Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

SCHOOL OF COMPUTER SCIENCE AND STATISTICS

CUSTOMER CHURN PREDICTION USING SYNTHETIC DATA

by SHRUTI KATHURIA, BAI

B.A.I COMPUTER ENGINEERING
Final year project- APRIL 2023
Supervisor- PROF. MERIEL HUGGARD

FINAL YEAR PROJECT REPORT
IN COMPLETION OF THE BAI COMPUTER
ENGINEERING SCHOOL OF COMPUTER SCIENCE &
STATISTICS TRINITY COLLEGE DUBLIN, IRELAND

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Name: _______________

Date: _____________

Acknowledgment

I would like to express my gratitude to Prof. Meriel Huggard for her valuable guidance and motivation during this dissertation. In particular, Prof. Meriel Huggard, has provided me with continuous patient guidance and support at every stage of this dissertation.

I am grateful to my sister Kriti for her companionship, encouragement, and inspiration.

I would like to thank my parents for providing me with the financial resources and enthusiasm to come and study at Trinity. Finally, I would like to thank my friends, who stand by me in all highs and lows.

University of Dublin, Trinity College April 2023

SHRUTI KATHURIA

Abstract

Customer churning has various problems in a number of different industries. Due to the loss of customers the companies not only have a lower revenue but there is a vast effect on the reputation and client number. Therefore, it has become crucial for businesses to anticipate customer churn. In this project, we want to create a model for predicting customer churn using artificially generated data.

The project will involve creating artificial data that closely resembles the traits of actual customer data. To create a dataset that accurately reflects the customer data, we will develop a framework for the generation of synthetic data. To make sure the synthetic data is appropriate for machine learning algorithms, it will be pre-processed and given engineered features. Using the simulated data, we will assess how well different machine learning algorithms predict customer churn. The model's performance will be contrasted with that of conventional models developed using real-world data. Later we will deploy the maximum accuracy machine learning model into a web application in a flask for companies to input data and predict customer churn prediction.

The project's findings will shed light on how well artificial data can be used to predict customer churn. Businesses will benefit from better customer retention strategies and more informed choices. Additionally, the framework developed for the project can be applied to numerous other applications where a lack of data is a bottleneck. The project's contribution goes beyond just predicting customer churn; it also offers a general method for creating synthetic data that can be used in other situations.

Table of Contents

TABLE OF FIGURES

Customer Churn Prediction using Synthetic Data

The purpose of this dissertation is to develop and evaluate a customer churn prediction model using fictitious data for a telecom firm. The research looks into the issue of sensitive or limited real-world data as well as the efficacy of synthetic data in forecasting customer attrition. Using a data generating tool, a dataset replicating client behavior will be created for this project, and machine learning methods will be used to create and evaluate prediction models. The project's results may have a significant effect on businesses whose success depends on retaining clients.

The project will go through a number of phases, including data preparation and cleaning, feature engineering, model selection and assessment, and outcome interpretation. A number of machine learning methods including logistic regression, decision trees, and others will be used to create and evaluate, the prediction models.

Synthetic data has various advantages including the ability to build datasets that reflect a wide range of customer activities, a decreased risk of privacy violations, and greater control over the data generation technique. The dissertation will investigate the utility of employing fake data to estimate customer attrition and the findings will be compared to models constructed using actual data.

The expected outcome of this dissertation is a better understanding of how synthetic data may be used to estimate customer churn and increase customer retention rates for organizations. The results of this study might have significant effects on businesses whose success depends on attracting and keeping consumers.

Chapter 01 Introduction

This chapter explores the overview of customer churn prediction using synthetic data as well as the reasons behind the project's selection. We give a brief summary of the project objectives in part 1.3 and the general organization of the dissertation in section 1.4.

1.1 Research Background

Any business that wants to improve customer retention and profitability must view customer churn prediction as a crucial element of customer relationship management [1] [2]. A company's bottom line may be significantly impacted by the capacity to identify clients who are about to quit and take proactive actions to maintain them.

Customer behavior-related information, including purchase history involved with the company's products or services, and customer support contacts are often gathered and examined as part of the process of predicting customer churn (Narasimhan & Kumar, 2018). Then, using machine learning algorithms, prediction models that can detect trends and indicate which clients are most likely to churn are created (Narasimhan & Kumar, 2018; Hadden & Bove, 2017).

Data availability and quality are important challenges in developing customer churn prediction models. Companies typically lack appropriate data on client behavior or have untrustworthy or erroneous data (Narasimhan & Kumar, 2018). Synthetic data products can be a useful technique for addressing these issues [3] .

Synthetic data creation uses machine learning algorithms to produce precise data that closely mirrors the consumer information that companies already have on hand (Zaki, 2020). This approach can be particularly useful when data is scarce or when access to actual consumer data is hindered by privacy issues [4]. For a project concentrating on customer churn prediction, using synthetic data, large amounts of data would be created using machine learning methods, and this data would then be used to train and test predictive models [5] .

There are a number of ways for producing synthetic data, including deep learning models variational autoencoders & generative adversarial networks (GANs) (Kshirsagar et al., 2019). The goal of the study is to identify the data components and prediction models that are most accurate and valuable in forecasting customer attrition.

This initiative to anticipate customer turnover using synthetic data has the potential to provide businesses with valuable information about customer behavior and aid firms in adopting proactive actions to retain customers and boost profitability (Narasimhan & Kumar, 2018; Hadden & Bove, 2017).

What is Customer Churning?

Customer Churning [6] is the practice of identifying consumers who are most likely to discontinue using a product or service by applying machine learning algorithms and data analysis approaches. Churn Prediction's objective is to assist organizations in taking proactive steps to keep consumers before they depart.

Businesses can prevent customer attrition by taking proactive actions like targeted marketing campaigns, individualized offers, and improved customer service by anticipating client churn. One approach to tackling this problem is the use of synthetic data.

What is Synthetic Data?

Synthetic data a computer-generated representation of actual data can be created to produce large datasets for machine learning and other data-driven applications. The benefit of utilizing synthetic data is that it can be produced in huge volumes & offers, a more reliable dataset for developing prediction models [7].

Synthetic data has been shown to be an excellent tool for predicting customer attrition in previous studies. Many studies have built predictive models using approaches such as decision trees random forests and neural networks. When it comes to identifying consumers who are in danger of leaving a company, these models have demonstrated excellent levels of accuracy (Wang et al., 2020).

1.2 Research Project

Predicting client attrition is a common use of data science and machine learning. It comprises identifying consumers, who are most likely to stop utilizing a service or product which may be helpful information for organizations in order to retain customers and increase revenue.

Synthetic data is data, that is generated artificially by statistical models or algorithms. It can be used to secure sensitive information as a substitute for genuine data when it is unavailable or to modify the dataset for testing and review.

Synthetic data may be used to produce additional data that can be utilized, to improve the performance of machine learning models in the context of customer churn prediction. By developing synthetic data, that resembles the patterns and qualities of actual customer data, machine learning models may be taught to generate more exact predictions about, which customers are most likely to churn.

Numerous studies have looked into the prediction of customer churn using artificial data. For instance, from 2021 used synthetic data to boost the effectiveness of machine learning models for predicting customer churn in a telecom company. The researchers created synthetic data that mimicked the patterns of actual customer data using a Generative Adversarial Network (GAN) and then used this synthetic data to expand the training dataset. The findings demonstrated that using synthetic data increased the predictive power of machine learning models for customer churn.

A different study, (Li, 2021) addressed the issue of class imbalance in predicting customer churn. The researchers balanced the dataset by using synthetic data for the minority class (churned customers), which was created using the synthetic minority oversampling technique (SMOTE). The findings demonstrated that using synthetic data enhanced the ability of machine learning models to predict customer churn.

Overall, research on predicting customer churn using synthetic data has produced encouraging outcomes for raising the precision and effectiveness of machine learning models. To investigate the potential of synthetic data for predicting customer churn in various industries and contexts, more study is necessary.

1.3 Motivation

According to [8], customer churn prediction with synthetic data can provide organizations with a useful tool for detecting clients who are likely to leave and keeping them by taking precautionary actions. Since it may result in a drop in income a loss of market share & a tarnished reputation for the brand, customer churn is a serious issue for firms in many industries. If businesses want to boost customer satisfaction and their financial line, client churn must be predicted and minimized.

Due to the lack of complete and accurate data, gathering and analyzing customer data for churn prediction can be difficult. The accessibility of actual customer data may be hampered by privacy issues, which reduces the efficiency of conventional machine learning algorithms. These difficulties can be overcome by using synthetic data generation techniques, which also make it possible to create effective and precise churn prediction models.

Businesses can use a customer churn prediction project that uses synthetic data as a potent tool to boost customer retention rates, improve marketing plans & boost sales. Businesses can prevent customers from leaving by offering personalized promotions or enhancing customer service when they can accurately identify, which customers are likely to leave. In the end, this project may assist companies in increasing client satisfaction and loyalty, which will result in higher profitability and long-term success (Ji & Chen, 2020).

The basic motivation of the project came from the fact that businesses need to save customer from leaving the company and reduce their losses and after a lot of research I decided to work on this project.

1.4 Problem Statement

To anticipate client attrition using artificial data is the project's stated problem. The goal is to specifically create a prediction model that can recognize consumers who are likely to leave and take preemptive action to keep them. Customer churn may result in lost income and higher expenditures for businesses. By using a predictive model to identify consumers who are at risk of churning, businesses can take preventative action to keep those customers.

