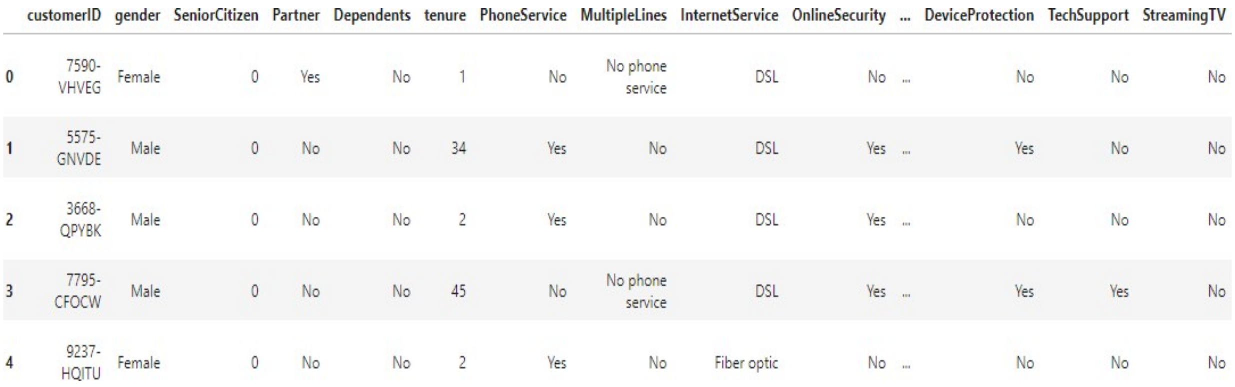
Data preprocessing transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks.

The techniques are generally used at the earliest stages of the machine learning and AI development pipeline to ensure accurate results. Processing of data includes loading the data , data pre processing, fitting the model , predicting the outcome.

Loading data is done by a method called read_csv(file_name)

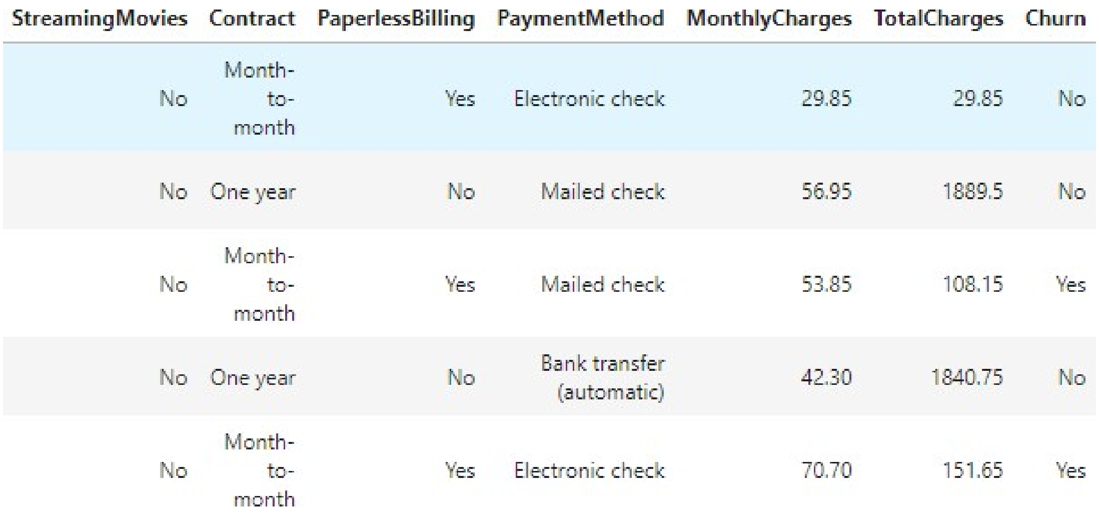
```
telco_base_data =  pd.read_csv
('WA_Fn-UseC_-Telco- CustomerChurn.csv')
 telco_base_data.head()
```

Output:



	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	No
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	No
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No	No

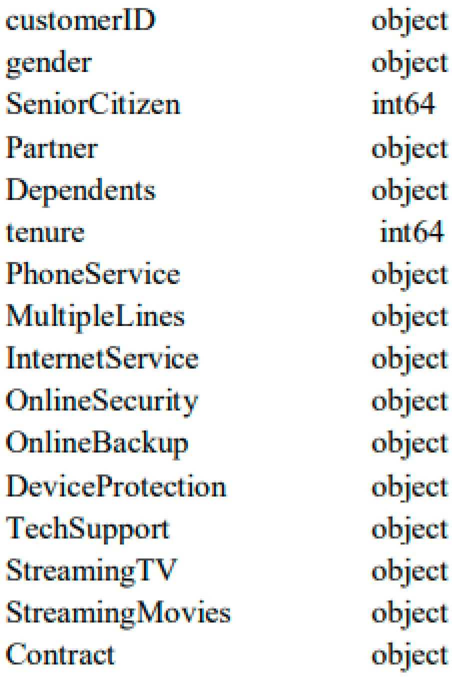
Figure 8. Table of Output Displayed on successful retrieval of Data i.e. from .csv



StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
No	Month-to-month	Yes	Electronic check	29.85	29.85	No
No	One year	No	Mailed check	56.95	1889.5	No
No	Month-to-month	Yes	Mailed check	53.85	108.15	Yes
No	One year	No	Bank transfer (automatic)	42.30	1840.75	No
No	Month-to-month	Yes	Electronic check	70.70	151.65	Yes

Figure 8. Table of Output Displayed

The dataset contains a total of 7043 rows and 21 columns. The attributes of the data and their respective data types can be shown as follows



```
customerID          object
gender              object
SeniorCitizen       int64
Partner             object
Dependents          object
tenure               int64
PhoneService        object
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies     object
Contract            object
```

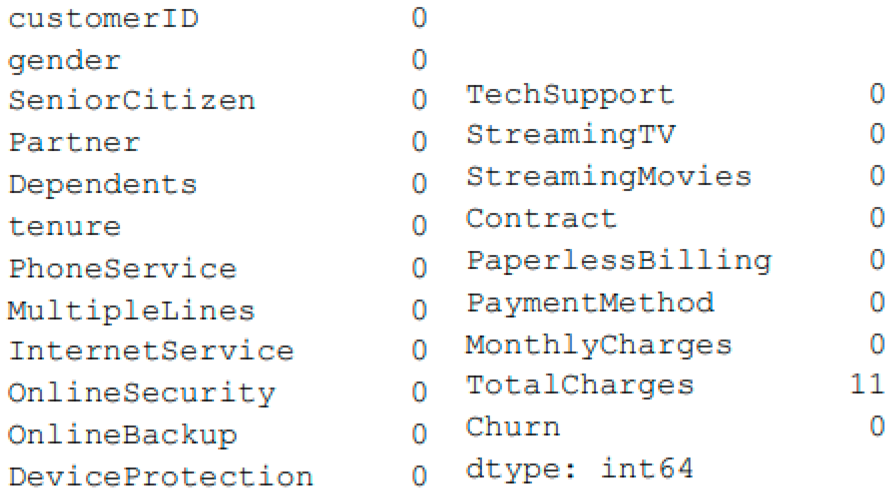
Data Cleaning :

Data cleaning refers to the process of dealing with missing data and inconsistent data so that the dataset remains clean and doesn’t affect the performance of the prediction model in a negative way. Generally, there are a few thumb rules to deal with missing data. If the number of records containing missing values are less, they can be filled out using regression methods or by calculating the arithmetic mean of the values of that field.For the features with high number of missing values, dropping the records containing missing values proves as an effective measure as they tend to give very less insights on the data and there exists a chance that the missing data might mislead the conclusions and give inaccurate insights. The decisions to be taken about what to do with the missing data should be taken carefully because the missing data also can hide some information within it and this depends on the particular scenario in consideration. The decisions should be taken considering the domain knowledge.

The total charges column must be of numeric type. So, we have converted the column to numeric type and checked for missing values.

Code :

```
telco_data.TotalCharges=pd.to_numeric(telco_data.TotalCharges,errors='coerce') telco_data.isnull().sum()
```



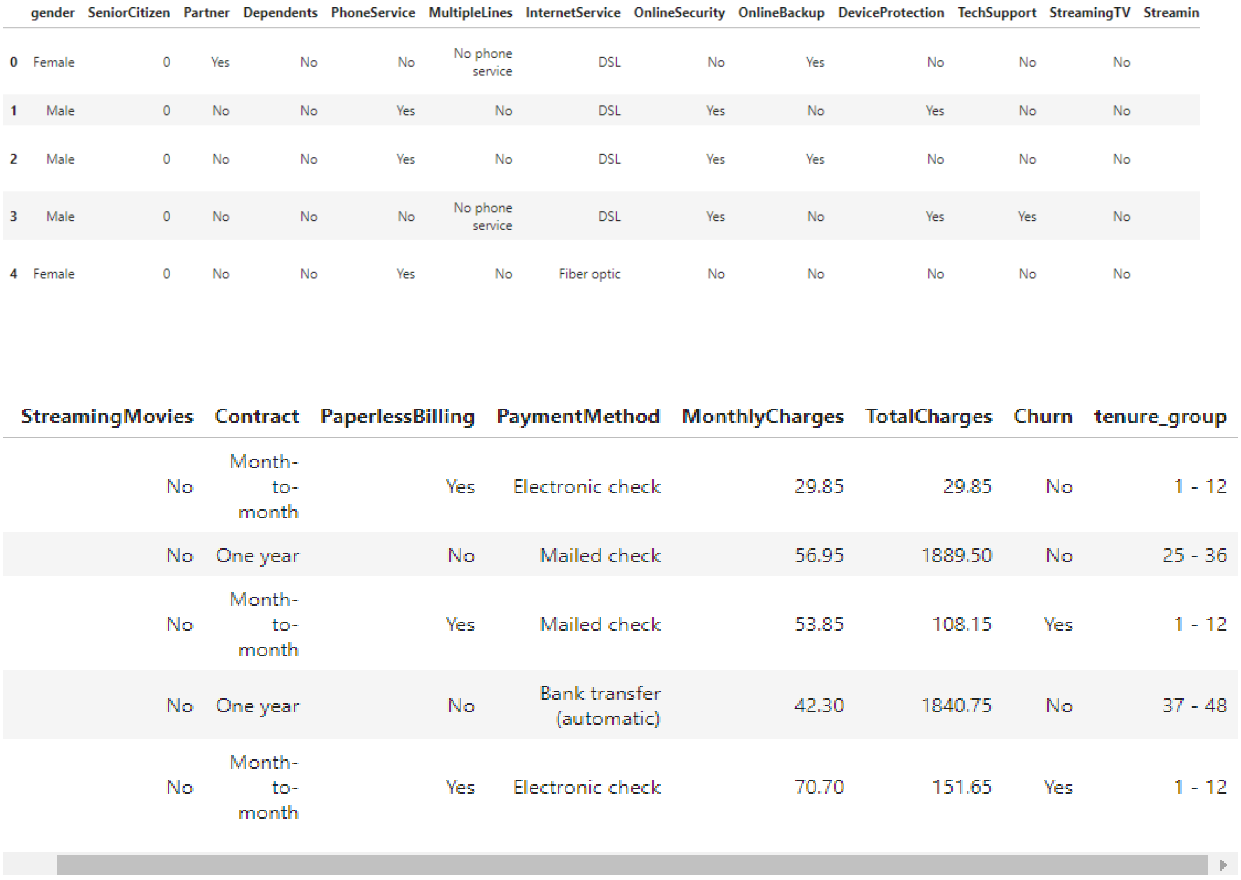
```
customerID          0
gender              0
SeniorCitizen       0   TechSupport          0
Partner             0   StreamingTV          0
Dependents          0   StreamingMovies      0
tenure              0   Contract             0
PhoneService        0   PaperlessBilling     0
MultipleLines       0   PaymentMethod        0
InternetService     0   MonthlyCharges       0
OnlineSecurity      0   TotalCharges        11
OnlineBackup        0   Churn                0
DeviceProtection    0   dtype: int64
```

This means there are 11 missing values in the field Total Charges. These records should be removed. Also, columns like customerID are not required for the analysis, so they are dropped.

It is easier to analyze the data when all the fields having numerical values are converted into categorical values. These fields are further subjected to a process called one hot encoding in which each possible value of the column is considered as a new column and these columns have boolean values.

Code :

```
telco_data.drop(columns=['customerID'],                axis=1,
inplace=True) telco_data.head();
```



	gender	SeniorCitizen	Partner	Dependents	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	Streamin
0	Female	0	Yes	No	No	No phone service	DSL	No	Yes	No	No	No	
1	Male	0	No	No	Yes	No	DSL	Yes	No	Yes	No	No	
2	Male	0	No	No	Yes	No	DSL	Yes	Yes	No	No	No	
3	Male	0	No	No	No	No phone service	DSL	Yes	No	Yes	Yes	No	
4	Female	0	No	No	Yes	No	Fiber optic	No	No	No	No	No	

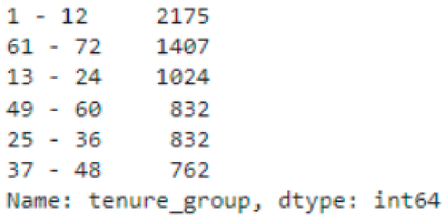
StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn	tenure_group
No	Month-to-month	Yes	Electronic check	29.85	29.85	No	1 - 12
No	One year	No	Mailed check	56.95	1889.50	No	25 - 36
No	Month-to-month	Yes	Mailed check	53.85	108.15	Yes	1 - 12
No	One year	No	Bank transfer (automatic)	42.30	1840.75	No	37 - 48
No	Month-to-month	Yes	Electronic check	70.70	151.65	Yes	1 - 12

Figure 9. Table of Output of dropping Irrelevant data

Code:

```
labels = ["{0} - {1}".format(i, i + 11) for i in range(1,
72, 12)]

telco_data['tenure_group'] = pd.cut(telco_data.tenure,
range(1, 80, 12), right=False, labels=labels)

telco_data['tenure_group'].value_counts()
```

Output:



```
1 - 12     2175
61 - 72    1407
13 - 24    1024
49 - 60     832
25 - 36     832
37 - 48     762
Name: tenure_group, dtype: int64
```

The tenure field has been converted into a tenure_group field. So we no longer need the tenure field. So it is also dropped.

Code:

```
telco_data.drop(columns= ['tenure'], axis=1, inplace=True)
```

5.2 Module 2 - Data Exploration

Exploratory data analysis is performed to understand the data and extract the patterns hidden in the data which can provide valuable insights which can help the companies understand the common characteristics among churners.

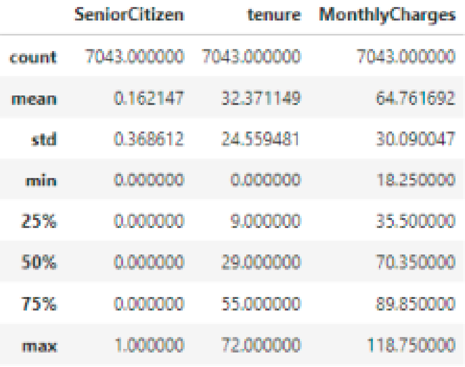
5.2.1 Data Description

This data exploration can be started by considering the original dataset and finding out the statistical description of the data. This is done as follows:

Code:

```
telco_base_data.describe()
```

Output:



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

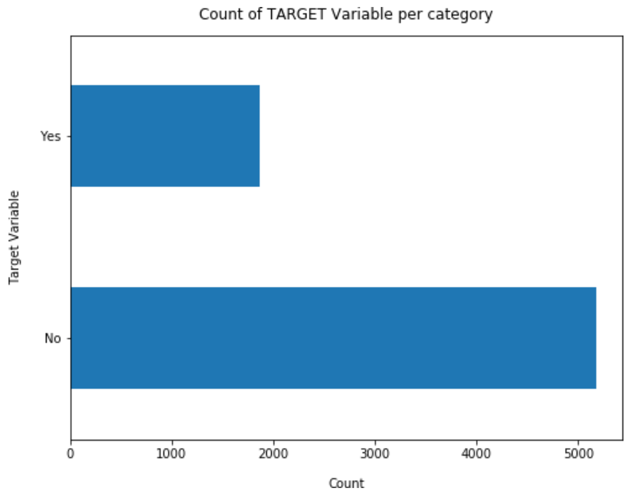
Figure 10. Table of Data Description

5.2.2 Data Distribution

As the target variable churn is a categorical variable and has values “Yes" and “No” only, we can find out the distribution of the data among churners and non churners which would give us an idea about whether the dataset is balanced or not. The distribution of data can be expressed in the following ways.

Code :

```
telco_base_data['Churn'].value_counts().plot(kind='barh',
figsize=(8, 6)) plt.xlabel("Count", labelpad=14)
plt.ylabel("Target Variable", labelpad=14) plt.title("Count
of TARGET Variable per category", y=1.02);
```



Output:

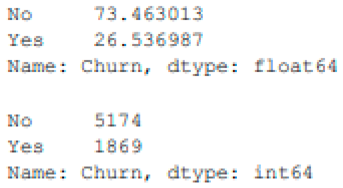
Figure 11. Table of Data Description

The above bar plot shows the distribution on the dataset among churners and non churners. This shows that the dataset is hugely imbalanced. The number of non churners is much greater than the number of churners in the given dataset.

Code :

```
100*telco_base_data['Churn'].value_counts()/len(telco_base_
data

                ['Churn'])
telco_base_data['Churn'].value_counts()
```

Output:



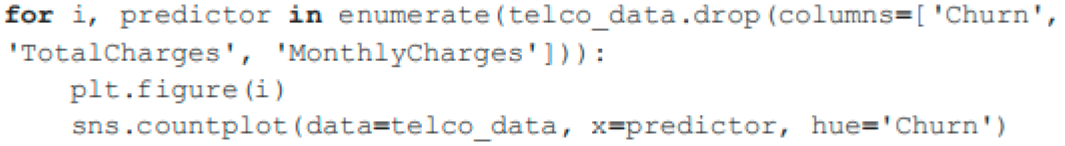
```
No     73.463013
Yes    26.536987
Name: Churn, dtype: float64

No     5174
Yes    1869
Name: Churn, dtype: int64
```

The above output shows the percentage of non churners to churners and the count of number of non churners and number of churners.

5.2.3 Performing Univariate analysis

Univariate analysis shows the relationship of each variable with the target variable. Uni means one, so it means that only one variable is being considered. It is the most straightforward form of data analysis. It is performed to describe, summarize and find patterns in the data.

Code:



```
for i, predictor in enumerate(telco_data.drop(columns=['Churn',
'TotalCharges', 'MonthlyCharges'])):
    plt.figure(i)
    sns.countplot(data=telco_data, x=predictor, hue='Churn')
```

Output:

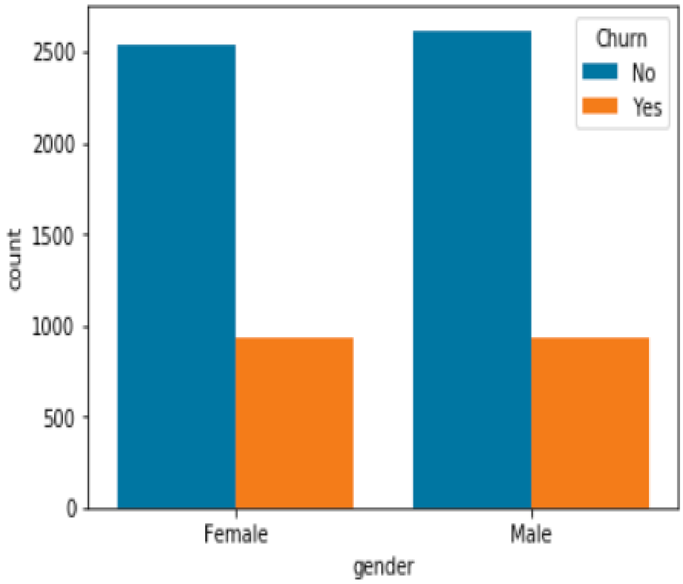


Figure 12.1 Univariate Analysis Factors

The above bar plot shows the distribution of corners and non churners with respect to gender. It can be observed that gender doesn't have a significant effect on the churn value. So, gender attributes can be considered unimportant in predicting churn for a customer. However, this only indicates that gender is not an important feature when only the gender variable is considerable. It can prove to be useful when a bivariate analysis is performed with gender as one of the variables.