The solution is we need to build a system that tells us which customers have a chance to leave the company. For this purpose, we need to train and train a machine learning model also we need some past data for training our model. After training, we need to deploy our model that comes in the shape of the website so that users can easily use it and get results about customer churning. This method involves a number of phases.

•Collect data.

•Train the Machine Learning

•Model.

•Deploy the model in Flask.

1.5 Project objectives

My project to predict customer churn using artificial data has the following primary goals:

1)To create superior synthetic data that closely resembles actual customer data so that precise and effective churn prediction models can be created.

2)To identify the key data and traits significant to predicting customer churns such as a customer's purchase history, involvement with a product or service, and demographics.

3)To create and evaluate various machine learning models in order to determine which ones are most effective and accurate at predicting customer churn.

4)To provide organizations with a tool to aid them in identifying clients who may be considering leaving and taking proactive measures to maintain them, so boosting client happiness, loyalty, and profitability.

5)To overcome the obstacles of limited and incomplete data and privacy concerns in order to increase the overall effectiveness of customer churn prediction.

6)To gather knowledge about consumer behavior and preferences in order to enhance marketing efforts and customer service.

7)To offer a repeatable process for creating fictitious data and creating churn prediction models that can be used in other fields and scenarios.

By attaining these objectives, a customer churn prediction project utilizing synthetic data may help firms improve their customer retention rates, streamline their marketing programs, and increase their profitability.

1.6 Dissertation Overview

My Dissertation is divided into 6 chapters overall.

Chapter 1- Introduction -This chapter introduces the idea and concept of the project and discusses the motivation and purpose of this dissertation.

Chapter 2- Background- In this chapter we will go through the literature review, the advantages and challenges of using synthetic data. We would also study the business Impact Analysis for this project.

Chapter 3- Design & Methodologies- In this Chapter we will study how preprocessing of data is done and understand some graphs.

Chapter 4- Implementation and Results- In this Chapter we will study Exploratory Data analysis and some machine learning models.

Chapter 5- Evaluation- In this Chapter we will study the Accuracy and cross validation of the results. Further we will learn how to create a webpage in flask and how to do customer churn prediction on that website.

Chapter 6- Conclusions and future work - This chapter provides a summary of the dissertation report, highlighting the major contributions made by this project to the existing body of work as well as the system's limitations. It also suggests areas for improvement and further research in the future.

Chapter 02 Background

In this chapter, we will go through the literature review and the advantages and challenges of using synthetic data. Later we will go through the Business Analysis of the project.

2.1 Literature Review

Customer churn prediction has grown in importance as a research area in the fields of marketing and customer relationship management (CRM) in recent years. Companies are more and more interested in knowing why customers leave and how they may reduce customer turnover as consumer data becomes more readily available. Using synthetic data, which enables the development of extra data points typical of the customer population, is one technique to increase the precision of customer churn prediction models. The purpose of this research is to examine the use of synthetic data in customer churn prediction and assess its efficacy in raising prediction model accuracy.

Numerous studies have looked into the usage of various methods to forecast client churn. These studies, however, have mainly relied on actual consumer data, which has its volume and quality limitations. In order to predict customer attrition, academics have recently begun to investigate the use of synthetic data.

One study, [9] boosted the precision of models for predicting customer attrition. When compared to models trained purely on real-world data, the authors discovered that the use of synthetic data produced a greater accuracy rate.

Similarly, this, [10] enhances customer attrition prediction in a telecom company. The scientists discovered that using synthetic data produced a prediction model that was more accurate, with an AUC score of 0.827 as opposed to 0.759 for models trained exclusively on real-world data.

A subsequent study, [11] investigated how to better predict customer attrition in an e-commerce company using synthetic data produced by a Gaussian mixture model (GMM). The scientists found that employing synthetic data yielded a more accurate prediction model with an F1-score of 0.77 compared to models trained, only on real-world data which had an F1-score of 0.72.

These studies collectively indicate that using synthetic data can enhance the precision of customer turnover prediction models, and this strategy shows potential for businesses aiming to lower customer churn rates.

2.2 Introduction

A Customer Churn Prediction System using synthetic data and deployed using Flask is a machine learning model that is trained and tested using synthetic (artificial or generated) data and then deployed as a web application using the Flask web framework (Koohikamali, Salamzadeh, & Moghaddam, 2020).

In this method, I used pandas, a synthetic data generator to generate a collection of customer information that closely resembles real-world customer data. After then, a machine learning model trained on this dataset would be used, to forecast client attrition. I developed a model utilizing several machine learning techniques such as Random Forest, Decision Tree, and Logistic Regression and then evaluated the algorithm's precision before deciding, on the best one for prediction. After the model has been trained a web application utilizing the trained model would be released using Flask a Python web framework that is small and lightweight.

This would allow users to access the model through a web interface and make predictions about customer churn. Because it is simple to use lightweight and versatile, Flask is a popular choice for deploying machine learning models. It also makes it simple, to integrate Flask with other web technologies like HTML, CSS & JavaScript. The model may be trained and tested using synthetic data, which has a number of benefits, including resolving the issue of data privacy and being useful in situations, when actual data is either not accessible or insufficient (Koohikamali et al., 2020).

2.3 Advantages of using Synthetic Data

Synthetic data can be created by a computer program which makes it relatively affordable to manufacture. Comparing this to gathering and processing real-world data can result in huge financial savings for businesses. [12].

PRIVACY - Synthetic data can be used to safeguard the privacy of individuals or organizations. Because synthetic data has no personally identifying information it may be shared and evaluated without jeopardizing privacy.

DIVERSITY - Using synthetic data, scenarios that would be difficult to recreate in the actual world may be constructed. This is especially useful in fields such, as artificial intelligence (AI) and machine learning where a diverse set of training data, is required.

SCALABILITY - Synthetic data generation is a quick and efficient process, making it simple to increase production as necessary. For businesses that must produce vast amounts of data for testing and analysis, this can be extremely helpful.

CONTROL - Synthetic data may be entirely controlled in terms of its properties. Businesses may now create data that is suited to their requirements rather than depending on pre-existing sources that might not be sufficient for their objectives.

2.4 Challenges of using Synthetic Data

The technique of identifying consumers who are most likely to discontinue using a product or service is known as customer churn prediction. Artificially produced data, or synthetic data, is used to resemble real-world data. While employing synthetic data for customer churn prediction offers certain benefits, it can also present some difficulties. Here are a few potential difficulties:

1. **Absence of diversity:** A set of guidelines or patterns is used to generate synthetic data. The generated synthetic data might not fully capture the range of variability inherent in real-world data if these rules are not sufficiently varied. Predictions could become skewed or erroneous as a result.
2. **Overfitting:** When using synthetic data, it's important to make sure the model hasn't gotten overly matched to it. The model should thus generalize well to fresh untested data. If the model is too closely tailored to the synthetic data, it could not perform well on real-world data.
3. **Restricted applicability:** It's possible that the assumptions and standards utilized to produce synthetic data weren't always true. This suggests that not every real-world circumstance may be represented by the generated synthetic data. As a result, the model

created using fake data could only be helpful in particular circumstances.

4. **Data quality:** The quality of synthetic data depends on the rules that were employed to produce it. The quality of the synthesized data may be subpar if these guidelines are faulty or lacking. This can result in faulty insights and inaccurate predictions.

In general, using synthetic data to anticipate customer turnover may be beneficial, but it's important to be aware of any potential disadvantages as well. To ensure that the created model generalizes successfully to new, unobserved data and that the synthetic data accurately reflect the real-world data, these two criteria must be met.

2.5 Scope

The scope of customer churn prediction refers to the range of applications and use cases for which the model is designed and can be applied. Some examples of the scope of customer churn prediction include:

i) Identifying customers who are at risk of leaving a company, so that proactive measures can be taken to retain them.

ii) Analyzing the causes that cause customer turnover so that these problems may be resolved such as product dissatisfaction or bad customer service.

iii) Enhancing consumer loyalty and minimizing, the effects of missed sales on a company's revenue and profitability.

iv) Assisting businesses in making smarter choices on marketing customer service and product development

v) Identifying patterns and trends in customer data that can inform business strategy and improve the overall customer experience.

vi) Using synthetic data to overcome data privacy issues and also to generate, new data for training and testing models where real data is not available or insufficient.

vii) Helping companies to prioritize their retention efforts by identifying the most valuable customers who are at the highest risk of churning.

viii) Identifying the most effective retention strategies such as targeted promotions or personalized communication based on customer data and behavior.

2.6 Initial Investigation

We must first do an initial investigation of the customer churn prediction system using synthetic data before moving on to the deployment stage. Data selection is the initial stage since data are required to train every machine learning model.

2.6.1 Dataset Selection

At first I researched a lot about which dataset to pick for the analysis. Working on time series data to learning and discovering about them I came through this dataset.

I have selected the (BankChurners.csv) dataset from Kaggle. Here is the link to the dataset.

https://www.kaggle.com/code/shiviyadav/customer-churn-prediction-eda-ml-techniques/data?select=BankChurners.csv [13].

I have used a software called Jupyter notebook for my project. This project can also be run on visual studio code.

2.6.2 Dataset Columns

Customer attrition is a challenge for business management with a portfolio of consumer credit cards. In order to anticipate which clients are most likely to stop buying from them, they want to analyze the data to determine the cause of this. Here are the dataset columns:

i CLIENTNUM

Client number, Unique identifier for the customer holding the account

ii. Attrition_Flag

Internal event (customer activity) variable - if the account is closed then 1 else 0

iii. Customer_Age

Demographic variable - Customer's Age in Years

iv. Genders

Demographic variable - M=Male, F=Female

v. Dependent_count

Demographic variable - Number of dependent

vi. Education_Level

Demographic variable - Educational Qualification of the account holder (example: high school, college graduate, etc.)

vii. Marital_Status

Demographic variable - Married, Single, Divorced.

viii. Income_Category

Demographic variable - Annual Income Category of the account holder (<

$40K, $40K - 60K, $60K - $80K, $80K-$120K).

ix. Card_Category

Product Variable - Type of Card (Blue, Silver, Gold)

x. Months_on_book

Period of relationship with the bank.

xi. Total relationship count

Total no. of products held by the customer.

xii. months_inactive_12_mon

No. of months inactive in the last 12 months

xiii. contact_count_12_mon

No. of Contacts in the last 12 months

Xiv. Credit limit

Credit Limit on the Credit Card

xv. Total_revolving_bal

Total Revolving Balance on the Credit Card

2.6.3 Why bankchurners.csv dataset?

I have selected the bankchurners.csv dataset (Kaggle, n.d.) because it provides a large and diverse data set of customer information that can be used to train and test predictive models. The collection comprises a variety of bank customers' demographic and account details, as well as data on their credit card usage & behavior which may be used to spot patterns and trends linked, to customer turnover.

It's also important to note that the bank churn dataset is frequently utilized since banking is one of the businesses that is significantly impacted, by customer turnover and because this particular dataset has been the subject of a lot of studies. As a result, it is an established set in the industry and may be used as a benchmark dataset to assess how well new models or algorithms perform.

2.6.4 Programming Language Used:

The Programming language that I have used for my Project is Python. Some important characteristics make Python easy to use:

Simplicity, Efficiency, Security, Flexibility.

2.7 Business Impact Analysis (BIA)

Business Impact Analysis (BIA) [14] is a method that aids organizations in determining and evaluating the potential effects of an incident or disturbance on their operational processes. Quantifying prospective effects and figuring out how crucial particular corporate operations, systems, and assets are the two main objectives of BIA. A bigger business continuity planning (BCP) strategy which helps businesses prepare for and respond to disruptive events like natural disasters, cyberattacks, and other incidents that might affect their ability to function, occasionally includes BIA as a component.

2.7.1 Business Impact Analysis for Customer Churn Prediction using Synthetic data.

Here is how we can apply Business Impact Analysis (BIA) to customer churn prediction using synthetic data:

•Define the scope and objective

Defining the BIA's scope and goal is the initial stage. In this situation, the goal is to evaluate the financial effects of customer churn on a hypothetical e-commerce company and to determine which client categories are most important to the company.

•List the data sources.

Finding the data sources that will be used for the BIA is the next stage. In order to complete this project we will use synthetic consumer data which will include details on customer demographics purchasing patterns and engagement indicators.

•Complete a risk analysis.

The following phase involves analyzing customer data to determine the risk of client attrition. Finding patterns and trends in the data that might point to a higher chance of churn includes doing this. For example, we may learn that customers, who haven't purchased anything in the past three months are more likely to depart.

•Assess the effects of client attrition.

We can then determine the probable financial effect of losing the at-risk consumers after identifying them. Analyzing indicators like average revenue per client, profit margin, and cost of customer acquisition is necessary for this. For illustration let's say that each customer makes an average of $100 per month and that it costs, $50 to bring in a new client. We may anticipate a monthly revenue loss, of $10,000 and an increase, in client acquisition expenditures of $5,000 if we lose 10% of our customer base.

•Set customer segment priorities

We can then order client categories according to importance to the business based on the risk assessment and impact analysis. For instance, we can discover that clients with high average spending are the most beneficial to the company, therefore we should concentrate on keeping them.

•Create a mitigation strategy

Finally, we may create a strategy to lessen the effects of client attrition. This could entail building loyalty programs to reward high-value consumers, enhancing customer engagement and happiness, or focusing marketing efforts on at-risk clients.

Overall, we may better assess the potential financial impact of customer churn by utilizing synthetic data and BIA approaches and creating a data-driven strategy to lessen that impact.

Chapter 03 Design and Methodologies

This chapter explains the data reading, data selection, and data pre-processing, in the data. We shall discover the definitions of these words and how to apply them to the project.

3.1 IMPLEMENTATION PHASE 1

My implementation phase 1 includes data selection, data reading, and data pre-processing.

Data Selection: Selecting relevant features or variables from a bigger dataset that is beneficial for a particular investigation is known as data selection. [15]. Finding factors that significantly affect the analysis's results and eliminating redundant or irrelevant variables that don't add anything to the analysis are both parts of the data selection process. This makes the data more manageable by reducing its dimensionality.

Data Reading: Accessing and importing data from a data source is known as data reading. Data from several file formats, including CSV, Excel, SQL, etc., must be obtained and loaded into a computer for additional analysis. Recognizing the data structure and ensuring that the data is in the right format for processing are essential while reading data. I have accessed the Panda Library to read my data. I glance at the first five rows, of the dataset to get a sense of what our dataset is and what the values are. We in this project have read the csv file by pandas.

Data pre-processing: Data preprocessing is the process of preparing raw data for analysis by cleaning and turning it, into an appropriate format. This necessitates a variety of techniques, such as feature engineering, feature scaling, normalization, and data purification. Data pre-processing strives to guarantee that the data is correct consistent and ready, for analysis as well as to get rid of any biases or mistakes that might impact the results of the research. This stage determines how well and effectively the machine learning algorithms can function.

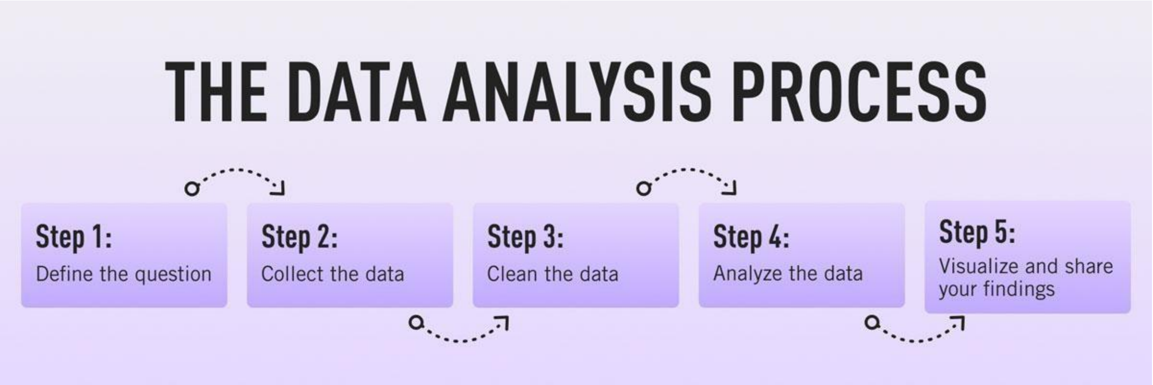
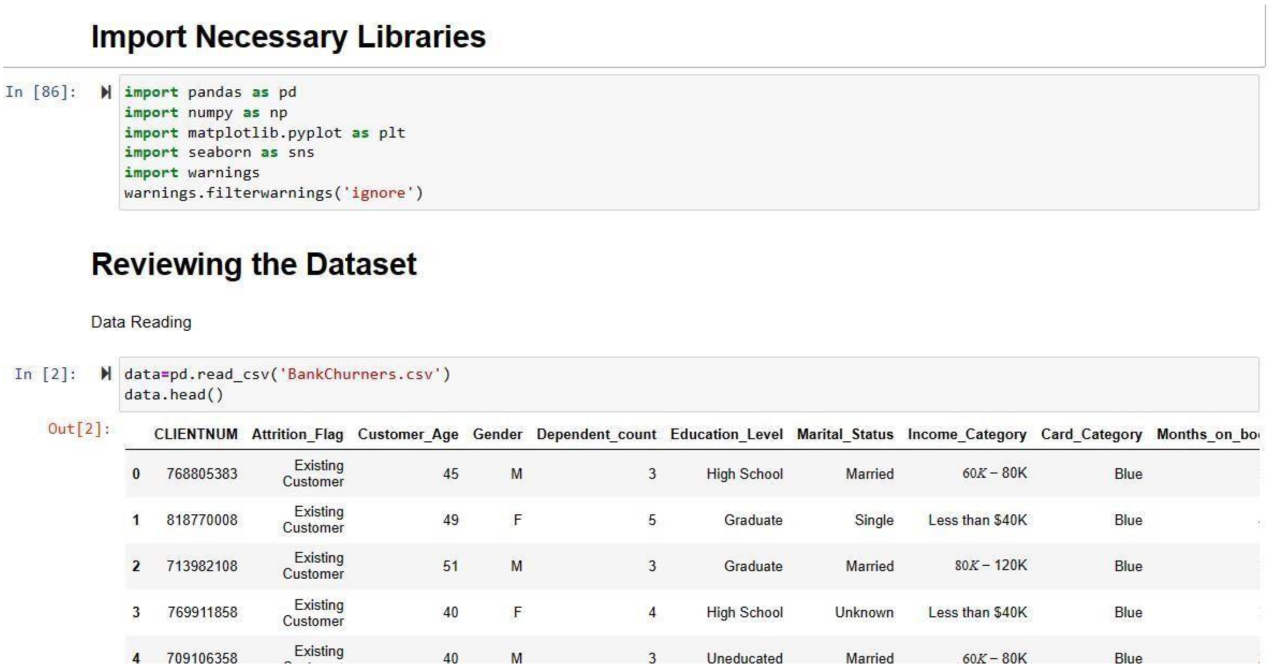


Figure 3.1 Data Analysis project

3.2 Importing Libraries

I have imported different libraries like Pandas, NumPy, matplotlib & seaborn. These libraries are necessary to complete our tasks such as Data Analysis, Data preprocessing, and Algorithm Applying.