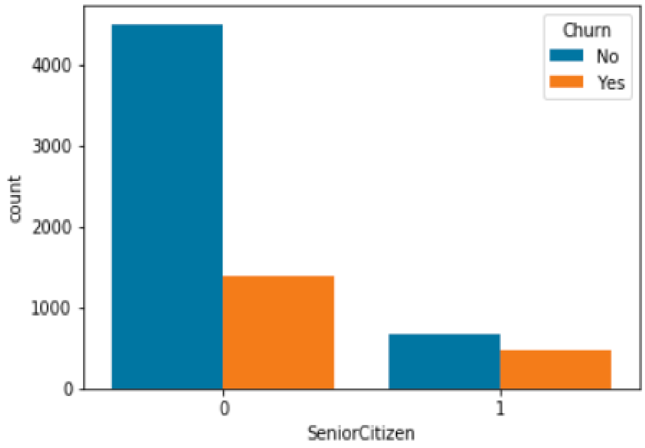


Figure 12.2 Univariate Analysis

The above bar plot shows the distribution of churners and non churners with respect to whether the customer is a senior citizen or not. It can be clearly observed that the people who are not senior citizens have churned more than the pople who are senior citizens. This observation can be explained by assuming that as the senior citizens are generally less aware of technology, they tend not to go through all the hectic process of changing from one company to another company. Thus, the senior citizen attribute can help in prediction of churn.

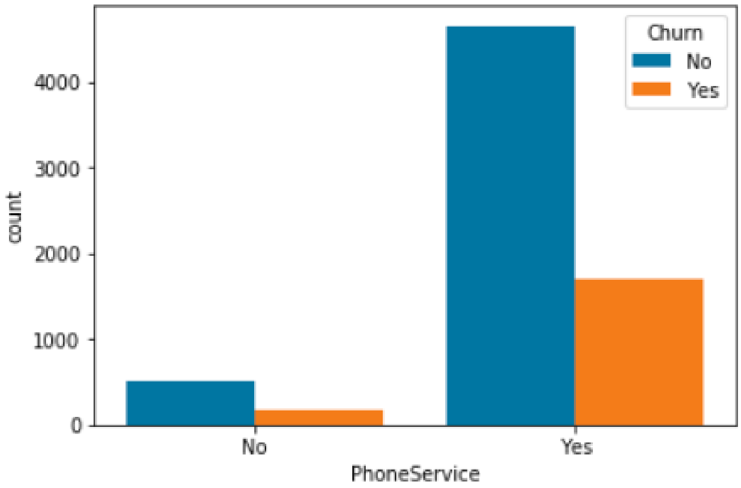


Figure 12.3 Univariate Analysis

The above bar plot represents the distribution of churners and non churners with respect to the phone service attribute. The bar plot indicates that the people who have opted for the phone service have churned more than the people who have not opted for the phone service. So, the phone service attribute is considered to be positively correlated with the churn variable.

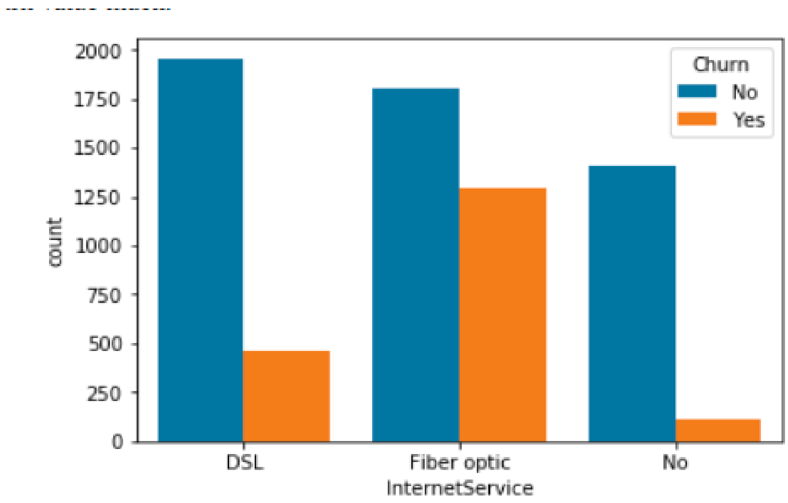


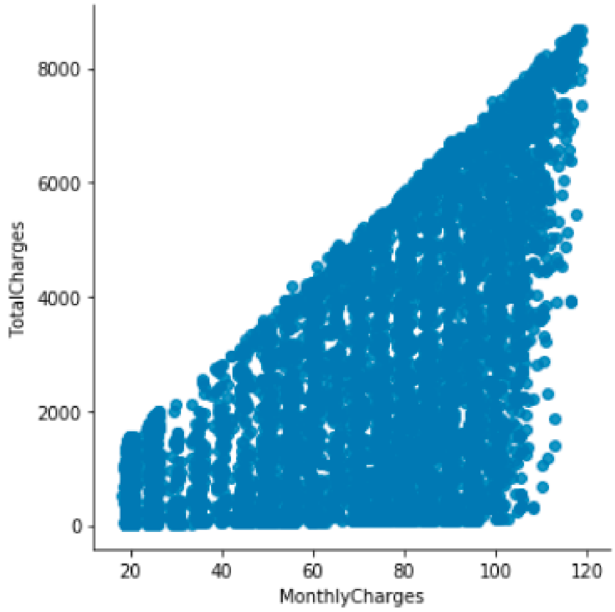
Figure 12.4 Univariate Analysis

The above bar plot represents the distribution of churns and non churners with respect to the kind of internet service they have opted for. It can be observed that the customers who had Fiber Optic Internet service seem to churn more than the other types. So this is a valuable insight in finding the churn for a new customer.

And We need to Perform Univariate Analysis Like How the Each and Every Variable in the Input is going to Interpret The outcomes

The correlation among other attributes is checked as follows:

Figure 13 Co–relation between factors after Univariate Analysis



The results match with expectations. The value of total charges increases as monthly charges increase. This means total charges and monthly charges variables are positively correlated.

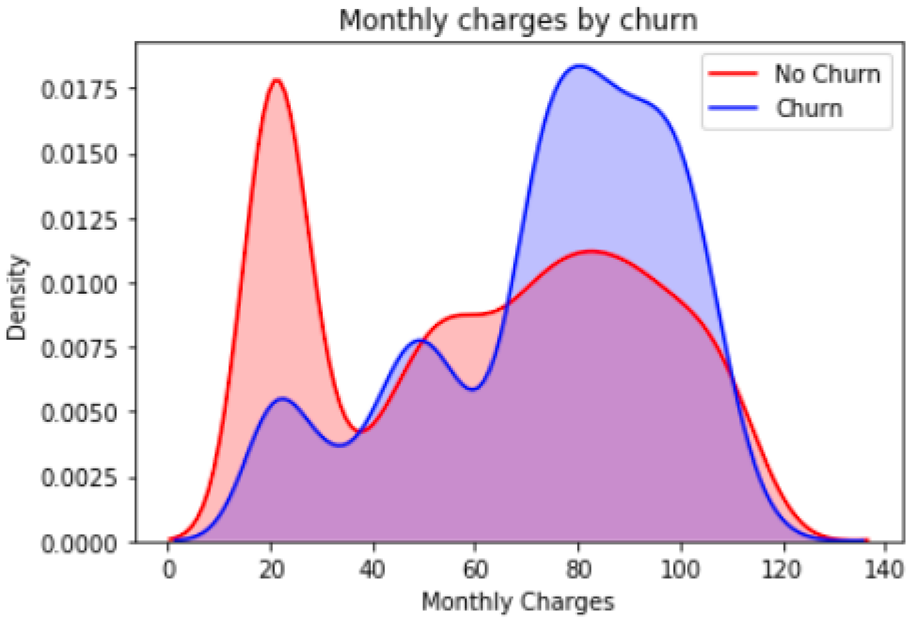
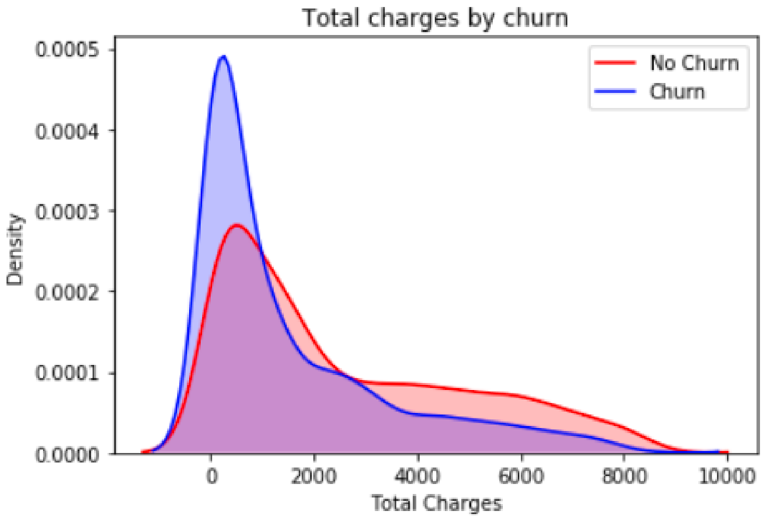


Figure 14 Increasing in Churn Rate

This plot indicates that the churn increases at higher monthly charges.

```
Tot =
sns.kdeplot(telco_data_dummies.TotalCharges[(telco_data_dummies["Chu
rn"] == 0) ],
                    color="Red", shade = True)
Tot =
sns.kdeplot(telco_data_dummies.TotalCharges[(telco_data_dummies["Chu
rn"] == 1) ],
                    ax =Tot, color="Blue", shade= True)
Tot.legend(["No Churn","Churn"],loc='upper right')
Tot.set_ylabel('Density')
Tot.set_xlabel('Total Charges')
Tot.set_title('Total charges by churn')
```

This is a surprising insight as the number of churners is very high at lesser total charges. This can be explained as the tenure is low, total charges will also be less, and the churn is negatively correlated to tenure so transitively, churn is negatively correlated with total charges.



Below is a correlation of all the attributes with the churn attribute. The bar graph value above 0.0 scale means that the attribute is positively correlated and the value below 0.0 scale means that the attribute is negatively correlated with the churn

Code:

```
plt.figure(figsize=(20,8))
telco_data_dummies.corr()['Churn'].sort_values(ascending =
False).plot(kind='bar')
```

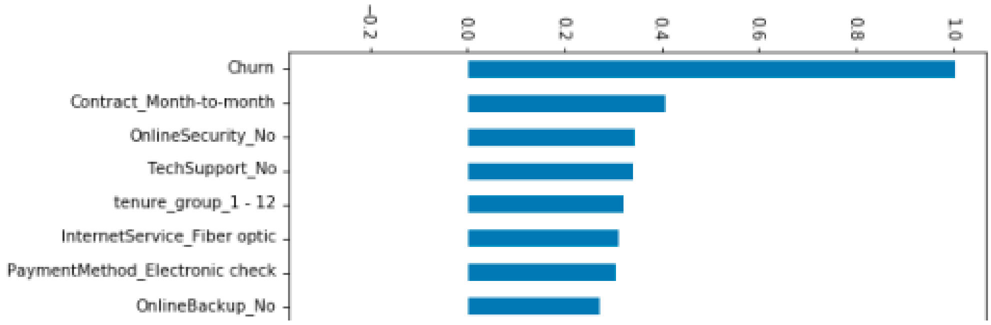


Figure 15. Correlation of churn with each factor or Variable as Input

The insights derived from the above bar graph are that there has been a high churn among the customers who have opted for month to month contracts, No online security, No Tech support, First year of subscription and Fibre Optics Internet. The churn is very low in the customers who had long term contracts, Subscriptions without internet service and the customers who engaged for 5+ years.

The attributes like gender, availability of phone service and having multiple lines have almost no impact on the churn when considered as standalone attributes for analysis,

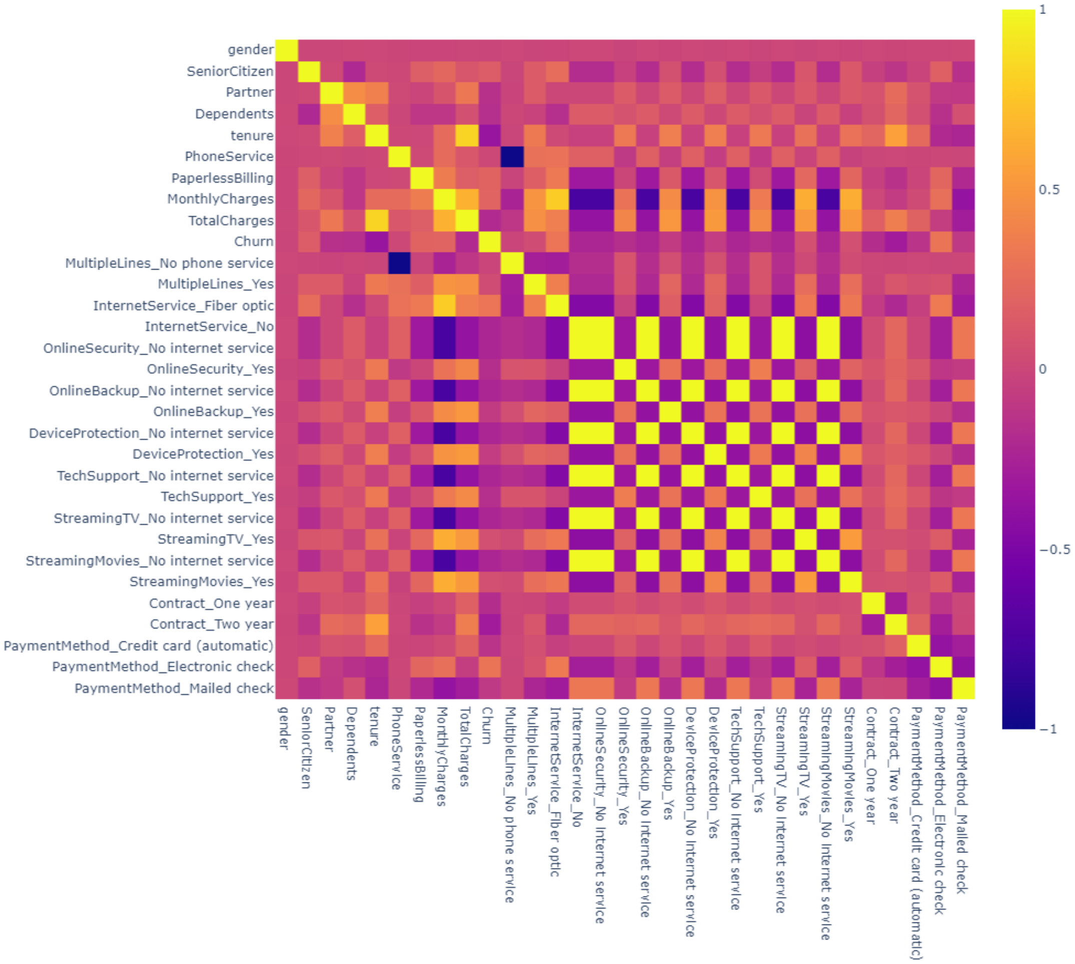


Figure 16. Insights can also be confirmed by the heatmap shown

5.2.4 Performing Bivariate analysis

Code:

```
new_df1_target0=telco_data.loc[telco_data["Churn"]==0]
new_df1_target1=telco_data.loc[telco_data["Churn"]==1]
def uniplot(df,col,title,hue =None):

    sns.set_style('whitegrid')
    sns.set_context('talk')
    plt.rcParams["axes.labelsize"] = 20
    plt.rcParams['axes.titlesize'] = 22
    plt.rcParams['axes.titlepad'] = 30
    temp = pd.Series(data = hue)
    fig, ax = plt.subplots()
    width = len(df[col].unique()) + 7 + 4*len(temp.unique())
    fig.set_size_inches(width , 8)
    plt.xticks(rotation=45)
    plt.yscale('log')
    plt.title(title)
    ax = sns.countplot(data = df, x= col,
order=df[col].value_counts().index,hue = hue,palette='bright')

    plt.show()
uniplot(new_df1_target1,col='Partner',title='Distribution of Gender for
Churned Customers',hue='gender')
```

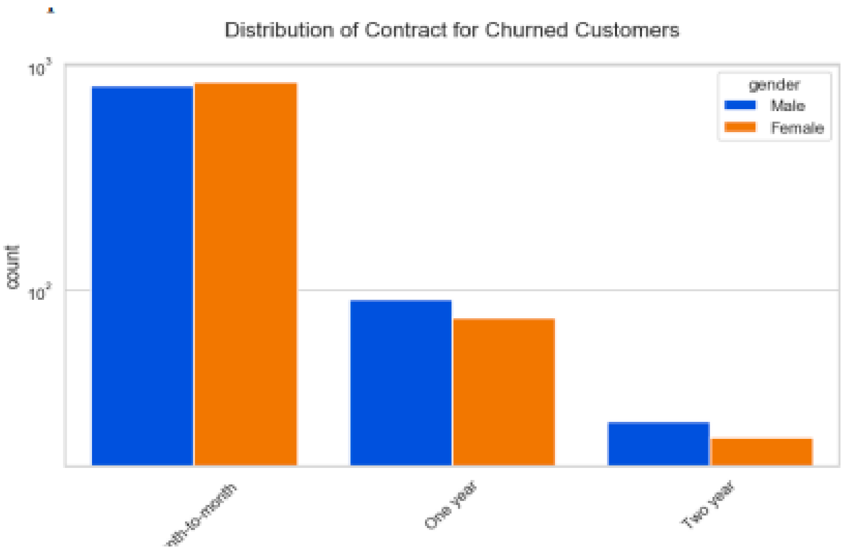


Figure 17. Bivariate analysis

5.3 Module 3 - Interpretation of Models

Customer churn prediction is a classification problem. So, we have implemented various classification algorithms like Decision Tree Classifier, Random Forest Classifier to choose the algorithm with best results. The Random Forest classifier has been used in this scenario as it proved to perform the best compared to other algorithms in this kind of scenario. Next step is building the prediction version. Lot of supervised gadget studying strategies is presently available to try this churn classification. First a part of this churn prediction section decided on a subset of authentic dataset, to educate the more than one gadget studying classifiers to pick the quality one. Then the second one a part of this churn prediction section starts after selecting the quality classifier compared with different classifiers. Here 2nd a part of this churn prediction section determined to apply the Principal Component Analysis to extract the brand new set of capabilities to educate the formerly decided on version once more to test whether or not any adjustments arise with the present stage of version accuracy.

Decision Tree Classifier :

In the Decision Tree Classifier, there are two steps, tree building and tree pruning. It begins with the root node which contains the complete dataset. Then it finds the best attribute using attribute selection measures and divides the dataset into possible values of that dataset. This process is performed recursively until further classification is not possible.

The attribute selection methods can be information gain or gini index. Information gain can be calculated using equation

Information Gain= Entropy(S) - [(Weighted Average) *Entropy (each feature)] (1) The entropy in equation (1) can be calculated using equation (2).