Import Necessary Libraries

```
In [86]:  import pandas as pd
          import numpy as np
          import matplotlib.pyplot as plt
          import seaborn as sns
          import warnings
          warnings.filterwarnings('ignore')
```

Reviewing the Dataset

Data Reading

```
In [2]:  data=pd.read_csv('BankChurners.csv')
         data.head()
```

Out[2]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_bo
0	768805383	Existing Customer	45	M	3	High School	Married	60K – 80K	Blue	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than $40K	Blue	
2	713982108	Existing Customer	51	M	3	Graduate	Married	80K – 120K	Blue	
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than $40K	Blue	
4	709106358	Existing	40	M	3	Uneducated	Married	60K – 80K	Blue	

Figure 3.2 Library's imported

.

3.2.1 Reviewing the dataset

This is the very first step in model training and it includes Data Reading, Data Exploration. I started my project with the implementation of importing some libraries. The libraries that I have imported are:

1)Pandas: A well-liked open-source data manipulation library for Python is called Pandas. Built upon the NumPy library, it provides straightforward data structures and data analysis tools for working with structured data. Pandas offers a variety of tools and techniques for transforming and cleaning data, including ways for dealing with missing data, getting rid of duplicates, and altering data kinds. Moreover, Pandas offers strong data analysis tools, such as options for data filtering, grouping, and aggregation. Moreover, it offers tools for working with dates and times and supports time-series data processing. A variety of data sources including CSV files Excel files SQL databases and more can be read and written by Pandas.

2)NumPy: The famous open-source NumPy library for the Python programming language supports large, multi-dimensional arrays and matrices and provides a broad variety of mathematical operations for working with enormous arrays. The dependable array object in NumPy is capable of handling large, multidimensional arrays and matrices. These arrays are optimized for performance and can, successfully handle huge datasets. NumPy offers a wide range of mathematical operations that may be used with arrays including fundamental arithmetic operations, trigonometric functions, statistical functions, linear algebra functions & more. Additionally, NumPy has broadcasting which makes working with arrays in a variety of situations easier by letting mathematical operations be performed, on arrays of different sizes and types. Popular Python libraries for scientific computing include SciPy, Pandas & Matplotlib which work well with NumPy.

3)Matplotlib: A well-liked open-source charting package for Python is called Matplotlib. For building static, animated, and interactive visualizations in Python, it offers a wide variety of visualization capabilities.

Plotting Functions: There are several plotting functions offered by Matplotlib such as those for line plots, scatter plots, bar plots, histogram plots & more.

Customization: Matplotlib has numerous choices for altering the colors, styles, labels, titles, axes & other aspects of plots.

Publication-Quality Figures: Matplotlib provides a range of tools for creating high-quality visuals that may be used in scientific reports, presentations, and publications.

Integration with Other Libraries: NumPy and Pandas, two more popular Python libraries for scientific computing, operate well with Matplotlib.

4)Seaborn: A popular open-source data visualization framework for the Python programming language Seaborn was constructed on top of matplotlib. Creating captivating and illuminating statistical graphics provides a more sophisticated Python interface.

Seabourn’s salient characteristics include:

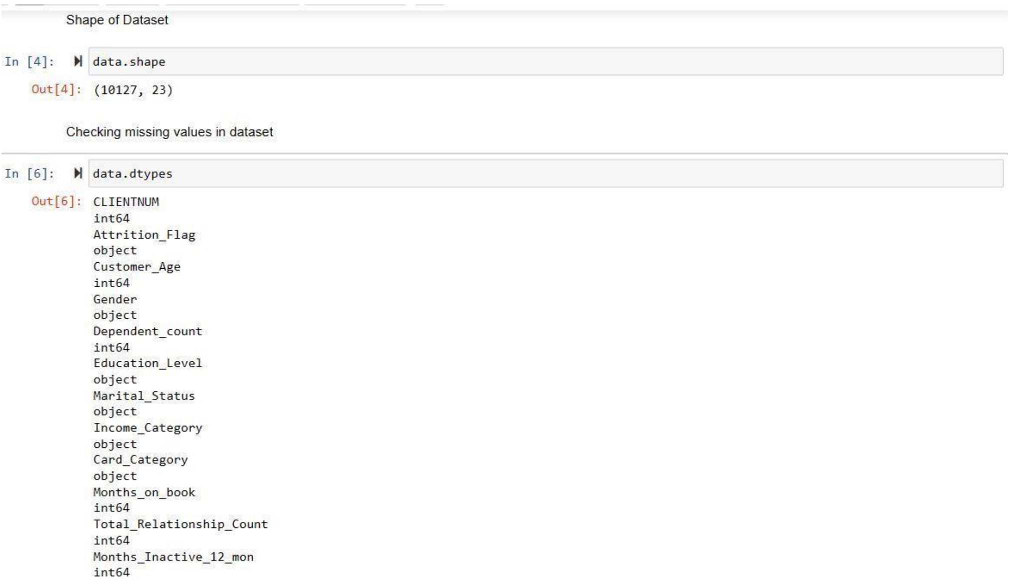
Statistical Plots: Seaborn offers a variety of statistical plots, such as scatter plots, line plots, bar plots, heatmap plots, and more, for showing correlations between variables in data.

Built-in Datasets: Seaborn has a variety of built-in datasets that may be tested out to see how different visualization techniques perform.

Color schemes: Seaborn offers a broad variety of color schemes that can be utilized to alter the plot's colors and improve its aesthetic appeal.

Simple Customization: Seaborn offers a simple interface for altering the plots' visual elements, including choices for changing the colors, styles, labels, titles, axes, and more.

Overall, Seaborn is a robust Python data visualization tool that is extensively used in applications related to data science, scientific computing, and machine learning. With just a tiny bit of code, users may create practical and beautiful visualizations thanks to its high-level interface.



Shape of Dataset

```
In [4]:  data.shape
Out[4]:  (10127, 23)
```

Checking missing values in dataset

```
In [6]:  data.dtypes
Out[6]:  CLIENTNUM
         int64
         Attrition_Flag
         object
         Customer_Age
         int64
         Gender
         object
         Dependent_count
         int64
         Education_Level
         object
         Marital_Status
         object
         Income_Category
         object
         Card_Category
         object
         Months_on_book
         int64
         Total_Relationship_Count
         int64
         Months_Inactive_12_mon
         int64
```

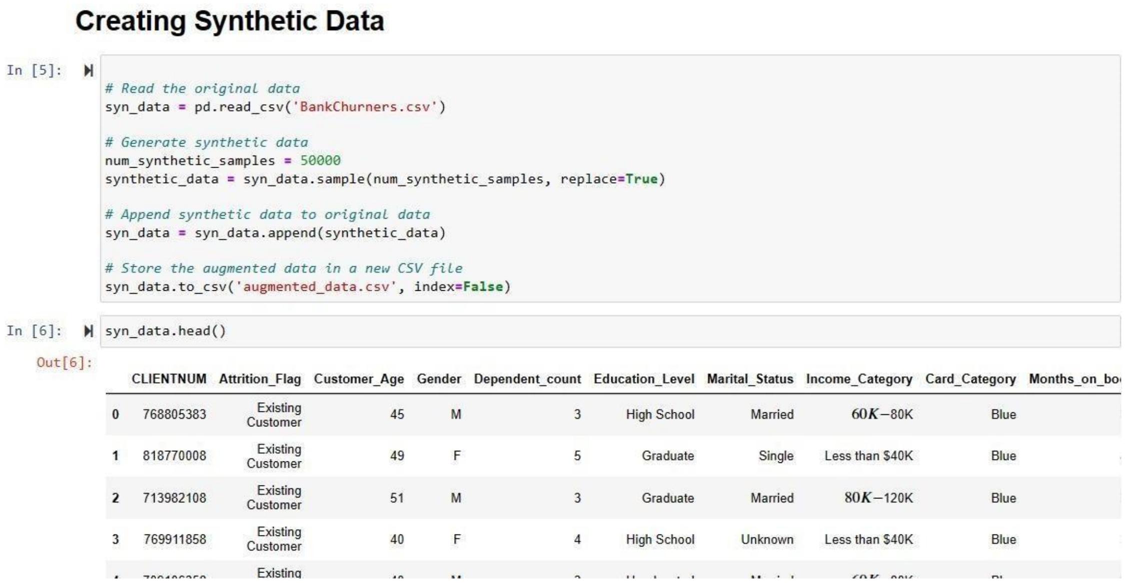
Figure 3.2.1 Checking dataset shape

In Figure- 3.2.1 above I have checked the data shape because it helps to understand the size of the dataset. For many data analysis and machine learning activities having this knowledge is essential since it enables you to confirm, that your data is the proper size and format before processing or modeling it.

Additionally, examining the form might assist in identifying pre-processing or data-loading issues such as missing or erroneous numbers. In addition, I verified the data types for each column so we could see which forms of data were included in the dataset most frequently. Pandas is what I chose for my project because it makes preparing the CSV file simpler to read.

3.3 Creating Synthetic data

Data that is manufactured artificially as opposed to being gathered from authentic sources is referred to as synthetic data. The greatest approach to improve model performance and accuracy is to train models using synthetic data. I created synthetic data using the Pandas library.



Creating Synthetic Data

```
In [5]:
# Read the original data
syn_data = pd.read_csv('BankChurners.csv')

# Generate synthetic data
num_synthetic_samples = 50000
synthetic_data = syn_data.sample(num_synthetic_samples, replace=True)

# Append synthetic data to original data
syn_data = syn_data.append(synthetic_data)

# Store the augmented data in a new CSV file
syn_data.to_csv('augmented_data.csv', index=False)
```

```
In [6]: syn_data.head()
```

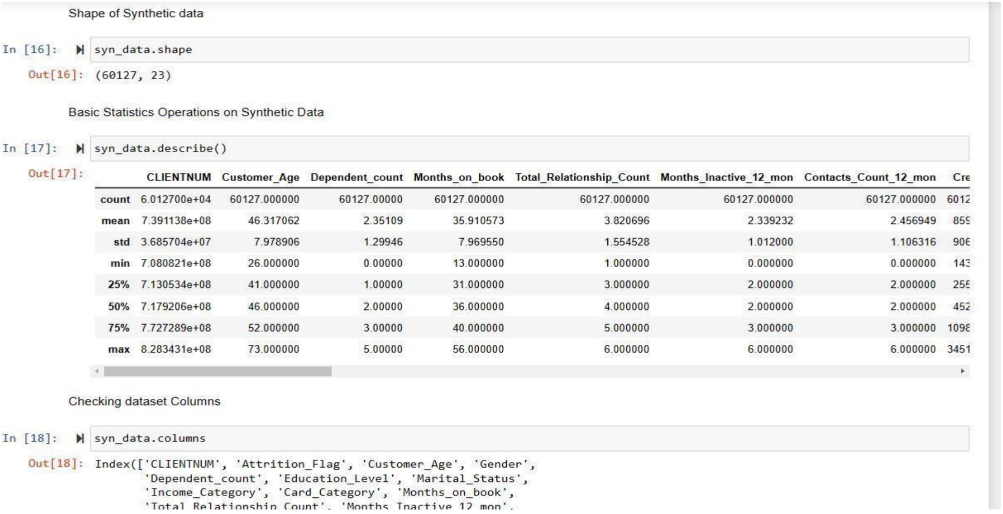
Out[6]:

	CLIENTNUM	Attrition_Flag	Customer_Age	Gender	Dependent_count	Education_Level	Marital_Status	Income_Category	Card_Category	Months_on_bo
0	768805383	Existing Customer	45	M	3	High School	Married	$60K-$80K	Blue	
1	818770008	Existing Customer	49	F	5	Graduate	Single	Less than $40K	Blue	
2	713982108	Existing Customer	51	M	3	Graduate	Married	$80K-$120K	Blue	
3	769911858	Existing Customer	40	F	4	High School	Unknown	Less than $40K	Blue	
4	[illegible]	Existing	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	

Figure 3.3(i) Synthetic data generation

In the above Figure-3.3(i) I have created Synthetic data and displayed its first five rows knowing that creating synthetic data does not affect our real data shape in terms of columns we can review our synthetic data. In my case, there is not enough real-world data available for training a predictive model effectively.

I have used Synthetic data to augment the available data and improve the model's accuracy. Churn prediction often involves imbalanced data, where the number of customers who remain loyal far outnumbers those who churn. Synthetic data can be used to balance the data and make the model more effective.



Shape of Synthetic data

```
In [16]:  syn_data.shape
Out[16]: (60127, 23)
```

Basic Statistics Operations on Synthetic Data

```
In [17]:  syn_data.describe()
Out[17]:
```

	CLIENTNUM	Customer_Age	Dependent_count	Months_on_book	Total_Relationship_Count	Months_Inactive_12_mon	Contacts_Count_12_mon	Cre
count	6.012700e+04	60127.000000	60127.00000	60127.000000	60127.000000	60127.000000	60127.000000	6012
mean	7.391138e+08	46.317062	2.35109	35.910573	3.820696	2.339232	2.456949	859
std	3.685704e+07	7.978906	1.29946	7.969550	1.554528	1.012000	1.106316	906
min	7.080821e+08	26.000000	0.00000	13.000000	1.000000	0.000000	0.000000	143
25%	7.130534e+08	41.000000	1.00000	31.000000	3.000000	2.000000	2.000000	255
50%	7.179206e+08	46.000000	2.00000	36.000000	4.000000	2.000000	2.000000	452
75%	7.727289e+08	52.000000	3.00000	40.000000	5.000000	3.000000	3.000000	1098
max	8.283431e+08	73.000000	5.00000	56.000000	6.000000	6.000000	6.000000	3451

Checking dataset Columns

```
In [18]:  syn_data.columns
Out[18]: Index(['CLIENTNUM', 'Attrition_Flag', 'Customer_Age', 'Gender',
                'Dependent_count', 'Education_Level', 'Marital_Status',
                'Income_Category', 'Card_Category', 'Months_on_book',
                'Total_Relationship_Count', 'Months_Inactive_12_mon',
```

Figure 3.3 (ii) Checking shape of synthetic data

In the figure-3.3(ii) I checked out the shape of Synthetic Data and it is as same as the real data but you can see that there is a change in the number of rows. The number of rows in Synthetic data increases and reaches up to 60K and their column shape is as same as to original data. I have added 50000 samples to create synthetic data.

I also applied some statistical methods to synthetic data to find the minimum, maximum, and standard deviation of data and show all columns of data that there is no column missing after creating synthetic data.

3.4 Data exploration and data preprocessing

Data preprocessing is the act of converting raw data into a format suitable for further examination. It is a critical step in the data analysis process. Data preparation is done to clean up noise, enhance data quality, and get the data ready for analysis. Typically, data preparation requires a number of processes, including:

1)Cleaning up the data by identifying and correcting any mistakes, missing data & inconsistencies.

2)The process of merging data from several datasets is known as data integration.

3)Data transformation is the process of converting data into a format suitable for analysis by encoding, normalizing, or scaling.

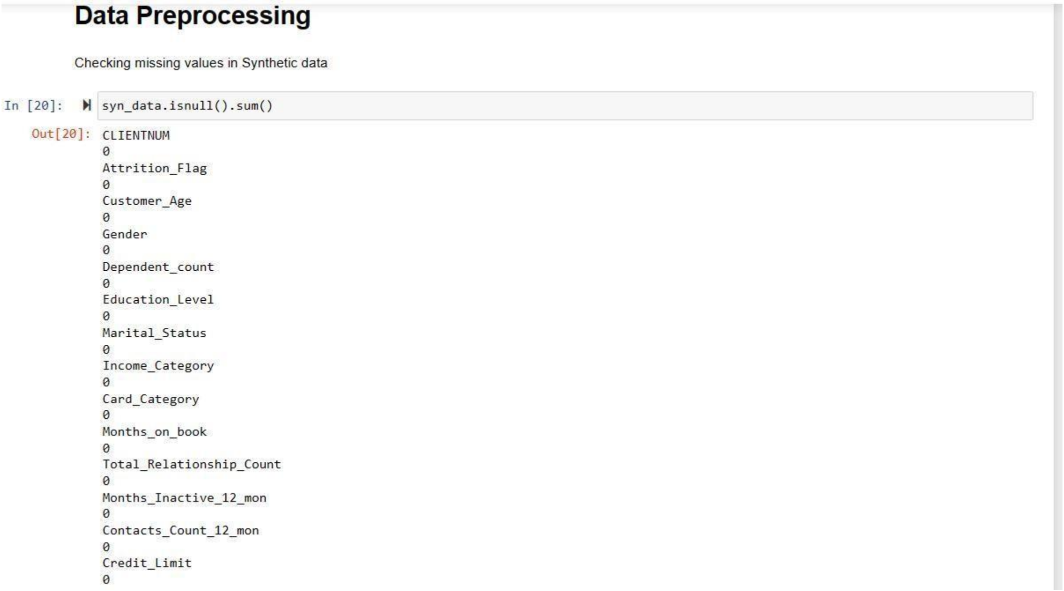
4)The practice of selecting a smaller number of applicable qualities or samples from the overall quantity of data to be assessed is known as data reduction.

5)Data discretization is the process of transforming continuous data into discrete categories.

3.4.1 Checking for missing values

Any analysis or modeling done on the data may be impacted by missing values that are introduced into the data. Missing values signify missing data which may produce incorrect findings.

Any data quality concerns, such as data input mistakes or inconsistent recording techniques can be found by looking for missing numbers.