The gini index can be calculated using equation

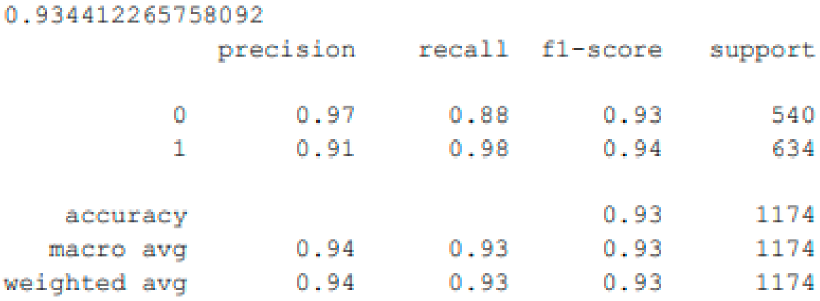
Gini Index = 1- j Pj 2 (3)

The accuracy in every model is very low as the dataset is highly imbalanced. 73.4 % of the data belongs to the non churner category and the remaining 26.6% of the data belongs to the churners category. To get a good accuracy and an efficient model, it is very important to have quality data which is balanced. So, the minority class data has been upsampled to make the dataset balanced which would provide better accuracy and performance,

After upsampling, the results shown by the Decision Tree classifier is:

```
df=pd.read_csv("tel_churn.csv")
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)

model_dt=DecisionTreeClassifier(criterion = "gini",random_state =
100,max_depth=6, min_samples_leaf=8)

model_dt.fit(x_train,y_train)

y_pred=model_dt.predict(x_test)

sm = SMOTEENN()

X_resampled, y_resampled = sm.fit_sample(x,y)

xr_train,xr_test,yr_train,yr_test=train_test_split(X_resampled,
y_resampled,test_size=0.2)

model_dt_smote=DecisionTreeClassifier(criterion = "gini",random_state =
100,max_depth=6, min_samples_leaf=8)

model_dt_smote.fit(xr_train,yr_train)

yr_predict = model_dt_smote.predict(xr_test)

model_score_r = model_dt_smote.score(xr_test, yr_test)

print(model_score_r)

print(metrics.classification_report(yr_test, yr_predict))
```

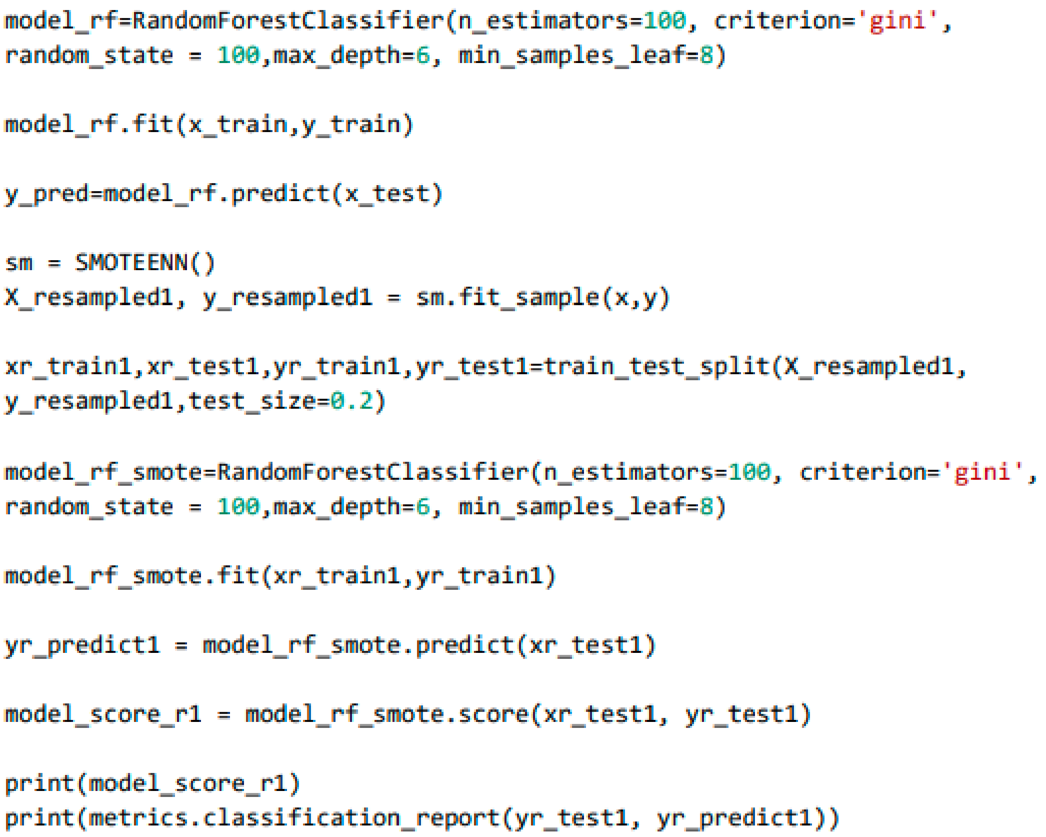
Output:



```
0.934412265758092
              precision    recall  f1-score   support

           0       0.97      0.88      0.93       540
           1       0.91      0.98      0.94       634

    accuracy                           0.93      1174
   macro avg       0.94      0.93      0.93      1174
weighted avg       0.94      0.93      0.93      1174
```

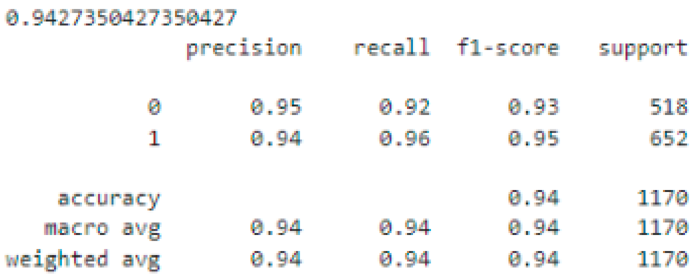
Random Forest Classifier

Code:



```
model_rf=RandomForestClassifier(n_estimators=100, criterion='gini',
random_state = 100,max_depth=6, min_samples_leaf=8)

model_rf.fit(x_train,y_train)

y_pred=model_rf.predict(x_test)

sm = SMOTEENN()
X_resampled1, y_resampled1 = sm.fit_sample(x,y)

xr_train1,xr_test1,yr_train1,yr_test1=train_test_split(X_resampled1,
y_resampled1,test_size=0.2)

model_rf_smote=RandomForestClassifier(n_estimators=100, criterion='gini',
random_state = 100,max_depth=6, min_samples_leaf=8)

model_rf_smote.fit(xr_train1,yr_train1)

yr_predict1 = model_rf_smote.predict(xr_test1)

model_score_r1 = model_rf_smote.score(xr_test1, yr_test1)

print(model_score_r1)
print(metrics.classification_report(yr_test1, yr_predict1))
```

Output:



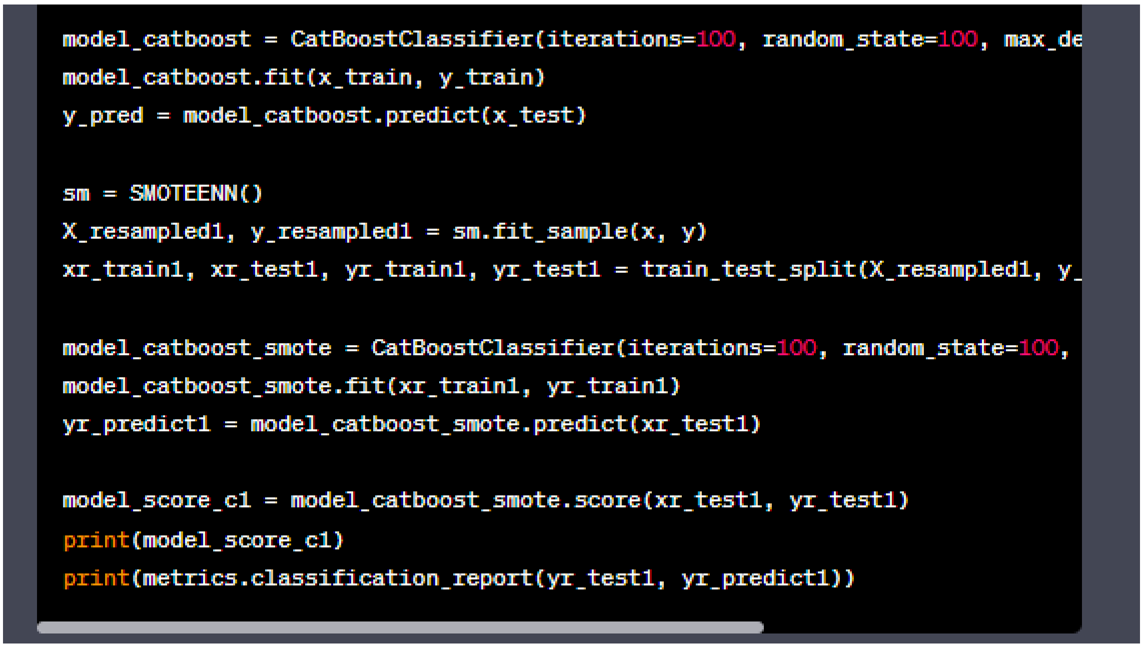
```
0.9427350427350427
              precision    recall  f1-score   support

           0       0.95      0.92      0.93       518
           1       0.94      0.96      0.95       652

    accuracy                           0.94      1170
   macro avg       0.94      0.94      0.94      1170
weighted avg       0.94      0.94      0.94      1170
```

Figure 18. Accuracy Level Comparison

To conclude, Accuracy Level is High With Random Forest Classifier

It's important to note that the accuracy level achieved by the CatBoost model may differ from the RandomForestClassifier. The performance of the model will depend on the dataset, the quality of the features, and the inherent patterns in the data. It's recommended to experiment with different models and hyperparameters to find the best-performing model for the specific churn prediction task.

If the data Set may be Like credit Score Allocated to Customer for Every Transaction They have made, they Data Set may vary we may use a different Classifier such as Catboost,Whose Code as Follows:



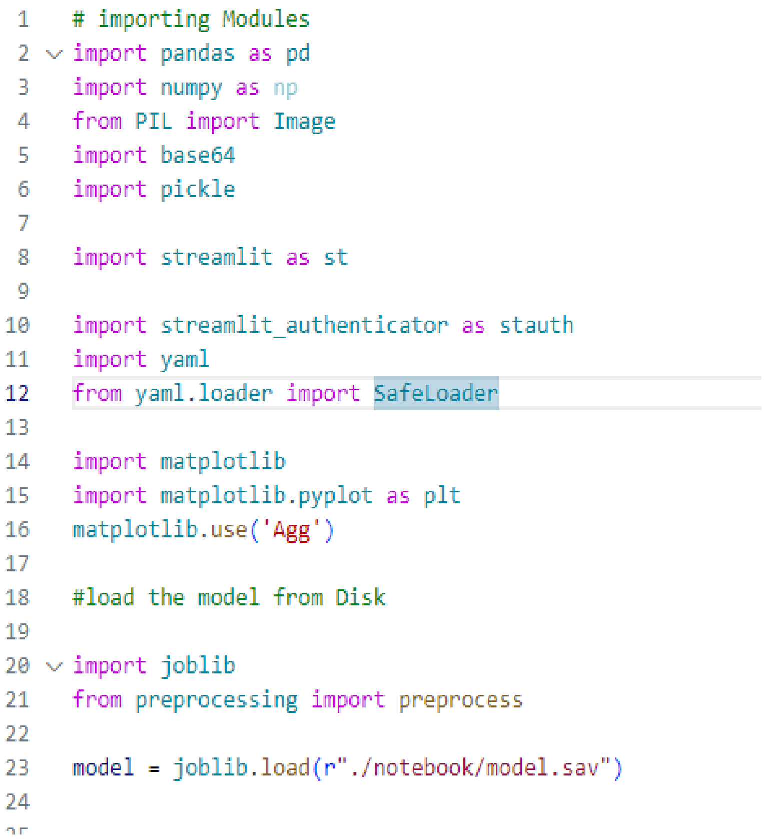
```
model_catboost = CatBoostClassifier(iterations=100, random_state=100, max_de
model_catboost.fit(x_train, y_train)
y_pred = model_catboost.predict(x_test)

sm = SMOTEENN()
X_resampled1, y_resampled1 = sm.fit_sample(x, y)
xr_train1, xr_test1, yr_train1, yr_test1 = train_test_split(X_resampled1, y_

model_catboost_smote = CatBoostClassifier(iterations=100, random_state=100,
model_catboost_smote.fit(xr_train1, yr_train1)
yr_predict1 = model_catboost_smote.predict(xr_test1)

model_score_c1 = model_catboost_smote.score(xr_test1, yr_test1)
print(model_score_c1)
print(metrics.classification_report(yr_test1, yr_predict1))
```

Figure 19. Catboost Classifier

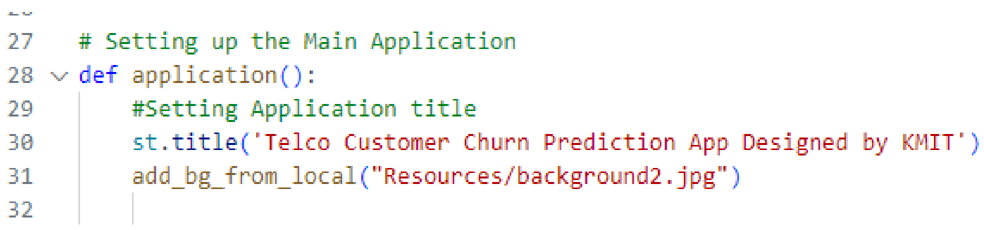
5.4 Module 4 - BUILDING APPLICATION

The frontend application is built using a streamlit module. Streamlit is a framework in python which enables the users to create simple machine learning and data related web apps without the usage of traditional frontend technologies like HTML, CSS and Javascript. Using streamlit, any one can create webapps with python-like syntax. Streamlit also provides an option to display the web content in a beautiful manner by accepting markdown syntax, which makes it easier to code and design the frontend. Streamlit contains predefined classes for many kinds of frontend elements which enable the user to concentrate on the logic and design rather than focussing on reinventing the wheel.

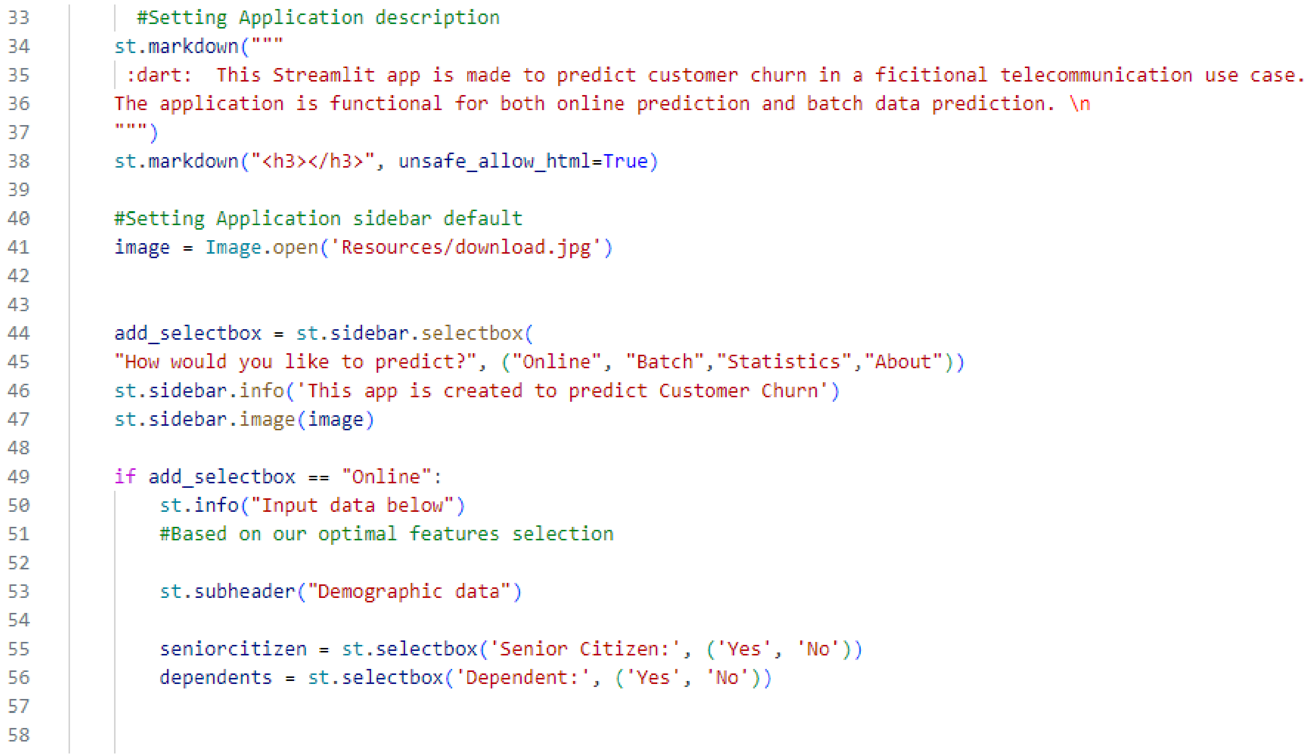
App.py:



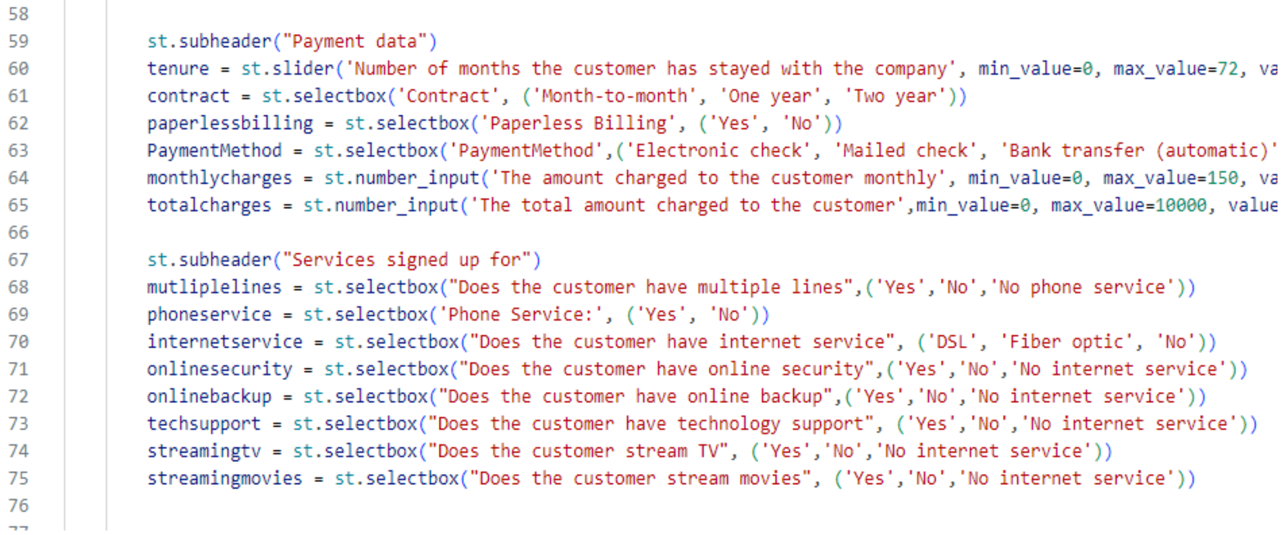
```python
# importing Modules
import pandas as pd
import numpy as np
from PIL import Image
import base64
import pickle

import streamlit as st

import streamlit_authenticator as stauth
import yaml
from yaml.loader import SafeLoader

import matplotlib
import matplotlib.pyplot as plt
matplotlib.use('Agg')

#load the model from Disk

import joblib
from preprocessing import preprocess

model = joblib.load(r"./notebook/model.sav")