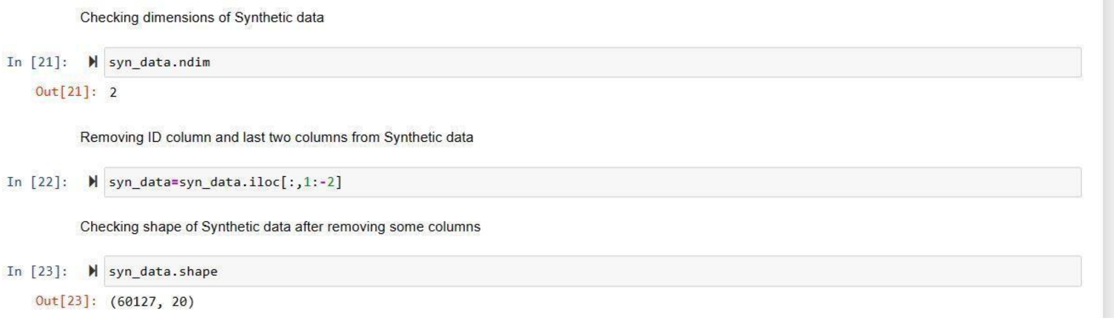
Data Preprocessing

Checking missing values in Synthetic data

```
In [20]:  syn_data.isnull().sum()
Out[20]:  CLIENTNUM
          0
          Attrition_Flag
          0
          Customer_Age
          0
          Gender
          0
          Dependent_count
          0
          Education_Level
          0
          Marital_Status
          0
          Income_Category
          0
          Card_Category
          0
          Months_on_book
          0
          Total_Relationship_Count
          0
          Months_Inactive_12_mon
          0
          Contacts_Count_12_mon
          0
          Credit_Limit
          0
```

Figure 3.4.1(i) Checking null values

In the figure-3.4.1(i) above there are no missing values in the dataset.



Checking dimensions of Synthetic data

```
In [21]:  syn_data.ndim
Out[21]:  2
```

Removing ID column and last two columns from Synthetic data

```
In [22]:  syn_data=syn_data.iloc[:,1:-2]
```

Checking shape of Synthetic data after removing some columns

```
In [23]:  syn_data.shape
Out[23]:  (60127, 20)
```

Figure 3.4.1 (ii) Checking the dimension of dataset created

In figure-3.4.1(ii) I checked the data dimensionality to know about what dimensions of the data are. This helps us to train our model accurately. We remove 3 columns from the dataset because it is not required for the analysis. I removed the ID column as it does not put an impact on the prediction.

After removing these columns we need to check the shape of the data that we had removed. We check for any extra columns if they are generated by mistake.

3.4.2 Looking for outliers

Outliers are statistical observations that deviate significantly from other statistical observations in a dataset. These data points go outside of the typical range of values for a certain variable. Outliers may appear for a number of reasons such as measurement or data entry mistakes or they may truly represent extraordinary results, that call for more study.

In a dataset, we look for outliers for a variety of reasons. In the first place, outliers may drastically impact statistical parameters like the mean and standard deviation, if they are not properly taken into account which might lead to unreliable conclusions. Second, outliers might disclose details

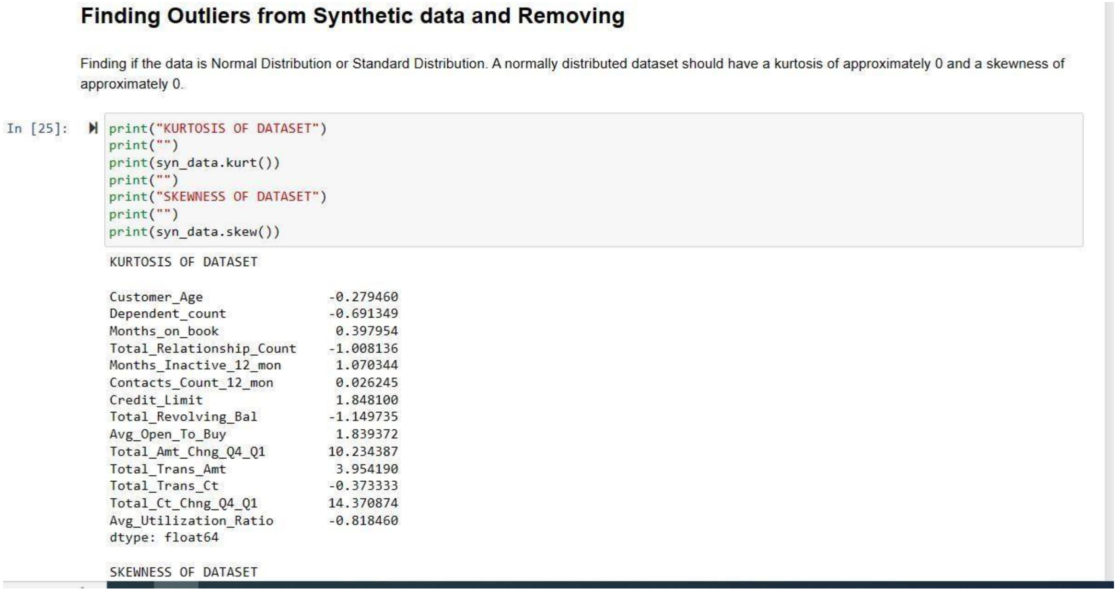
about certain events or patterns that would otherwise go unnoticed by focusing just on the center trend of the data.

We start by looking for the skewness and the kurtosis in a dataset.

SKEWNESS [16]**-** Indicating the asymmetry of a distribution statistically is skewness. A normal distribution without skewness is used to assess deviation from that distribution.
If the distribution is positively skewed to the right the tail is longer on the right side and the distribution is considered to be skewed. Conversely, if the distribution is skewed to the left it has a negative skewness score indicating that the distribution's left side has a longer tail.

KURTOSIS – It is a statistical term that indicates how peaky or flat a distribution is in relation to the normal distribution. It gauges, how much a distribution's tails deviate from those of a normal distribution.
Extreme values are more frequent than in a normal distribution in a distribution with a high kurtosis which features a sharp peak and heavy tails. Contrarily, a distribution with low kurtosis has a flatter peak and lighter tails than the normal distribution indicating that the data is more tightly grouped around the mean and has fewer outliers.

Determining if the data is distributed normally or uniformly.
A dataset with a properly distributed distribution should have a kurtosis and skewness of around 0. Skewness gauges how asymmetrically a variable's distribution is distributed. Kurtosis gauges how peaked or flat a variable's distribution is in comparison to a normal distribution. Using the data's mean and the standard deviation is a frequent technique for spotting outliers. Particularly, an outlier is defined as an observation that deviates from the mean by more than three standard deviations.



Finding Outliers from Synthetic data and Removing

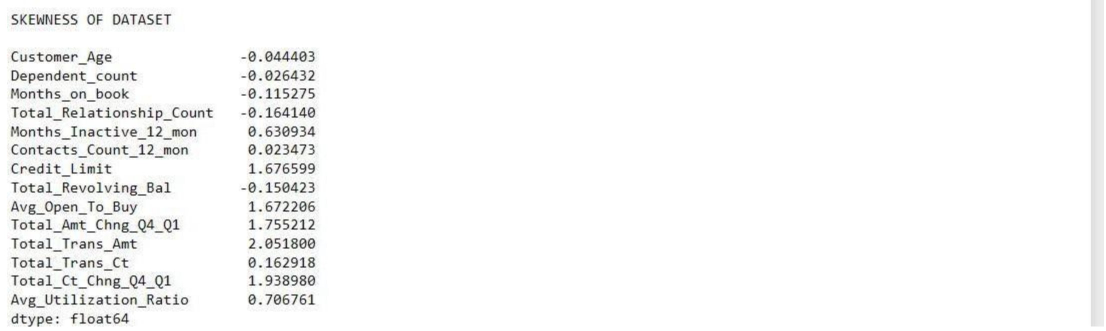
Finding if the data is Normal Distribution or Standard Distribution. A normally distributed dataset should have a kurtosis of approximately 0 and a skewness of approximately 0.

```
In [25]:  print("KURTOSIS OF DATASET")
          print("")
          print(syn_data.kurt())
          print("")
          print("SKEWNESS OF DATASET")
          print("")
          print(syn_data.skew())
```

```
KURTOSIS OF DATASET

Customer_Age                -0.279460
Dependent_count             -0.691349
Months_on_book               0.397954
Total_Relationship_Count    -1.008136
Months_Inactive_12_mon       1.070344
Contacts_Count_12_mon        0.026245
Credit_Limit                 1.848100
Total_Revolving_Bal         -1.149735
Avg_Open_To_Buy              1.839372
Total_Amt_Chng_Q4_Q1        10.234387
Total_Trans_Amt              3.954190
Total_Trans_Ct              -0.373333
Total_Ct_Chng_Q4_Q1         14.370874
Avg_Utilization_Ratio       -0.818460
dtype: float64

SKEWNESS OF DATASET
```