```



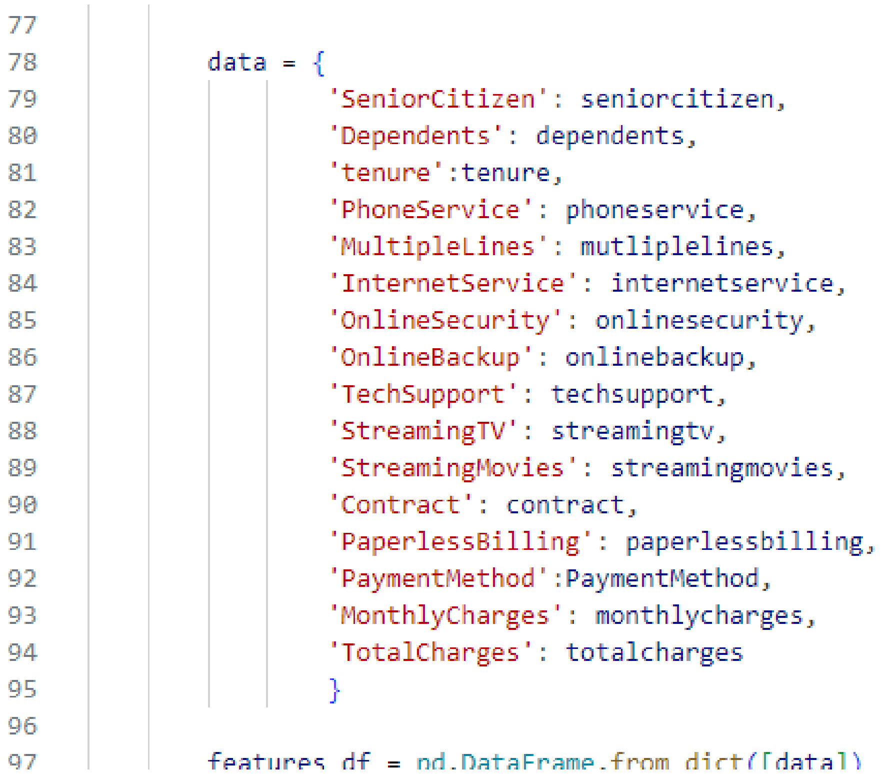
```python
# Setting up the Main Application
def application():
    #Setting Application title
    st.title('Telco Customer Churn Prediction App Designed by KMIT')
    add_bg_from_local("Resources/background2.jpg")

```



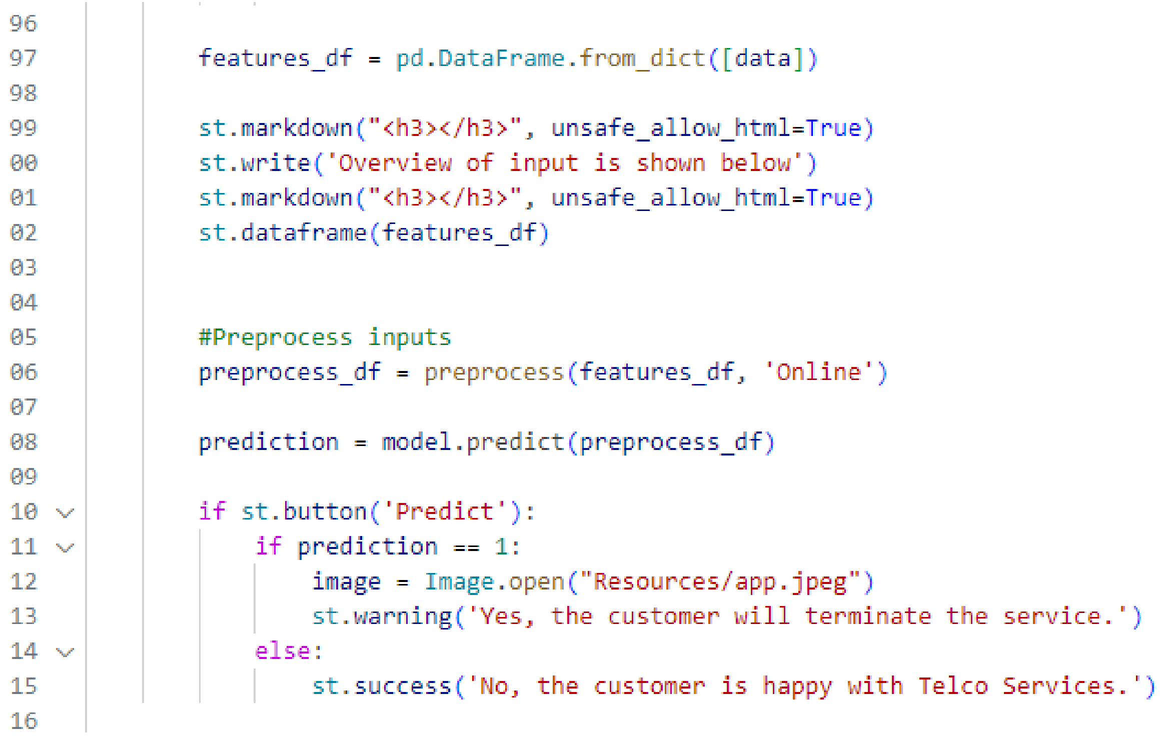
```
 #Setting Application description
st.markdown("""
 :dart:  This Streamlit app is made to predict customer churn in a ficitional telecommunication use case.
The application is functional for both online prediction and batch data prediction. \n
""")
st.markdown("<h3></h3>", unsafe_allow_html=True)

#Setting Application sidebar default
image = Image.open('Resources/download.jpg')


add_selectbox = st.sidebar.selectbox(
"How would you like to predict?", ("Online", "Batch","Statistics","About"))
st.sidebar.info('This app is created to predict Customer Churn')
st.sidebar.image(image)

if add_selectbox == "Online":
    st.info("Input data below")
    #Based on our optimal features selection

    st.subheader("Demographic data")

    seniorcitizen = st.selectbox('Senior Citizen:', ('Yes', 'No'))
    dependents = st.selectbox('Dependent:', ('Yes', 'No'))


```



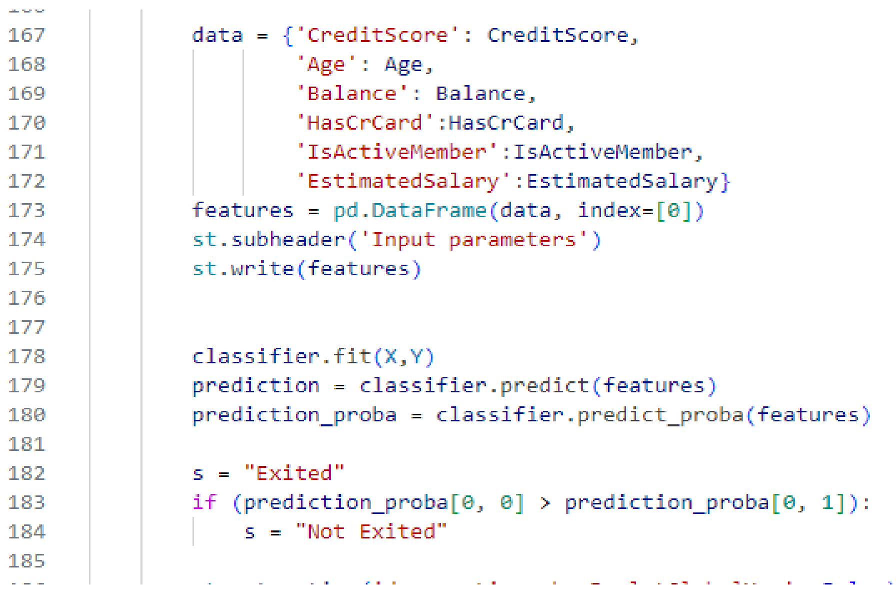
```

    st.subheader("Payment data")
    tenure = st.slider('Number of months the customer has stayed with the company', min_value=0, max_value=72, va
    contract = st.selectbox('Contract', ('Month-to-month', 'One year', 'Two year'))
    paperlessbilling = st.selectbox('Paperless Billing', ('Yes', 'No'))
    PaymentMethod = st.selectbox('PaymentMethod',('Electronic check', 'Mailed check', 'Bank transfer (automatic)'
    monthlycharges = st.number_input('The amount charged to the customer monthly', min_value=0, max_value=150, va
    totalcharges = st.number_input('The total amount charged to the customer',min_value=0, max_value=10000, value

    st.subheader("Services signed up for")
    mutliplelines = st.selectbox("Does the customer have multiple lines",('Yes','No','No phone service'))
    phoneservice = st.selectbox('Phone Service:', ('Yes', 'No'))
    internetservice = st.selectbox("Does the customer have internet service", ('DSL', 'Fiber optic', 'No'))
    onlinesecurity = st.selectbox("Does the customer have online security",('Yes','No','No internet service'))
    onlinebackup = st.selectbox("Does the customer have online backup",('Yes','No','No internet service'))
    techsupport = st.selectbox("Does the customer have technology support", ('Yes','No','No internet service'))
    streamingtv = st.selectbox("Does the customer stream TV", ('Yes','No','No internet service'))
    streamingmovies = st.selectbox("Does the customer stream movies", ('Yes','No','No internet service'))

```



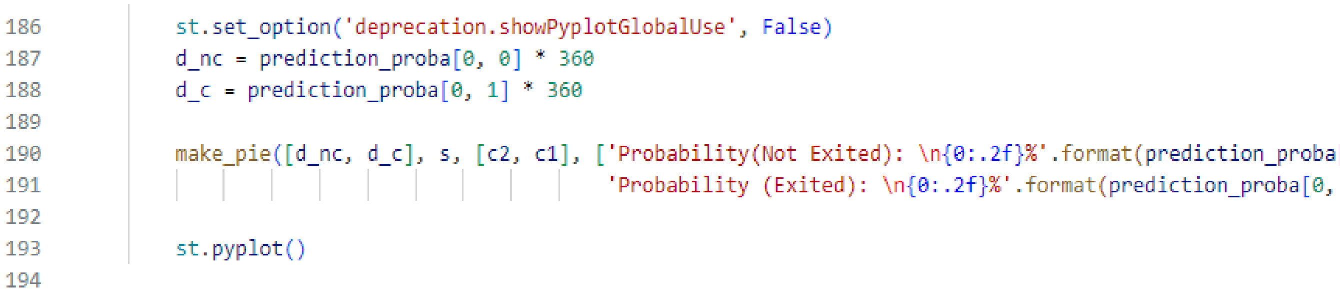
```
        data = {
                'SeniorCitizen': seniorcitizen,
                'Dependents': dependents,
                'tenure':tenure,
                'PhoneService': phoneservice,
                'MultipleLines': mutliplelines,
                'InternetService': internetservice,
                'OnlineSecurity': onlinesecurity,
                'OnlineBackup': onlinebackup,
                'TechSupport': techsupport,
                'StreamingTV': streamingtv,
                'StreamingMovies': streamingmovies,
                'Contract': contract,
                'PaperlessBilling': paperlessbilling,
                'PaymentMethod':PaymentMethod,
                'MonthlyCharges': monthlycharges,
                'TotalCharges': totalcharges
                }

        features_df = pd.DataFrame.from_dict([data])
```



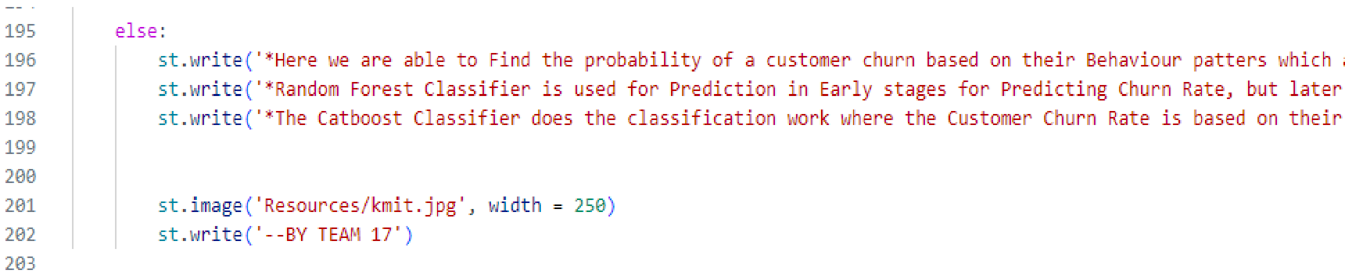
```
        features_df = pd.DataFrame.from_dict([data])

        st.markdown("<h3></h3>", unsafe_allow_html=True)
        st.write('Overview of input is shown below')
        st.markdown("<h3></h3>", unsafe_allow_html=True)
        st.dataframe(features_df)


        #Preprocess inputs
        preprocess_df = preprocess(features_df, 'Online')

        prediction = model.predict(preprocess_df)

        if st.button('Predict'):
            if prediction == 1:
                image = Image.open("Resources/app.jpeg")
                st.warning('Yes, the customer will terminate the service.')
            else:
                st.success('No, the customer is happy with Telco Services.')
```



```
        data = {'CreditScore': CreditScore,
                'Age': Age,
                'Balance': Balance,
                'HasCrCard':HasCrCard,
                'IsActiveMember':IsActiveMember,
                'EstimatedSalary':EstimatedSalary}
        features = pd.DataFrame(data, index=[0])
        st.subheader('Input parameters')
        st.write(features)


        classifier.fit(X,Y)
        prediction = classifier.predict(features)
        prediction_proba = classifier.predict_proba(features)

        s = "Exited"
        if (prediction_proba[0, 0] > prediction_proba[0, 1]):
            s = "Not Exited"

```

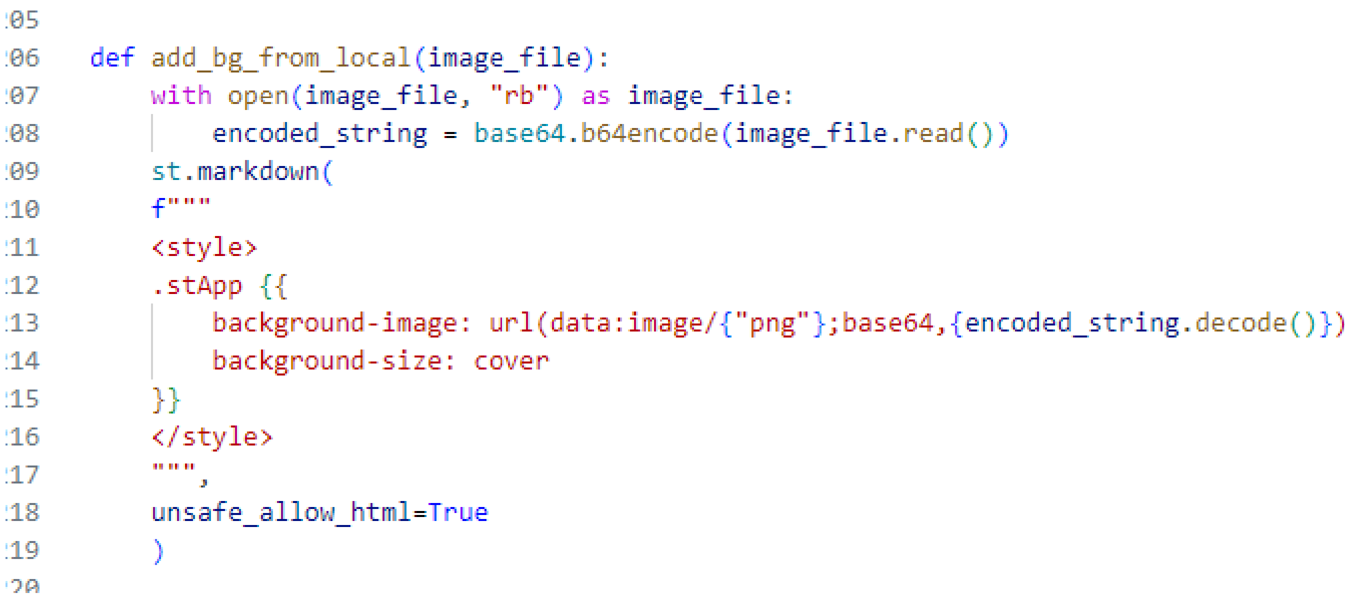


```
        st.set_option('deprecation.showPyplotGlobalUse', False)
        d_nc = prediction_proba[0, 0] * 360
        d_c = prediction_proba[0, 1] * 360

        make_pie([d_nc, d_c], s, [c2, c1], ['Probability(Not Exited): \n{0:.2f}%'.format(prediction_proba
                                            'Probability (Exited): \n{0:.2f}%'.format(prediction_proba[0,

        st.pyplot()

```

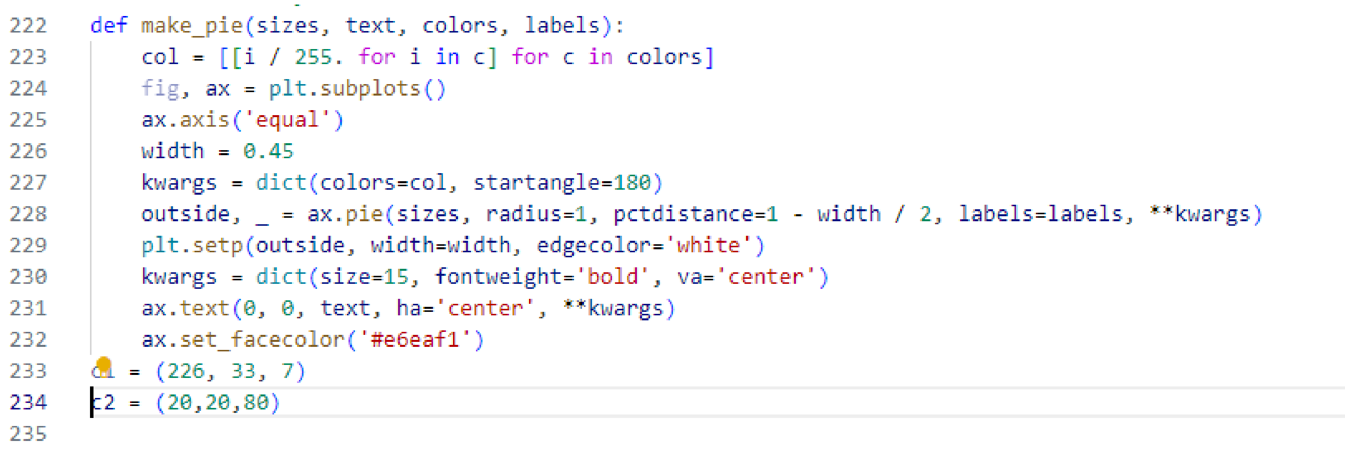


```
    else:
        st.write('*Here we are able to Find the probability of a customer churn based on their Behaviour patters which
        st.write('*Random Forest Classifier is used for Prediction in Early stages for Predicting Churn Rate, but later
        st.write('*The Catboost Classifier does the classification work where the Customer Churn Rate is based on their


        st.image('Resources/kmit.jpg', width = 250)
        st.write('--BY TEAM 17')

```

UI.py :

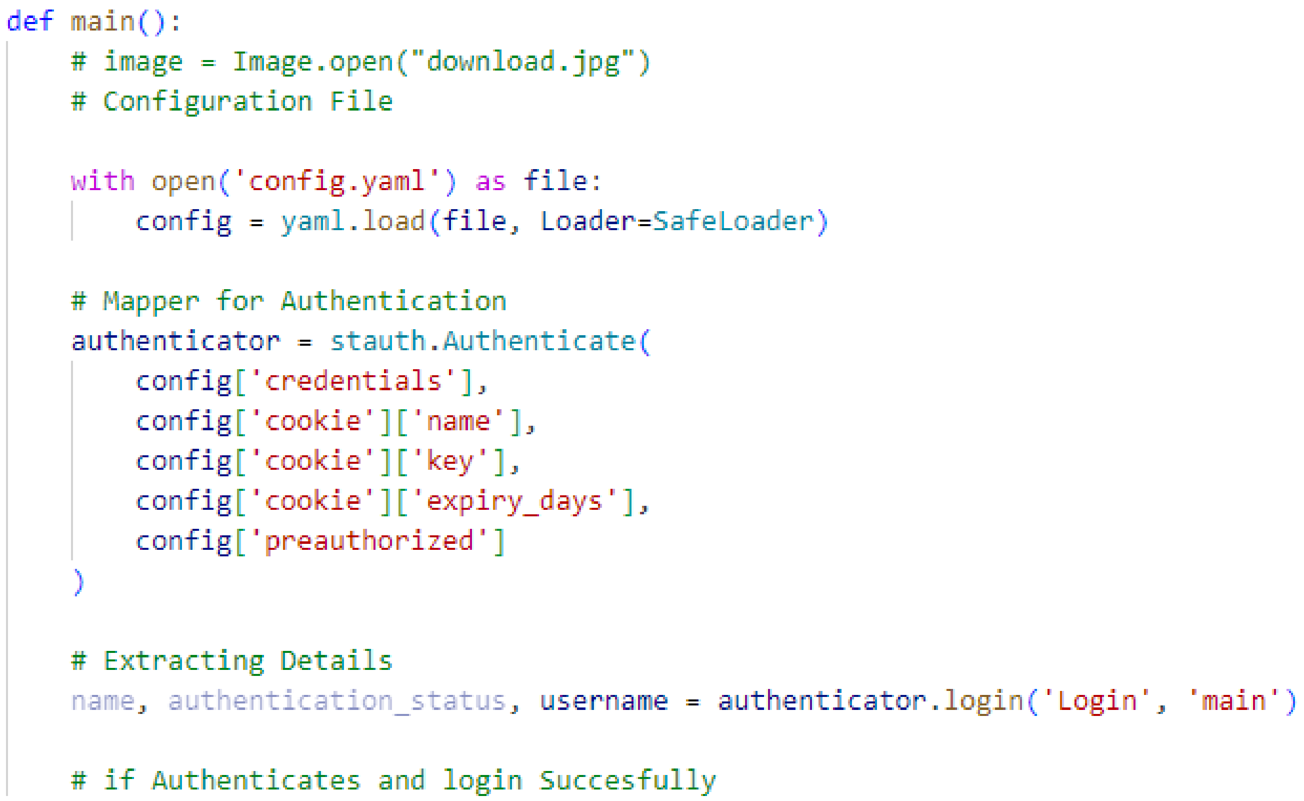


```python

def add_bg_from_local(image_file):
    with open(image_file, "rb") as image_file:
        encoded_string = base64.b64encode(image_file.read())
    st.markdown(
    f"""
    <style>
    .stApp {{
        background-image: url(data:image/{"png"};base64,{encoded_string.decode()})
        background-size: cover
    }}
    </style>
    """,
    unsafe_allow_html=True
    )

```

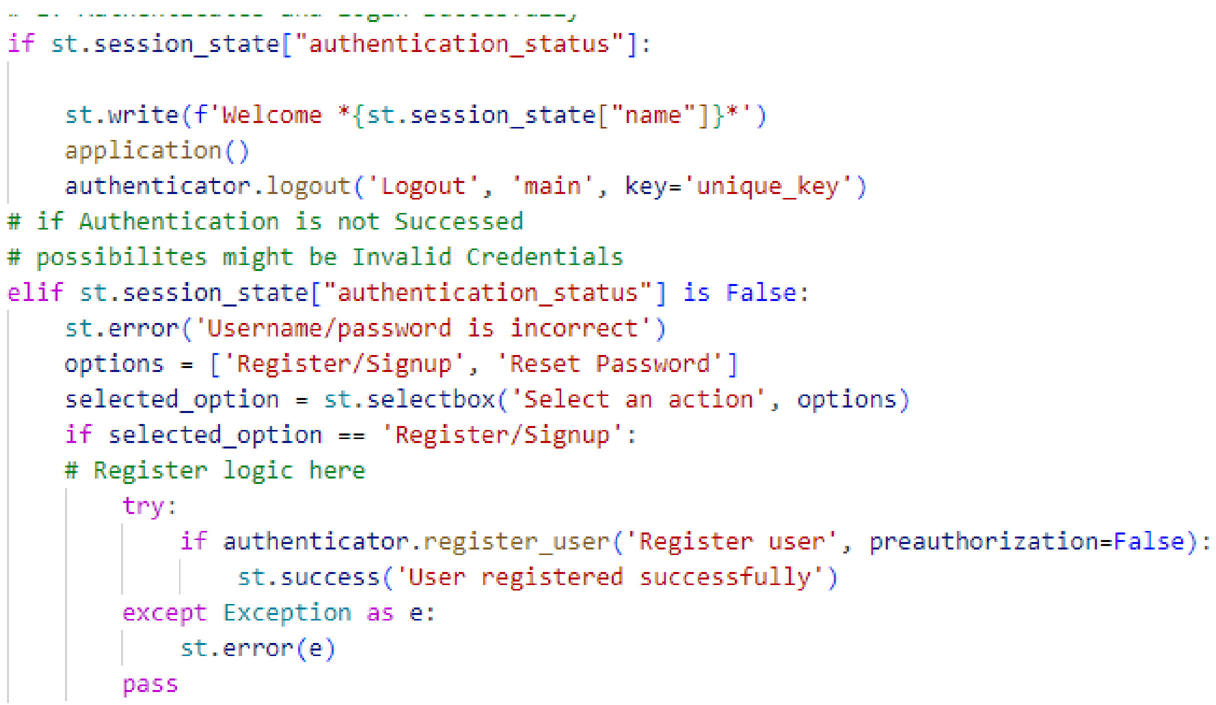


```python
def make_pie(sizes, text, colors, labels):
    col = [[i / 255. for i in c] for c in colors]
    fig, ax = plt.subplots()
    ax.axis('equal')
    width = 0.45
    kwargs = dict(colors=col, startangle=180)
    outside, _ = ax.pie(sizes, radius=1, pctdistance=1 - width / 2, labels=labels, **kwargs)
    plt.setp(outside, width=width, edgecolor='white')
    kwargs = dict(size=15, fontweight='bold', va='center')
    ax.text(0, 0, text, ha='center', **kwargs)
    ax.set_facecolor('#e6eaf1')
c1 = (226, 33, 7)
c2 = (20,20,80)

```

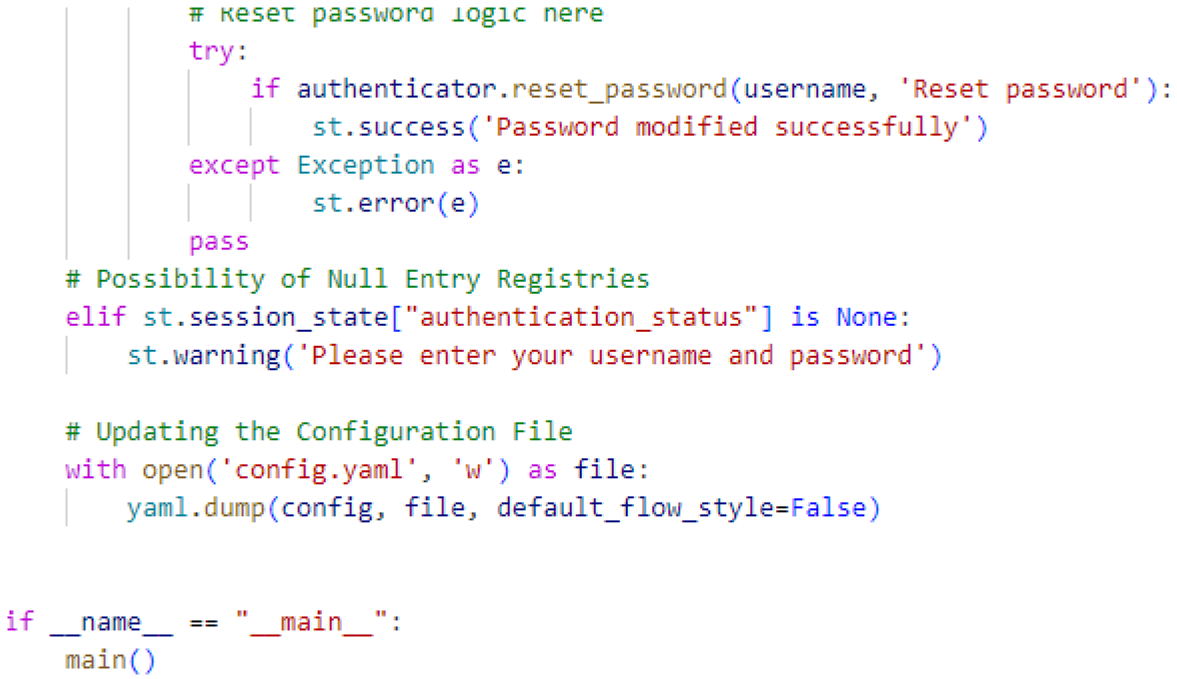
LoginAnd Signup Module :



```python
def main():
    # image = Image.open("download.jpg")
    # Configuration File

    with open('config.yaml') as file:
        config = yaml.load(file, Loader=SafeLoader)

    # Mapper for Authentication
    authenticator = stauth.Authenticate(
        config['credentials'],
        config['cookie']['name'],
        config['cookie']['key'],
        config['cookie']['expiry_days'],
        config['preauthorized']
    )

    # Extracting Details
    name, authentication_status, username = authenticator.login('Login', 'main')

    # if Authenticates and login Succesfully
```

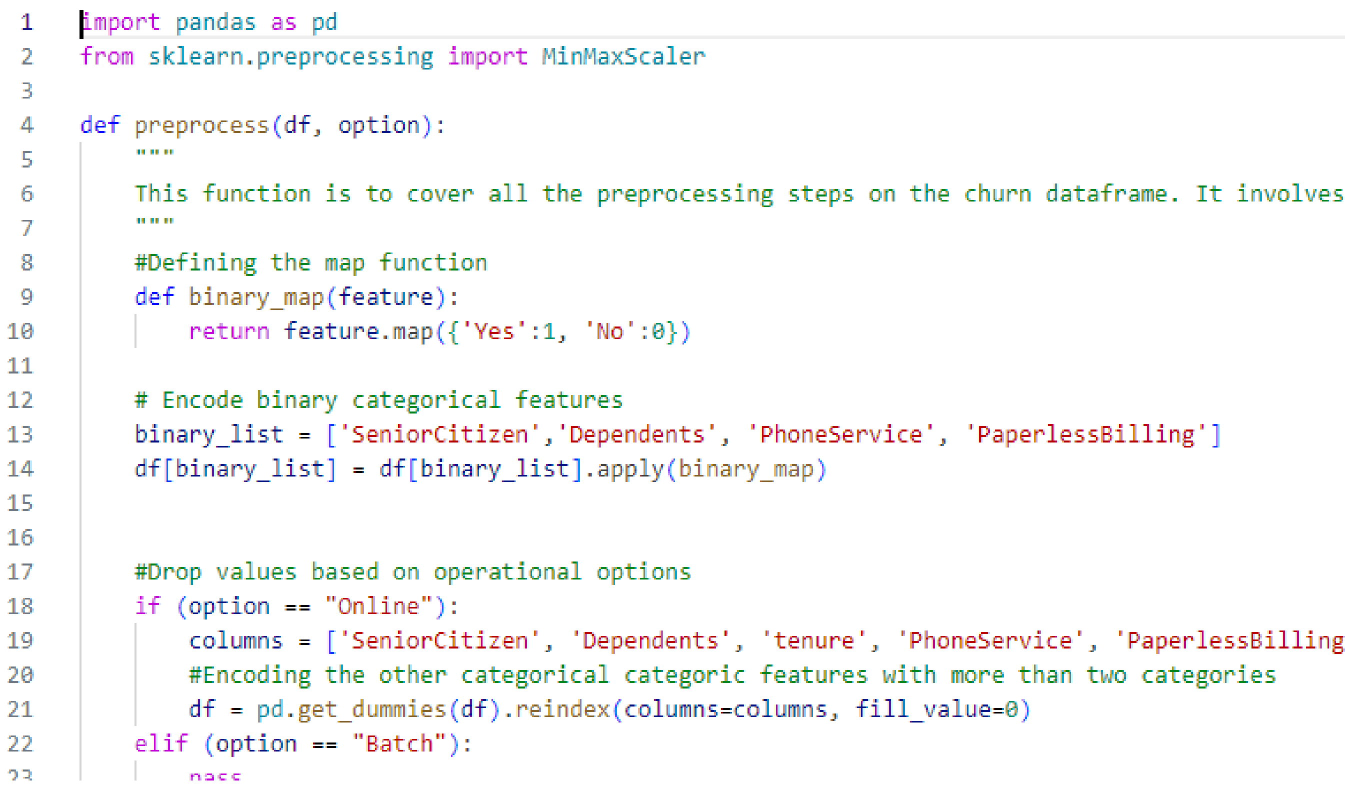


```
    # If Authenticated and Login Successfully
    if st.session_state["authentication_status"]:

        st.write(f'Welcome *{st.session_state["name"]}*')
        application()
        authenticator.logout('Logout', 'main', key='unique_key')
    # if Authentication is not Successed
    # possibilites might be Invalid Credentials
    elif st.session_state["authentication_status"] is False:
        st.error('Username/password is incorrect')
        options = ['Register/Signup', 'Reset Password']
        selected_option = st.selectbox('Select an action', options)
        if selected_option == 'Register/Signup':
        # Register logic here
            try:
                if authenticator.register_user('Register user', preauthorization=False):
                    st.success('User registered successfully')
            except Exception as e:
                st.error(e)
            pass
```

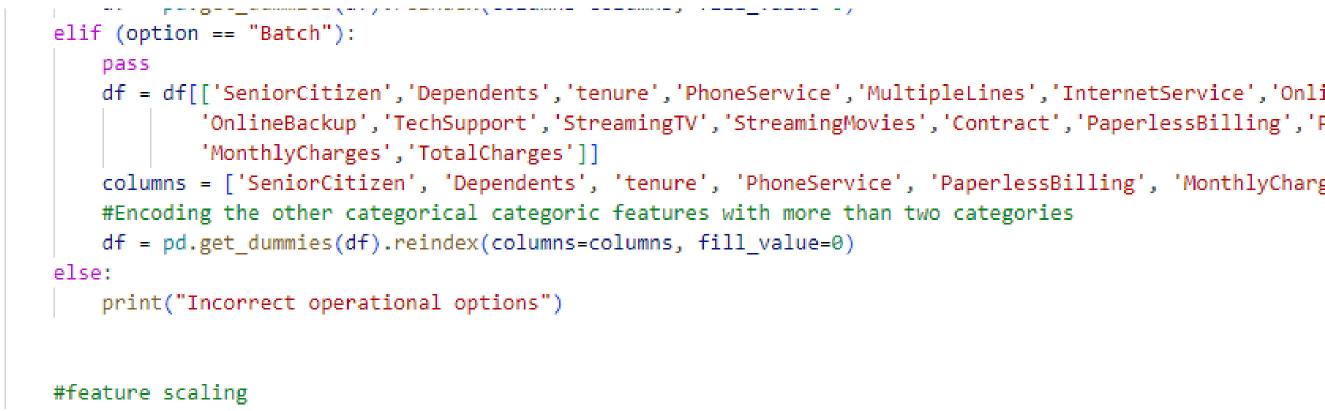


```
            # Reset password logic here
            try:
                if authenticator.reset_password(username, 'Reset password'):
                    st.success('Password modified successfully')
            except Exception as e:
                    st.error(e)
            pass
    # Possibility of Null Entry Registries
    elif st.session_state["authentication_status"] is None:
        st.warning('Please enter your username and password')

    # Updating the Configuration File
    with open('config.yaml', 'w') as file:
        yaml.dump(config, file, default_flow_style=False)


if __name__ == "__main__":
    main()
```

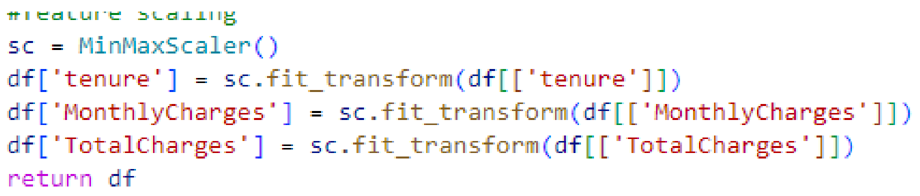
Preprocessing.py :



```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

def preprocess(df, option):
    """
    This function is to cover all the preprocessing steps on the churn dataframe. It involves
    """
    #Defining the map function
    def binary_map(feature):
        return feature.map({'Yes':1, 'No':0})

    # Encode binary categorical features
    binary_list = ['SeniorCitizen','Dependents', 'PhoneService', 'PaperlessBilling']
    df[binary_list] = df[binary_list].apply(binary_map)


    #Drop values based on operational options
    if (option == "Online"):
        columns = ['SeniorCitizen', 'Dependents', 'tenure', 'PhoneService', 'PaperlessBilling
        #Encoding the other categorical categoric features with more than two categories
        df = pd.get_dummies(df).reindex(columns=columns, fill_value=0)
    elif (option == "Batch"):
        pass
```

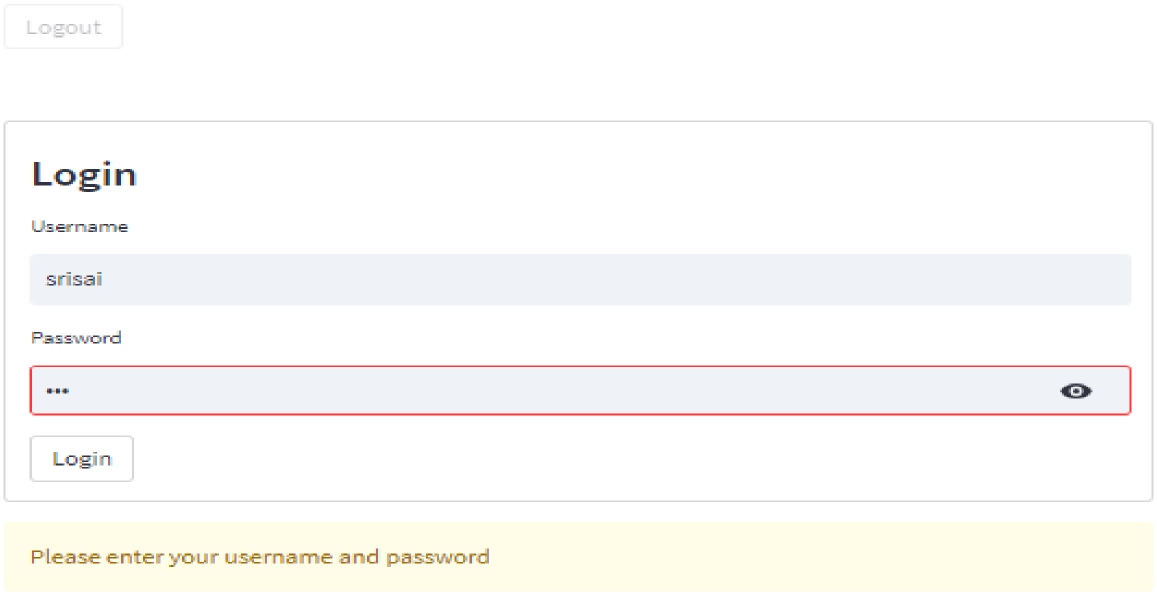


```
    elif (option == "Batch"):
        pass
        df = df[['SeniorCitizen','Dependents','tenure','PhoneService','MultipleLines','InternetService','Onli
                'OnlineBackup','TechSupport','StreamingTV','StreamingMovies','Contract','PaperlessBilling','P
                'MonthlyCharges','TotalCharges']]
        columns = ['SeniorCitizen', 'Dependents', 'tenure', 'PhoneService', 'PaperlessBilling', 'MonthlyCharg
        #Encoding the other categorical categoric features with more than two categories
        df = pd.get_dummies(df).reindex(columns=columns, fill_value=0)
    else:
        print("Incorrect operational options")


    #feature scaling
```



```
#feature scaling
sc = MinMaxScaler()
df['tenure'] = sc.fit_transform(df[['tenure']])
df['MonthlyCharges'] = sc.fit_transform(df[['MonthlyCharges']])
df['TotalCharges'] = sc.fit_transform(df[['TotalCharges']])
return df
```