Figure 3.4.2(i) Calculating kurtosis and skewness of dataset

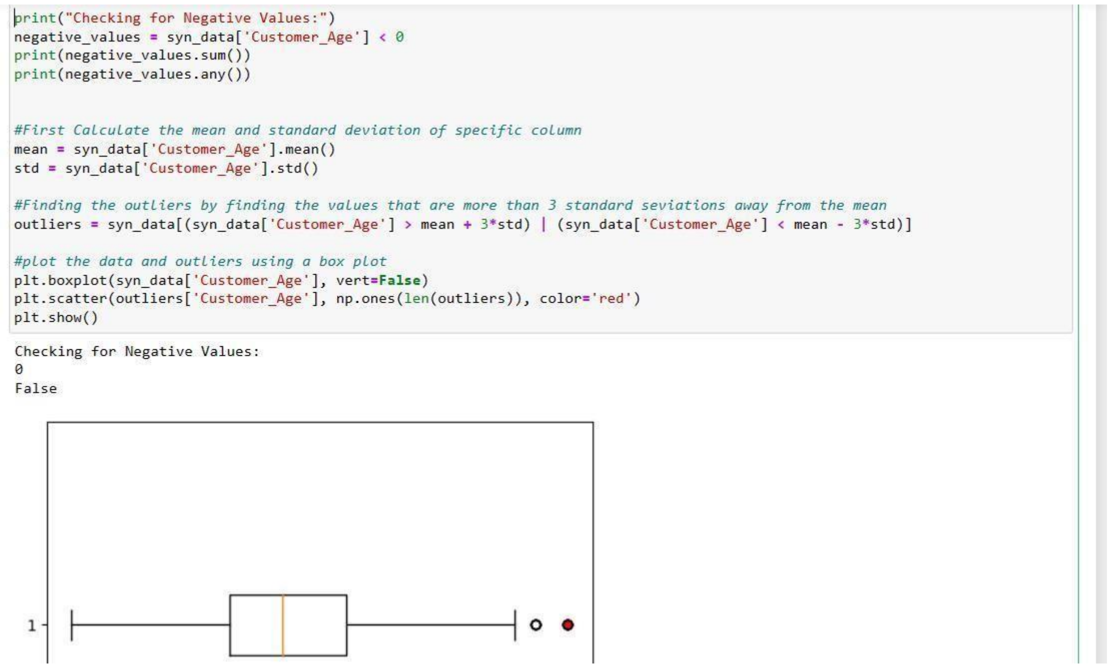
In the figure-3.4.2(i) above you can see that the kurtosis of dataset is zero.



```
SKEWNESS OF DATASET

Customer_Age               -0.044403
Dependent_count            -0.026432
Months_on_book             -0.115275
Total_Relationship_Count   -0.164140
Months_Inactive_12_mon      0.630934
Contacts_Count_12_mon       0.023473
Credit_Limit                1.676599
Total_Revolving_Bal        -0.150423
Avg_Open_To_Buy             1.672206
Total_Amt_Chng_Q4_Q1        1.755212
Total_Trans_Amt             2.051800
Total_Trans_Ct              0.162918
Total_Ct_Chng_Q4_Q1         1.938980
Avg_Utilization_Ratio       0.706761
dtype: float64
```

In figure-3.4.2(ii) above you can see that the skewness of the dataset is zero. So, the data is Standard distribution. Both Kurtosis and Skewness are not zero so we can find the outliers by standard deviation.

Checking for Outliers for different columns



```
print("Checking for Negative Values:")
negative_values = syn_data['Customer_Age'] < 0
print(negative_values.sum())
print(negative_values.any())


#First Calculate the mean and standard deviation of specific column
mean = syn_data['Customer_Age'].mean()
std = syn_data['Customer_Age'].std()

#Finding the outliers by finding the values that are more than 3 standard seviations away from the mean
outliers = syn_data[(syn_data['Customer_Age'] > mean + 3*std) | (syn_data['Customer_Age'] < mean - 3*std)]

#plot the data and outliers using a box plot
plt.boxplot(syn_data['Customer_Age'], vert=False)
plt.scatter(outliers['Customer_Age'], np.ones(len(outliers)), color='red')
plt.show()
```

```
Checking for Negative Values:
0
False
```

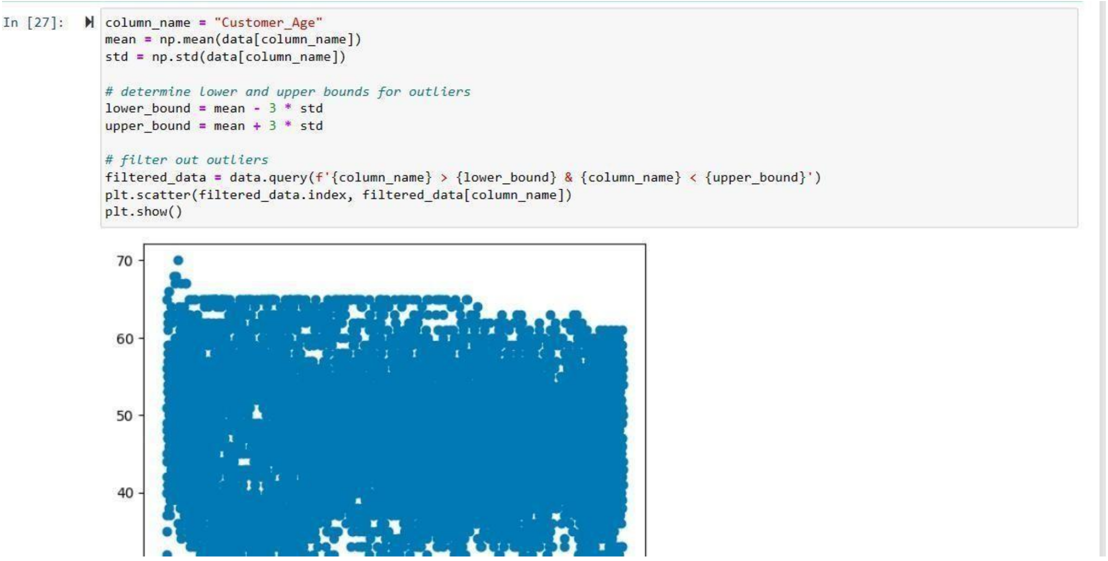
Figure 3.4.2 (iii)Finding Outliers

Starting with the Customer Age column we check the negative values there before, moving on to check the negative values of other columns. This is done because negative values can point to data quality problems such as incorrect data entry or measurement errors which could have an impact, on the analysis's or modeling's conclusions.

Negative numbers might not make sense in the context of the data and could require special handling. For instance, it would be meaningless to use negative numbers for a characteristic like "age" or "income".
Outliers are often defined in terms of positive values hence the definition and understanding, of outliers may be impacted by negative values.

Customer_Age:
In the figure above I have checked for negative values in the column before checking the outliers in the column and then finding the outliers in the column (Customer_Age)

There are no Negative values in the Customer_Age column but there are some outliers.



```
In [27]:  column_name = "Customer_Age"
          mean = np.mean(data[column_name])
          std = np.std(data[column_name])

          # determine lower and upper bounds for outliers
          lower_bound = mean - 3 * std
          upper_bound = mean + 3 * std

          # filter out outliers
          filtered_data = data.query(f'{column_name} > {lower_bound} & {column_name} < {upper_bound}')
          plt.scatter(filtered_data.index, filtered_data[column_name])
          plt.show()
```

Figure 3.4.2 (iv) Removing Outliers

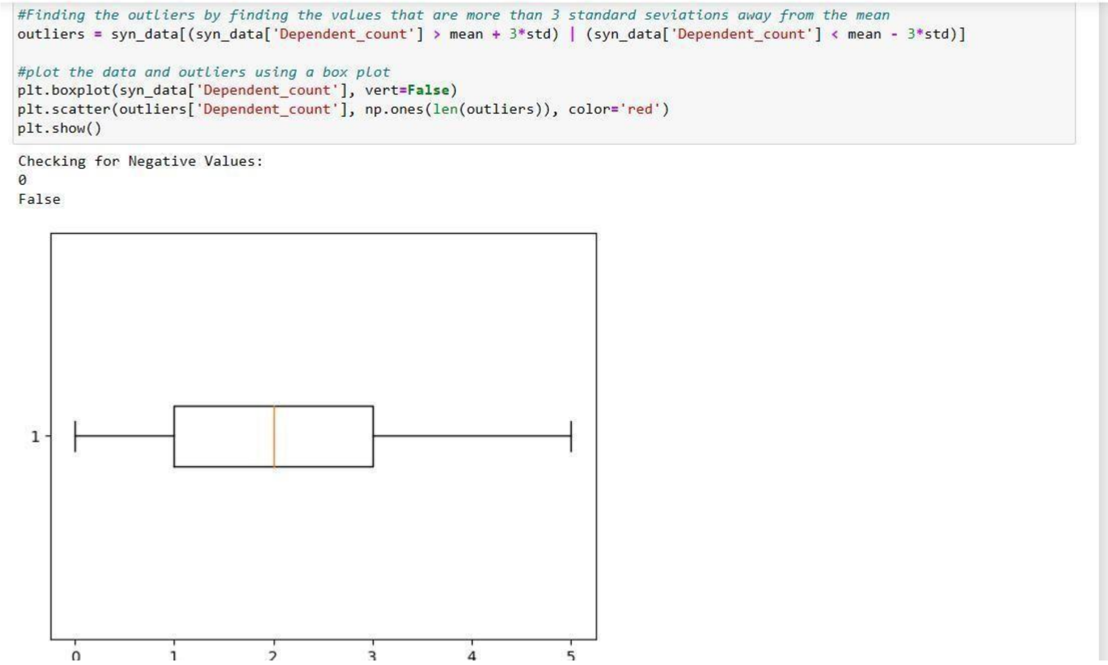
There are some outliers in the Customer_Age column in figure-3.4.2(iv) as seen that some values are above 70 and we have to remove them to make the column outlier free, So I removed the outliers by setting its lower bound and upper bound that is calculated through mean and standard deviation.

After finding the lower and upper bound values we need to find filtered data that have no outlier by fitting data in such a manner that it is greater than the lower bound and less than the upper bound. It can be seen that there is no outlier remaining in the column that the data is in proper shape and free from outliers.