5.5 RESULTS And SCREENSHOTS



Username

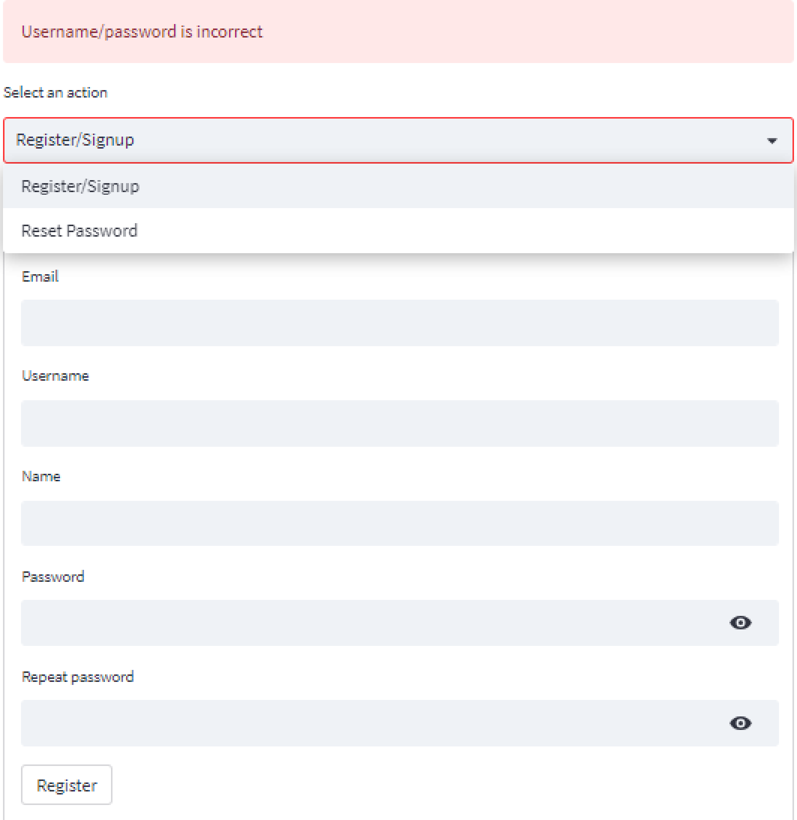
srisai

Password

•••

Please enter your username and password

Screenshot 1 : Enter Login Details

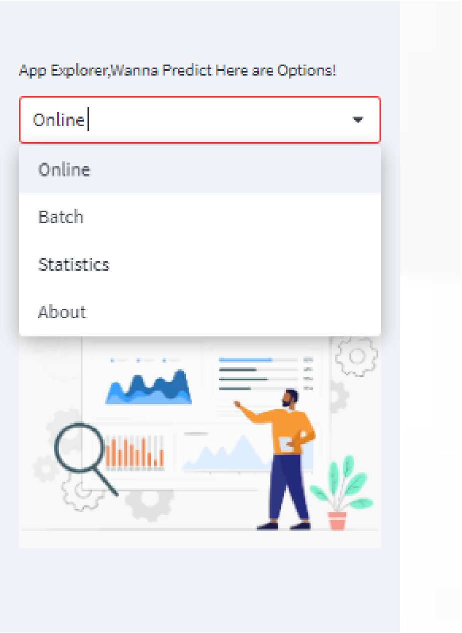


Username/password is incorrect

Select an action

Register/Signup

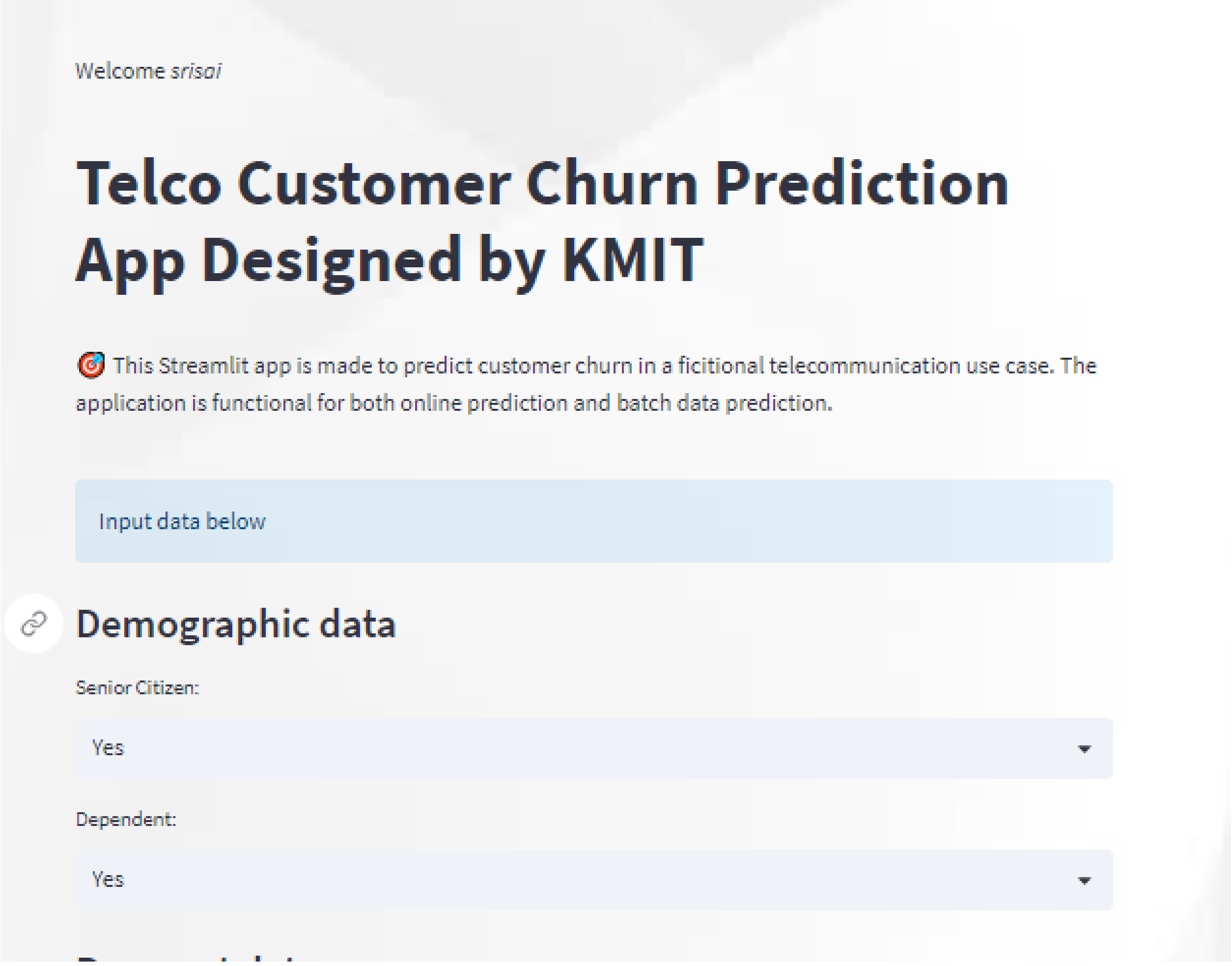
Screenshot 2 : User Authentication Fails Register or Reset Password



App Explorer,Wanna Predict Here are Options!

Online

Screenshot 3 : User Authentication is Successful,We can Find above Options



Welcome *srisai*

Telco Customer Churn Prediction App Designed by KMIT

🎯 This Streamlit app is made to predict customer churn in a ficitional telecommunication use case. The application is functional for both online prediction and batch data prediction.

Input data below

Demographic data

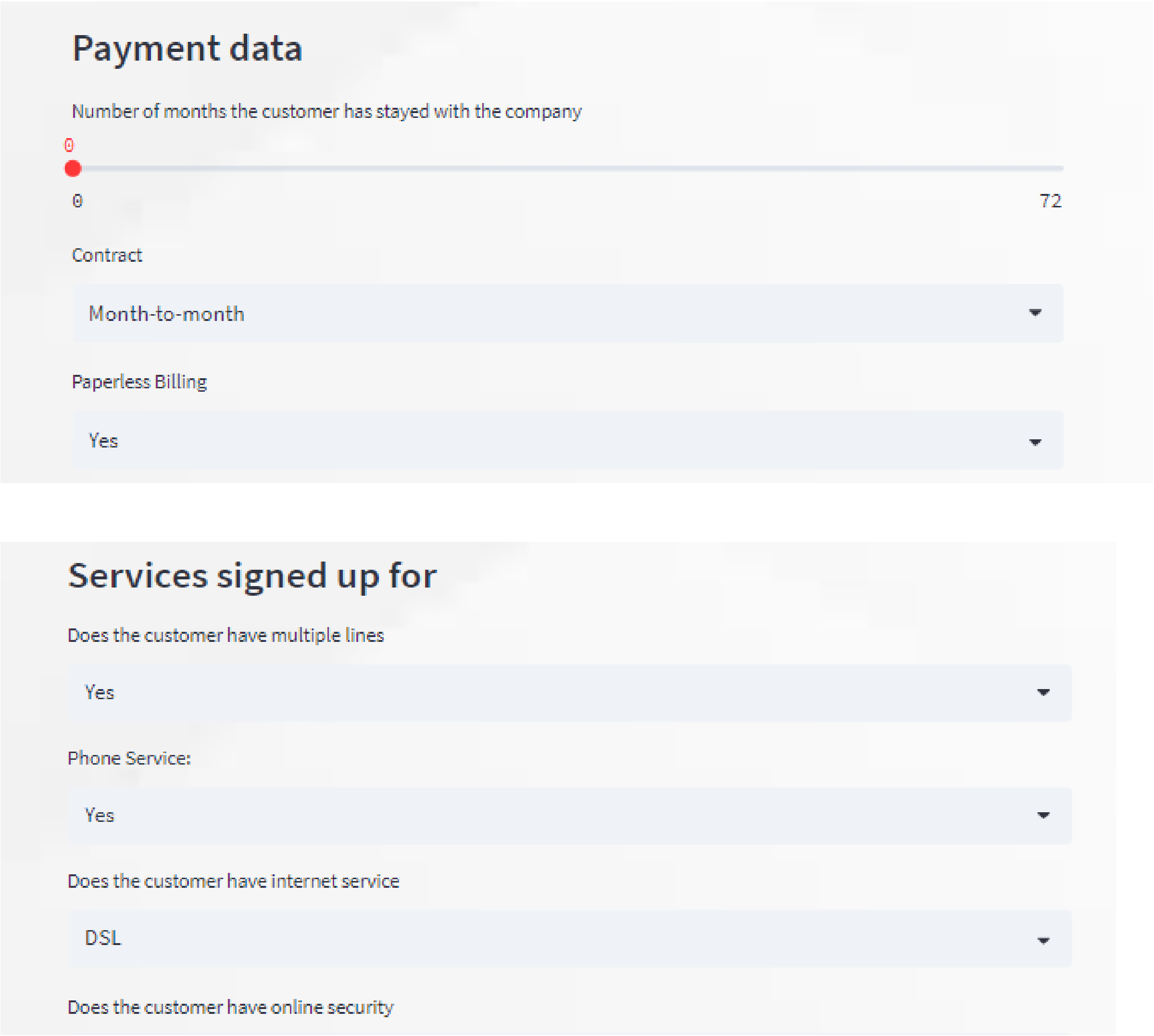
Senior Citizen:

Yes

Dependent:

Yes

Screenshot 4. Online Mode,Entering Demographic Data,



Payment data

Number of months the customer has stayed with the company

0

Contract

Month-to-month

Paperless Billing

Yes

Services signed up for

Does the customer have multiple lines

Yes

Phone Service:

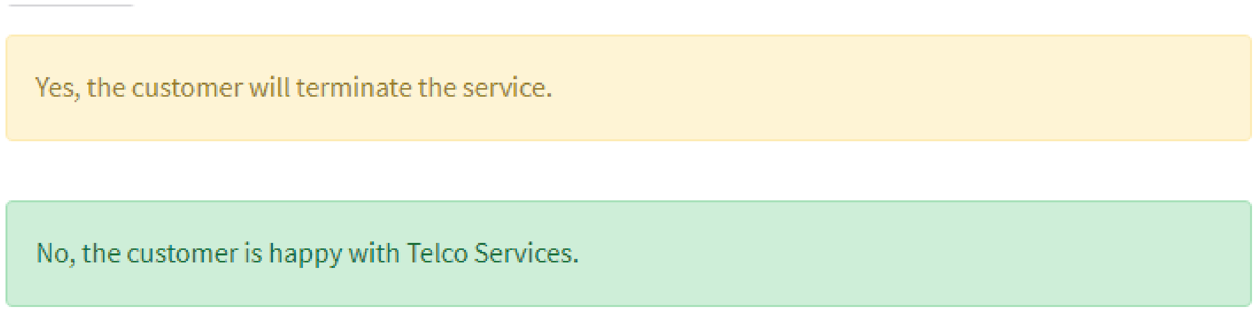
Yes

Does the customer have internet service

DSL

Does the customer have online security

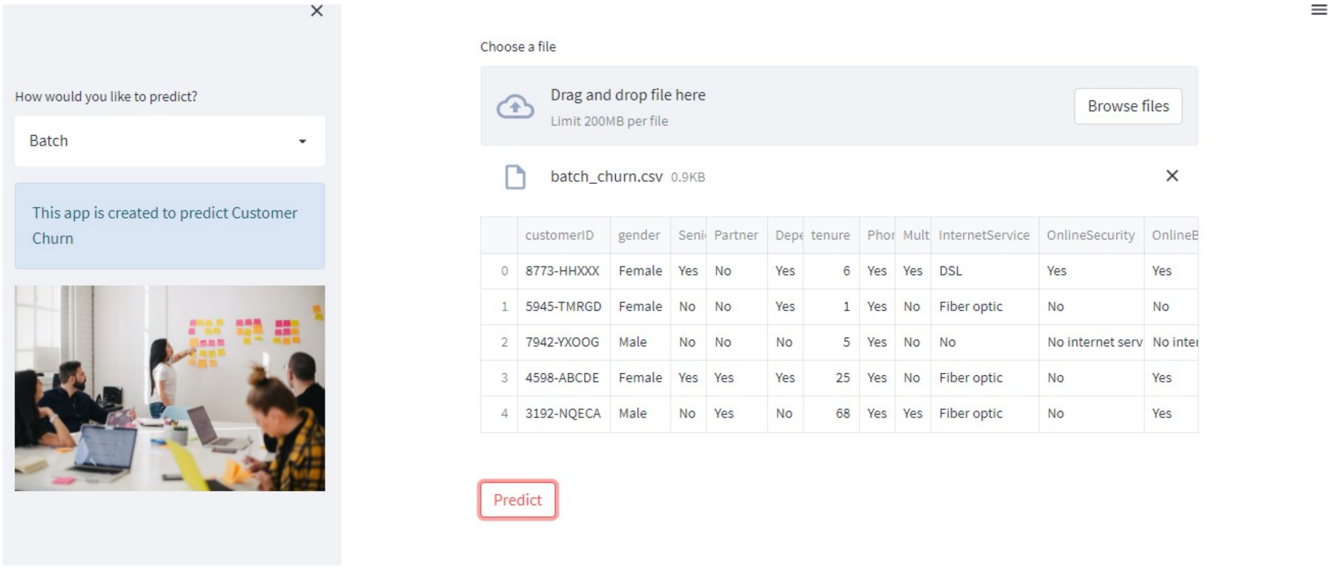
Screenshot 5 : Entering Payment Data and Services Signed data



Yes, the customer will terminate the service.

No, the customer is happy with Telco Services.

Screenshot 6 : Output Predicted, based on Rendered Data



How would you like to predict?

Batch

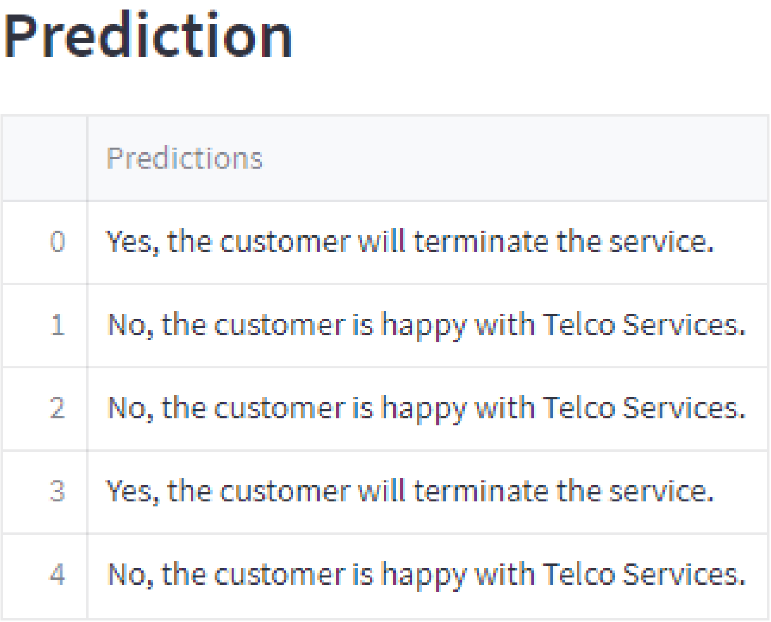
This app is created to predict Customer Churn

Choose a file

batch_churn.csv 0.9KB

	customerID	gender	Seni	Partner	Depe	tenure	Phor	Mult	InternetService	OnlineSecurity	OnlineB
0	8773-HHXXX	Female	Yes	No	Yes	6	Yes	Yes	DSL	Yes	Yes
1	5945-TMRGD	Female	No	No	Yes	1	Yes	No	Fiber optic	No	No
2	7942-YXOOG	Male	No	No	No	5	Yes	No	No	No internet serv	No inte
3	4598-ABCDE	Female	Yes	Yes	Yes	25	Yes	No	Fiber optic	No	Yes
4	3192-NQECA	Male	No	Yes	No	68	Yes	Yes	Fiber optic	No	Yes

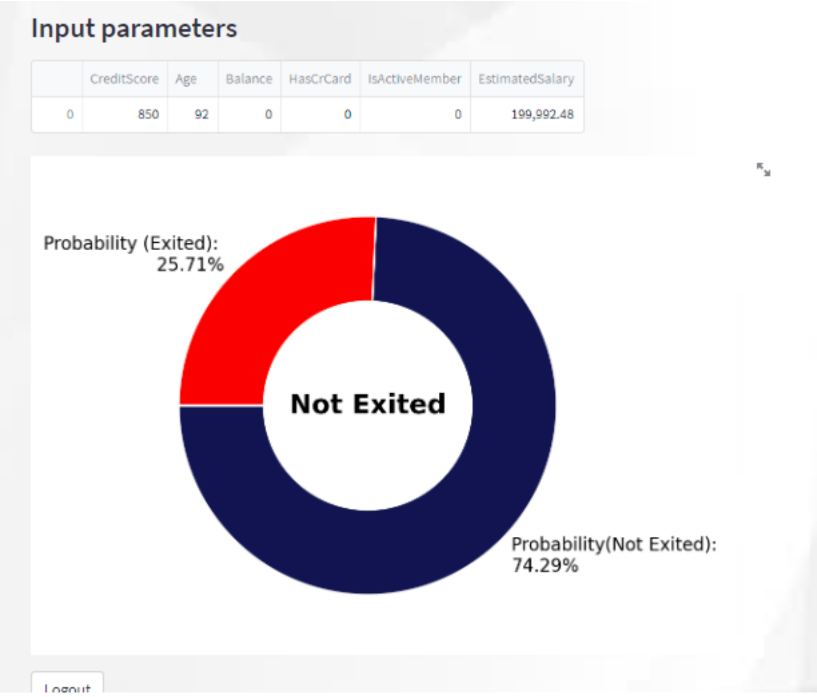
Screenshot 7 : Batch Processing ,i.e Prediction of Churn rate in Multiple Scenarios



Prediction

	Predictions
0	Yes, the customer will terminate the service.
1	No, the customer is happy with Telco Services.
2	No, the customer is happy with Telco Services.
3	Yes, the customer will terminate the service.
4	No, the customer is happy with Telco Services.

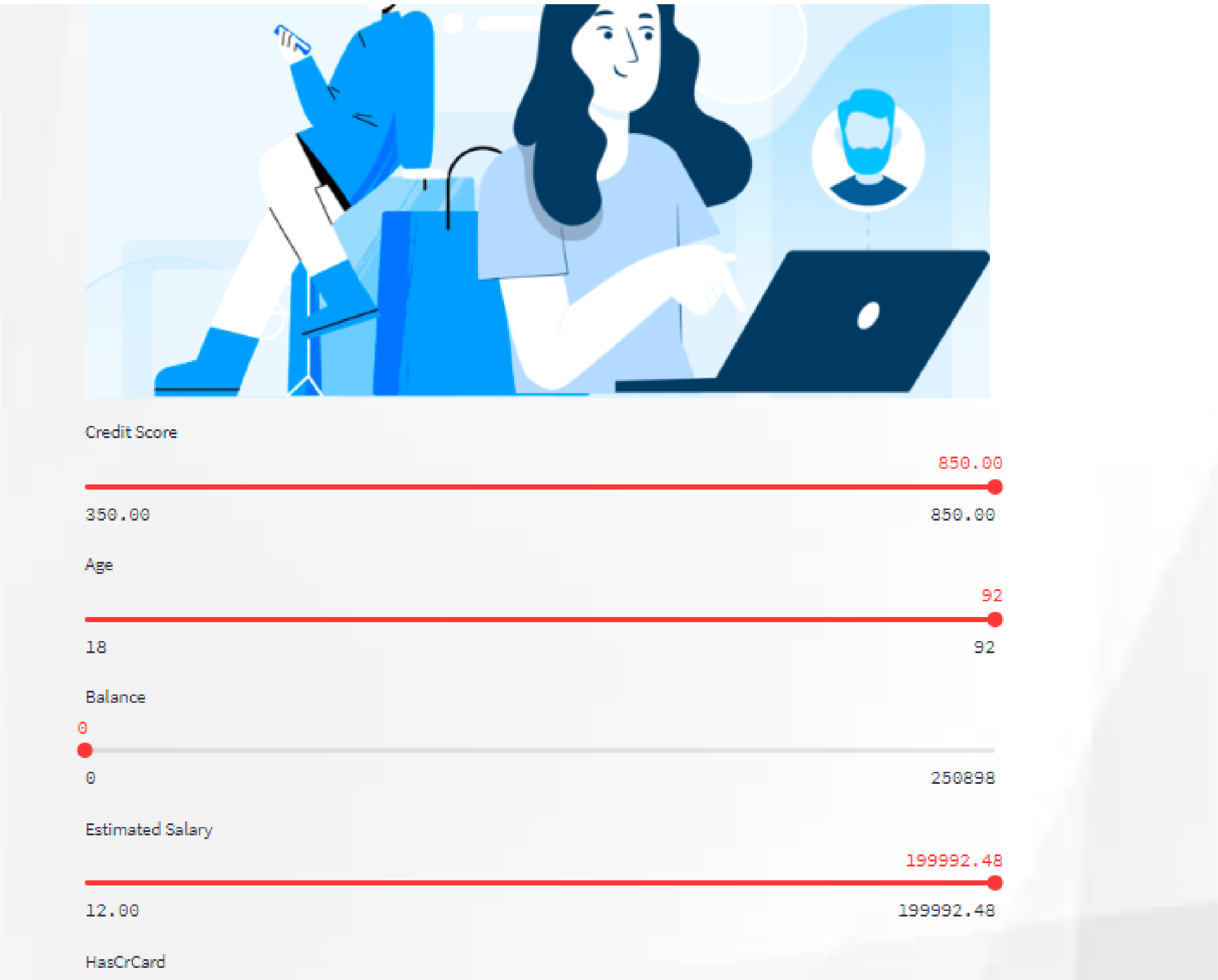
Screenshot 8 : Output predicted by the model for batch input



Input parameters

	CreditScore	Age	Balance	HasCrCard	IsActiveMember	EstimatedSalary
0	850	92	0	0	0	199,992.48

Screenshot 9 : PieGraph for the input Data to give the stats seemlessly



Credit Score

850.00

350.00 850.00

Age

92

18 92

Balance

0

0 250898

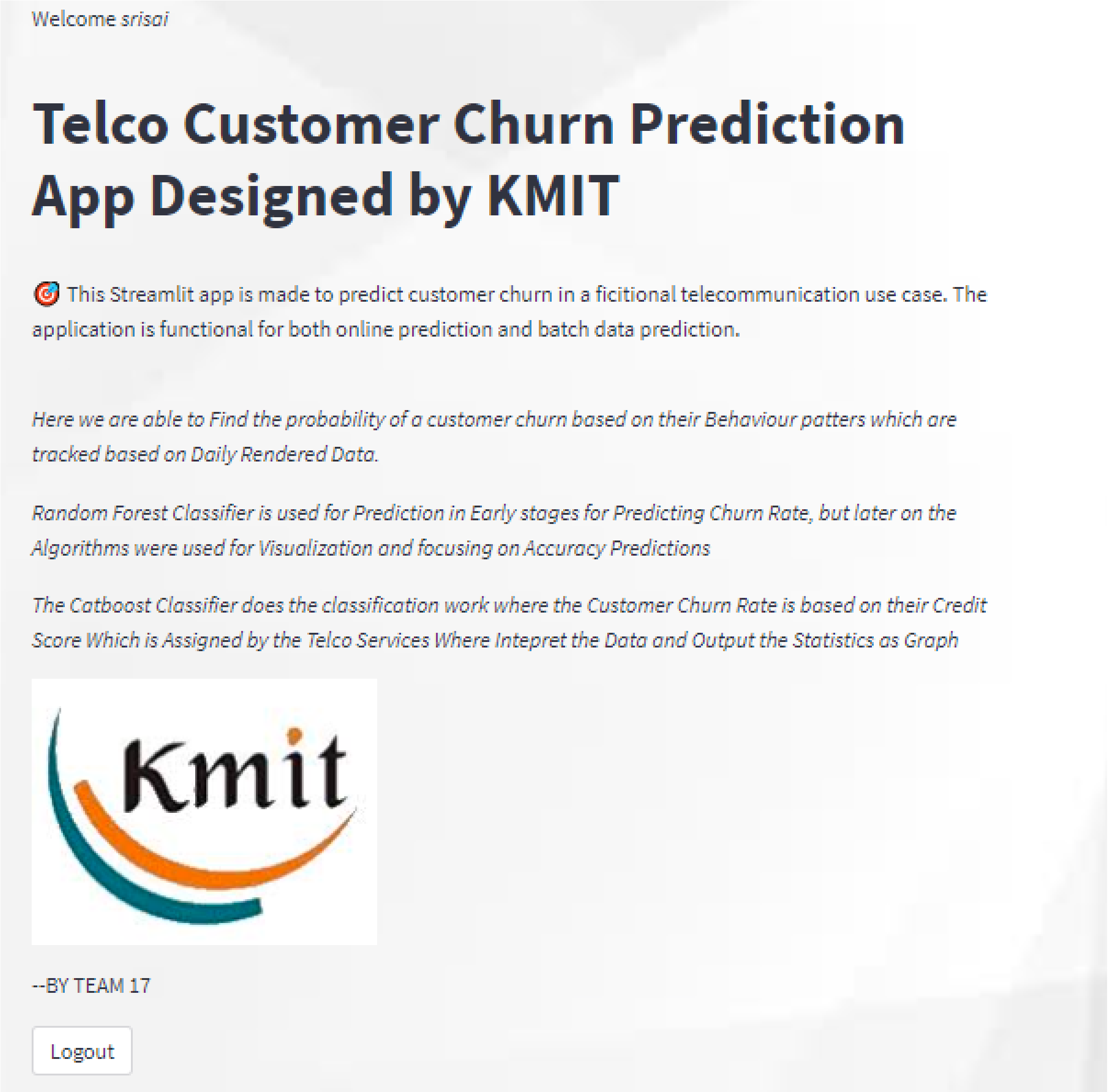
Estimated Salary

199992.48

12.00 199992.48

HasCrCard

Screenshot 10: Input of Credit Score for illustrating the stats and Check the Exciting States for Churn



Welcome *srisai*

Telco Customer Churn Prediction App Designed by KMIT

🎯 This Streamlit app is made to predict customer churn in a ficitional telecommunication use case. The application is functional for both online prediction and batch data prediction.

Here we are able to Find the probability of a customer churn based on their Behaviour patters which are tracked based on Daily Rendered Data.

Random Forest Classifier is used for Prediction in Early stages for Predicting Churn Rate, but later on the Algorithms were used for Visualization and focusing on Accuracy Predictions

The Catboost Classifier does the classification work where the Customer Churn Rate is based on their Credit Score Which is Assigned by the Telco Services Where Intepret the Data and Output the Statistics as Graph

--BY TEAM 17

Screenshot 11 : About Information regarding the Project

CHAPTER-6

6.1 Introduction to testing

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item. Who does Testing? It depends on the process and the associated stakeholders of the project(s). In the IT industry, large companies have a team with responsibilities to evaluate the developed software in context of the given requirements. Moreover, developers also conduct testing which is called Unit Testing. In most cases, the following professionals are involved in testing a system within their respective capacities:

- Software Tester
- Software Developer
- Project Lead/Manager
- End User Levels of testing include different methodologies that can be used while conducting software testing.

The main levels of software testing are:

- Functional Testing
- Non-functional Testing

Functional Testing This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for.

Functional testing of a software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

Software Testing Life Cycle The process of testing a software in a well planned and systematic way is known as software testing life cycle (STLC). Different organizations have different phases in STLC; however, the generic Software Test Life Cycle (STLC) for waterfall development model consists of the following phases.

1. Requirements Analysis
2. Test Planning
3. Test Analysis
4. Test Design

Requirements Analysis :In this phase testers analyze the customer requirements and work with developers during the design phase to see which requirements are testable and how they are going to test those requirements. It is very important to start testing activities from the requirements phase itself because the cost of fixing defects is very less if it is found in the requirements phase rather than in future phases.

Test Planning: In this phase all the planning about testing is done like what needs to be tested, how the testing will be done, test strategy to be followed, what will be the test environment, what test methodologies will be followed, hardware and software availability, resources, risks etc. A high level test plan document is created which includes all the planning inputs mentioned above and circulated to the stakeholders.

Test Analysis: After the test planning phase is over, the test analysis phase starts, in this phase we need to dig deeper into the project and figure out what testing needs to be carried out in each SDLC phase. Automation activities are also decided in this phase. If automation needs to be done for software products, how will the automation be done, how much time will it take to automate and which features need to be automated. Non functional testing areas(Stress and performance testing) are also analyzed and defined in this phase.

Test Design: In this phase various black-box and white-box test design techniques are used to design the test cases for testing, testers start writing test cases by following those design techniques, if automation testing needs to be done then automation scripts also need to be written in this phase.

6.1.1 Testing Objectives

The main objectives of software testing

- To find any defects or bugs that may have been created when the software was being developed
- To increase confidence in the quality of the software
- To prevent defects in the final product
- To ensure the end product meets customer requirements as well as the company specifications.

6.1.2 Testing Strategies

Software testing strategy is the planning done before testing commences and exercised systematically to test the software. The testing strategy could be developed by the project manager, or by the software engineers or it could even be a testing specialist.

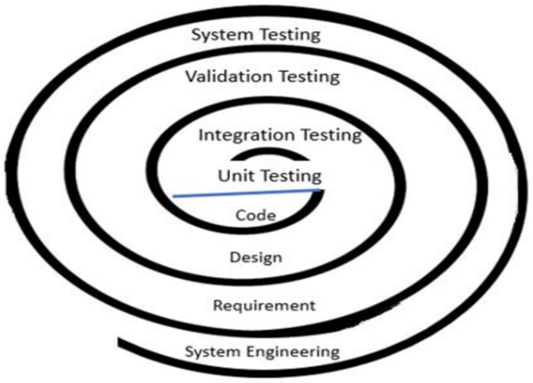
Developing a testing strategy for software is important because if testing is not conducted properly it would lead to wastage of time and effort and it would even be the case that some error or bugs remain undetected. Some general **characteristics** that should be considered while developing the testing strategy are as follow:

1. For successful testing, you should conduct the technical evaluation, it would reveal many of the error before the testing starts. It would help in correcting the technical error before the testing commences & would save time while testing the software.

2. Testing must start from the core of the software design and it must progress outward to incorporate testing of the entire software.

3. You should not follow the same strategy to test all the software. Appropriate testing strategies must be developed for different software engineering approach

Levels of Software Testing Strategies



Software Testing Strategy

Unit Testing

Unit testing focuses on testing the lowest component of the software individually which is also called unit. Unit testing involves the testing of each code segment to ensure that it functions properly.

Integration Testing

The unit components are integrated to implement the complete software. Integration testing involves testing the design structure of the software which include modeling and software architecture. It focuses on verifying whether functions of the software are working properly or not.

Validation Testing

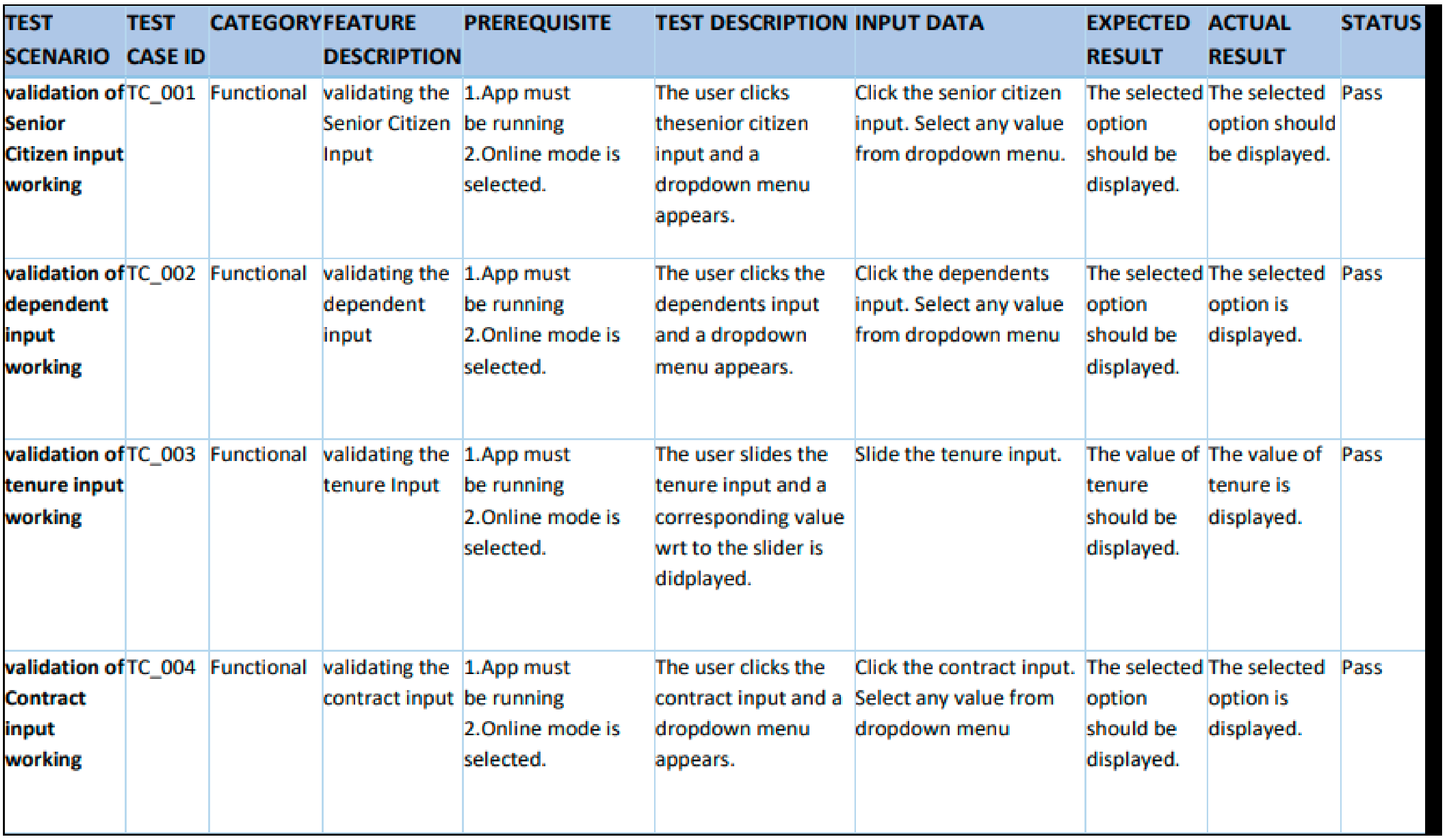
Validation testing focuses on the testing of software against the requirements specified by the customer.

System Testing

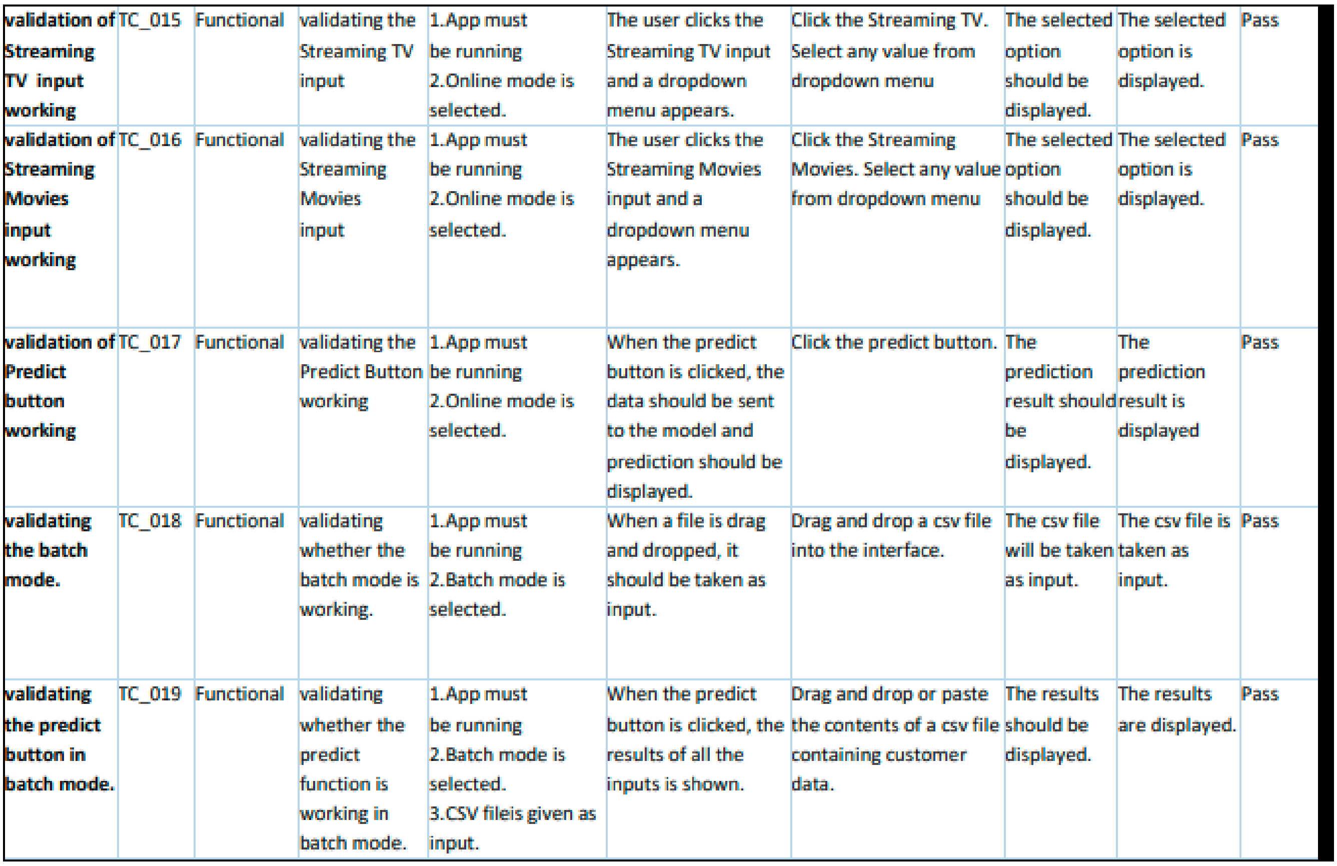
System testing focuses on testing the entire system as a whole and its other system elements. It tests the performance of the system.

6.2 Test cases

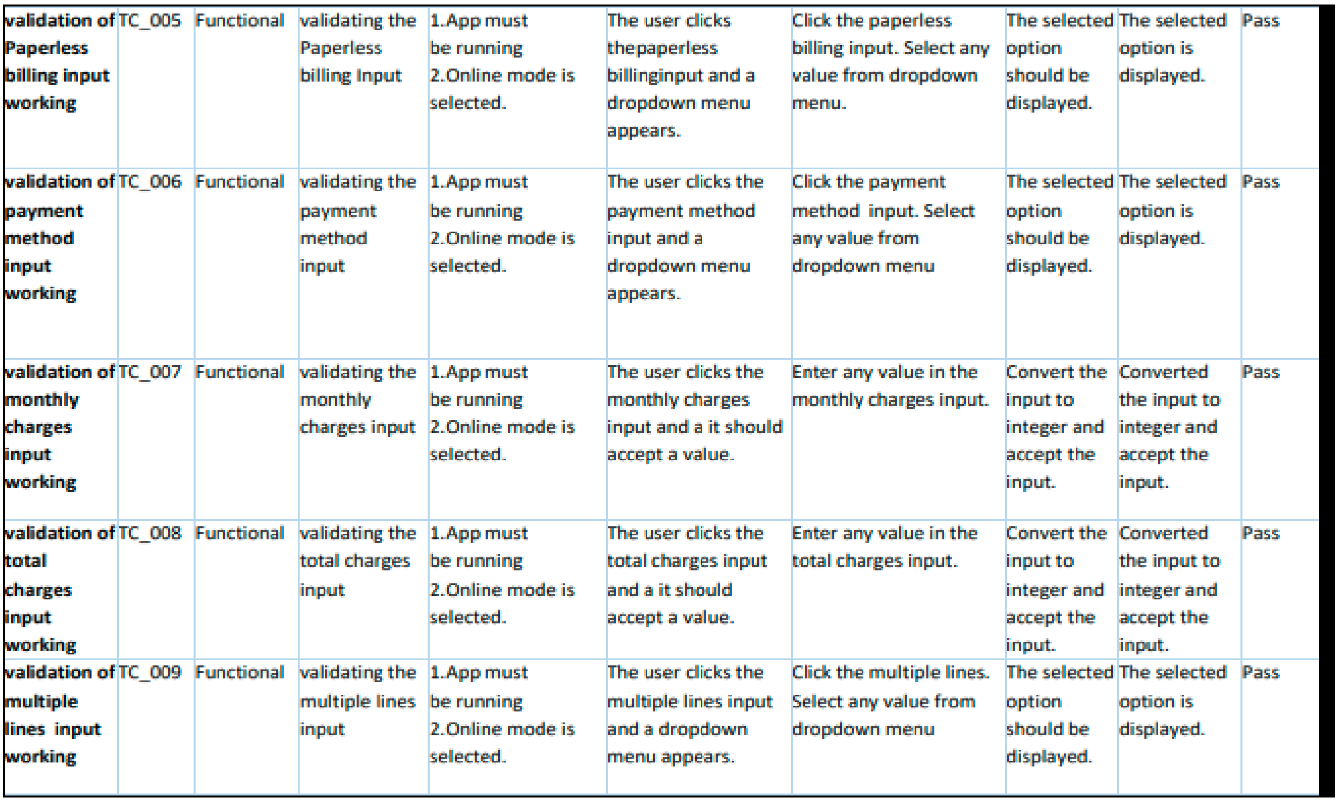
The following is a list of test cases used to perform functional testing at various stages of the project, in order to evaluate the efficacy of each module's behavior, and make immediate corrections upon identification of bugs.



TEST SCENARIO	TEST CASE ID	CATEGORY	FEATURE DESCRIPTION	PREREQUISITE	TEST DESCRIPTION	INPUT DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
validation of Senior Citizen input working	TC_001	Functional	validating the Senior Citizen Input	1.App must be running 2.Online mode is selected.	The user clicks thesenior citizen input and a dropdown menu appears.	Click the senior citizen input. Select any value from dropdown menu.	The selected option should be displayed.	The selected option should be displayed.	Pass
validation of dependent input working	TC_002	Functional	validating the dependent input	1.App must be running 2.Online mode is selected.	The user clicks the dependents input and a dropdown menu appears.	Click the dependents input. Select any value from dropdown menu	The selected option should be displayed.	The selected option is displayed.	Pass
validation of tenure input working	TC_003	Functional	validating the tenure Input	1.App must be running 2.Online mode is selected.	The user slides the tenure input and a corresponding value wrt to the slider is didplayed.	Slide the tenure input.	The value of tenure should be displayed.	The value of tenure is displayed.	Pass
validation of Contract input working	TC_004	Functional	validating the contract input	1.App must be running 2.Online mode is selected.	The user clicks the contract input and a dropdown menu appears.	Click the contract input. Select any value from dropdown menu	The selected option should be displayed.	The selected option is displayed.	Pass

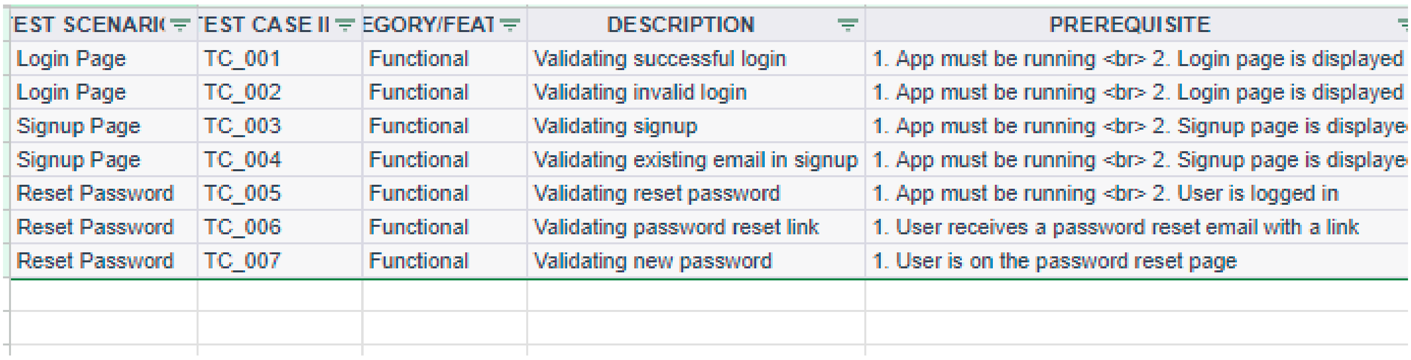


validation of Streaming TV input working	TC_015	Functional	validating the Streaming TV input	1.App must be running 2.Online mode is selected.	The user clicks the Streaming TV input and a dropdown menu appears.	Click the Streaming TV. Select any value from dropdown menu	The selected option should be displayed.	The selected option is displayed.	Pass
validation of Streaming Movies input working	TC_016	Functional	validating the Streaming Movies input	1.App must be running 2.Online mode is selected.	The user clicks the Streaming Movies input and a dropdown menu appears.	Click the Streaming Movies. Select any value from dropdown menu	The selected option should be displayed.	The selected option is displayed.	Pass
validation of Predict button working	TC_017	Functional	validating the Predict Button working	1.App must be running 2.Online mode is selected.	When the predict button is clicked, the data should be sent to the model and prediction should be displayed.	Click the predict button.	The prediction result should be displayed.	The prediction result is displayed	Pass
validating the batch mode.	TC_018	Functional	validating whether the batch mode is working.	1.App must be running 2.Batch mode is selected.	When a file is drag and dropped, it should be taken as input.	Drag and drop a csv file into the interface.	The csv file will be taken as input.	The csv file is taken as input.	Pass
validating the predict button in batch mode.	TC_019	Functional	validating whether the predict function is working in batch mode.	1.App must be running 2.Batch mode is selected. 3.CSV fileis given as input.	When the predict button is clicked, the results of all the inputs is shown.	Drag and drop or paste the contents of a csv file containing customer data.	The results should be displayed.	The results are displayed.	Pass

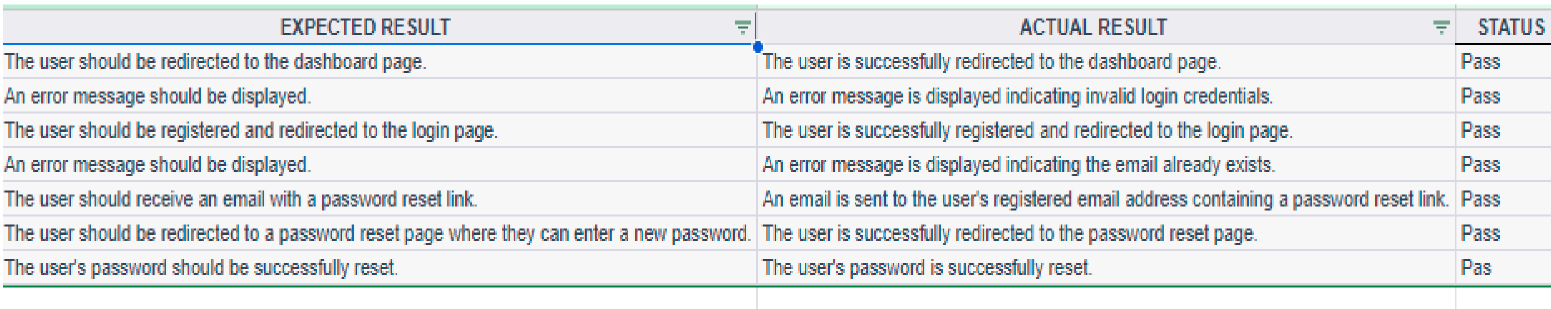


validation of Paperless billing input working	TC_005	Functional	validating the Paperless billing Input	1.App must be running 2.Online mode is selected.	The user clicks thepaperless billinginput and a dropdown menu appears.	Click the paperless billing input. Select any value from dropdown menu.	The selected option should be displayed.	The selected option is displayed.	Pass
validation of payment method input working	TC_006	Functional	validating the payment method input	1.App must be running 2.Online mode is selected.	The user clicks the payment method input and a dropdown menu appears.	Click the payment method input. Select any value from dropdown menu	The selected option should be displayed.	The selected option is displayed.	Pass
validation of monthly charges input working	TC_007	Functional	validating the monthly charges input	1.App must be running 2.Online mode is selected.	The user clicks the monthly charges input and a it should accept a value.	Enter any value in the monthly charges input.	Convert the input to integer and accept the input.	Converted the input to integer and accept the input.	Pass
validation of total charges input working	TC_008	Functional	validating the total charges input	1.App must be running 2.Online mode is selected.	The user clicks the total charges input and a it should accept a value.	Enter any value in the total charges input.	Convert the input to integer and accept the input.	Converted the input to integer and accept the input.	Pass
validation of multiple lines input working	TC_009	Functional	validating the multiple lines input	1.App must be running 2.Online mode is selected.	The user clicks the multiple lines input and a dropdown menu appears.	Click the multiple lines. Select any value from dropdown menu	The selected option should be displayed.	The selected option is displayed.	Pass

Login Module Test Cases



EST SCENARI	EST CASE II	EGORY/FEAT	DESCRIPTION	PREREQUISITE
Login Page	TC_001	Functional	Validating successful login	1. App must be running 2. Login page is displayed
Login Page	TC_002	Functional	Validating invalid login	1. App must be running 2. Login page is displayed
Signup Page	TC_003	Functional	Validating signup	1. App must be running 2. Signup page is displaye
Signup Page	TC_004	Functional	Validating existing email in signup	1. App must be running 2. Signup page is displaye
Reset Password	TC_005	Functional	Validating reset password	1. App must be running 2. User is logged in
Reset Password	TC_006	Functional	Validating password reset link	1. User receives a password reset email with a link
Reset Password	TC_007	Functional	Validating new password	1. User is on the password reset page



EXPECTED RESULT	ACTUAL RESULT	STATUS
The user should be redirected to the dashboard page.	The user is successfully redirected to the dashboard page.	Pass
An error message should be displayed.	An error message is displayed indicating invalid login credentials.	Pass
The user should be registered and redirected to the login page.	The user is successfully registered and redirected to the login page.	Pass
An error message should be displayed.	An error message is displayed indicating the email already exists.	Pass
The user should receive an email with a password reset link.	An email is sent to the user's registered email address containing a password reset link.	Pass
The user should be redirected to a password reset page where they can enter a new password.	The user is successfully redirected to the password reset page.	Pass
The user's password should be successfully reset.	The user's password is successfully reset.	Pas

6.2.1 Form Testing

Forms testing is a process that you conduct to test the quality of an online form on your website, checking for elements like copy, length and overall design. The purpose of using form testing is to

improve your conversion rates, which describes the percentage of people on your website who convert from visitors to customers. When you build a form properly, it can help you drive traffic through paid advertising and search engine optimization. Web forms are important tools for many websites because they serve as communication devices for customers, creating a connection between visitors and companies.

6.2.2 GUI Testing

GUI Testing is a software testing type that checks the Graphical User Interface of the Software. The purpose of Graphical User Interface (GUI) Testing is to ensure the functionalities of software application work as per specifications by checking screens and controls like menus, buttons, icons, etc.

GUI is what the user sees. Say if you visit guru99.com what you will see on the homepage is the GUI (graphical user interface) of the site. A user does not see the source code. The interface is visible to the user. Especially the focus is on the design structure, images that they are working properly or not.

CONCLUSION

Customer churn prediction plays a very important role in determining the success of any organization. Retaining the already acquired customers is very economical and takes lesser efforts compared to acquiring new customers. So it is very important to have effective mechanisms to analyze customer satisfaction and predict customers who would probably churn.

The proposed System uses various Data Mining, Data Science and Machine Learning techniques to gather valuable insights from the customer data which would help in formulation of ideas to make customers not churn.

We perform an exploratory data analysis (EDA) on the telecom dataset which consists of many attributes related to the customer's demographics, subscription models, services being consumed and payment details to identify the factors that affect the churn rate the most and then build a predictive model based on the results obtained from EDA and various machine learning algorithms like Random Forest Classifier and Decision Tree Classifier are used to build a predictive model. Finally, the Random Forest Classifier model achieved high accuracy (94%) for Rendered data Analysis from the Customer Data and Catboost Classifier for the Credit Score given by Behaviour patterns achieved (92%)and better predictions of customer churn compared to other four models.

We have also observed that factors like subscription model, tenure, monthly charges, total charges, senior citizens have a high impact on the churn whereas factors like gender, availability of phone service, having multiple lines does not affect the churn value considerably. This model can be further improved by including techniques to find out why exactly the customer is churning.

FUTURE ENHANCEMENTS

There is a lot of work that can be done in this domain. Since we are only predicting the churn status of the customers, the companies have to make note of these customers and provide them with schemes and plans which would help retain them. The formulation of these schemes can be easier if the company knows why exactly the customer is churning. The system can be developed to know the exact reason why the customer is churning and compare the competitor's offers to that of the company's offers. By analyzing this information, the system can be developed to recommend specific plans to specific customers which can hold them from churning

As of now We are Currently Choosing the Classifiers Based on the data input ,we may develop a version which automatically chooses the best Classifier by Interpreting the data type for producing Highest Accuracy,also we can bring up UI changes as well.

BIBLIOGRAPHY

1. Xin Hu, Yanfei Yang, Lanhua Chen, and Siru Zhu. "Research on a Customer Churn Combination Prediction Model Based on Decision Tree and Neural Network," 2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics.
2. Mohammad A. Hassonah, Ali Rodan, Abdel-Karim Al-Tamimi, and Jamal Alsakran. "Churn Prediction: A Comparative Study Using KNN and Decision Trees," 2019 Sixth HCT Information Technology Trends (ITT).
3. Mykola Malyar, Mykola Robotyshyn M.V, and Maryana Sharkadi. "Churn Prediction Estimation Based on Machine Learning Methods," 2020 IEEE 2nd International Conference on System Analysis & Intelligent Computing (SAIC).
4. Abinash Mishra and U. Srinivasulu Reddy. "A Comparative Study of Customer Churn Prediction in Telecom Industry Using Ensemble Based Classifiers," 2017 International Conference on Inventive Computing and Informatics (ICICI).
5. Pushkar Bhuse, Aayushi Gandhi, Parth Meswani, Riya Muni, Neha Katre. "Machine Learning Based Telecom-Customer Churn Prediction," 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS).
6. S. Stehani, N. Karunya, D.R.J.B. Ranjan, Sgara Sumathipala, T. C. Sandanayake. "Customer Churn Reasoning in Telecommunication Domain," 2020 International Conference on Image Processing and Robotics (ICIP).
7. V. Geetha, A. Punitha, A. Nandhini, T. Nandhini, S. Shakila, R. Sushmitha. "Customer Churn Prediction In Telecommunication Industry Using Random Forest Classifier," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN).
8. Qiu Yihui, Zhang Chiyu. "Research of Indicator System in Customer Churn Prediction for Telecom Industry," The 11th International Conference on Computer Science and Education (ICCSE 2016).
9. Pan Tang. "Telecom Customer Churn Prediction Model Combining K-means and XGBoost Algorithm," 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICM CCE).
10. Mehpara Saghir, Zeenat Bibi, Saba Bashir, Farhan Hassan Khan. "Churn Prediction using Neural Network based Individual and Ensemble Models," 2016 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST).

REFERENCES

1. Streamlit Authenticator GitHub repository. Available at: https://github.com/mkhorasani/Streamlit-Authenticator. Retrieved in 2022.
2. Streamlit documentation. Available at: https://docs.streamlit.io/. Retrieved in 2021.
3. Introduction to Random Forest in Machine Learning. Available at: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/. Retrieved in 2020.
4. CatBoost Classifier Python Reference. Available at: https://catboost.ai/en/docs/concepts/python-reference_catboostclassifier. Retrieved in 2021.
5. IEEE Xplore document: Churn Prediction using Machine Learning Algorithms. Available at: https://ieeexplore.ieee.org/document/9558876. Retrieved in 2021.