Dependent_Count:

The dependent_count column in a dataset typically provides information on the number of dependents that each individual or household has. According to the context of the data outliers in this column might be numbers that are much greater or lower than what is normal or anticipated.

For a number of reasons, it's critical to spot outliers in the dependent_count column. Finding outliers is a crucial step in assuring data quality since they might be a sign of data input or processing issues. Statistical models constructed utilizing the dependent_count column may be significantly impacted by outliers.



```
#Finding the outliers by finding the values that are more than 3 standard seviations away from the mean
outliers = syn_data[(syn_data['Dependent_count'] > mean + 3*std) | (syn_data['Dependent_count'] < mean - 3*std)]

#plot the data and outliers using a box plot
plt.boxplot(syn_data['Dependent_count'], vert=False)
plt.scatter(outliers['Dependent_count'], np.ones(len(outliers)), color='red')
plt.show()
```

```
Checking for Negative Values:
0
False
```

Figure 3.4.2 (v) Finding Outliers

In (Fig-3.4.2(v) there is a analysis of Dependent_Count and there are no outliers in this column so we don’t need to remove any outliers.

We do the same for all the columns in the dataset.

Chapter 04 Implementation and Results

Exploratory data analysis and a few machine-learning models will be covered in this chapter.

IMPLEMENTATION PHASE 2

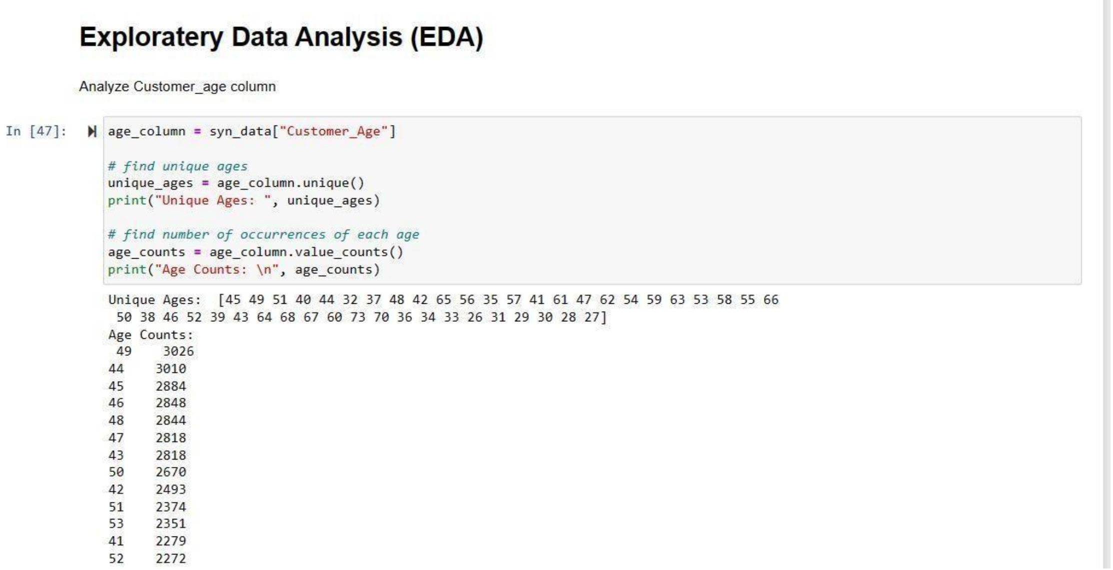
4.1 Exploratory Data Analysis

Exploratory data analysis is the act of breaking down and synthesizing data sets, in order to learn more and create ideas, about the underlying connections and patterns, found in the data. (biswal, 2023) The data has to be visually examined and summarized in order to discover patterns, trends, and outliers. EDA frequently involves examining variable distributions determining the relationships between variables and identifying, any problems or anomalies in the data.

EDA's main goal is to understand the data and discover, more about the relationships between variables rather than just verifying or refuting presumptions. Before more formal statistical modeling is implemented it is typically used, in the early phases of data analysis. EDA may help identify possible issues, with the data such as missing values or outliers and it can help choose, the appropriate statistical techniques for additional research.

4.1.1 Analyzing Columns:

Analysing the customer age column



Exploratery Data Analysis (EDA)

Analyze Customer_age column

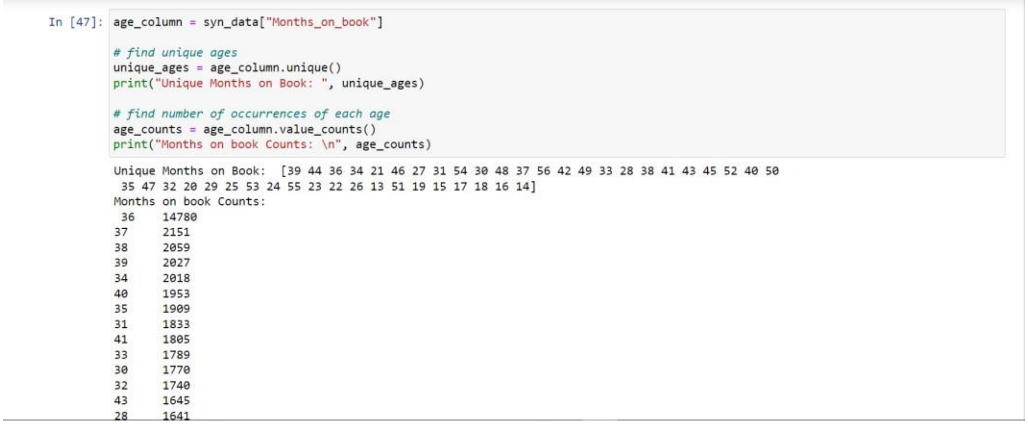
In [47]:

```
age_column = syn_data["Customer_Age"]

# find unique ages
unique_ages = age_column.unique()
print("Unique Ages: ", unique_ages)

# find number of occurrences of each age
age_counts = age_column.value_counts()
print("Age Counts: \n", age_counts)
```

```
Unique Ages:  [45 49 51 40 44 32 37 48 42 65 56 35 57 41 61 47 62 54 59 63 53 58 55 66
 50 38 46 52 39 43 64 68 67 60 73 70 36 34 33 26 31 29 30 28 27]
Age Counts: 
 49    3026
44    3010
45    2884
46    2848
48    2844
47    2818
43    2818
50    2670
42    2493
51    2374
53    2351
41    2279
52    2272
```

Figure 4.1.1 (i) Analyzing the customer age column

In figure-4.1.1(i) I analyzed the Customer Age and found two things:
Firstly, the unique age numbers in the column. The analysis shows that the unique age numbers start from 27 and go up to 73.
Secondly the count of each age number that how much time an age repeated is there in the data. For example, there are 3026 customers that have 49 ages.



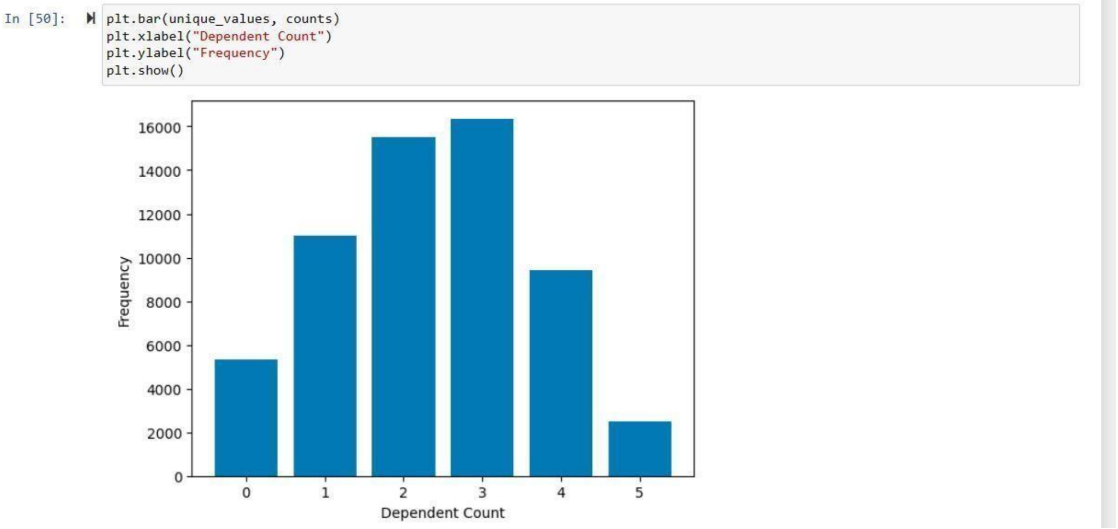
```
In [47]: age_column = syn_data["Months_on_book"]

         # find unique ages
         unique_ages = age_column.unique()
         print("Unique Months on Book: ", unique_ages)

         # find number of occurrences of each age
         age_counts = age_column.value_counts()
         print("Months on book Counts: \n", age_counts)

         Unique Months on Book:  [39 44 36 34 21 46 27 31 54 30 48 37 56 42 49 33 28 38 41 43 45 52 40 50
          35 47 32 20 29 25 53 24 55 23 22 26 13 51 19 15 17 18 16 14]
         Months on book Counts:
          36    14780
         37     2151
         38     2059
         39     2027
         34     2018
         40     1953
         35     1909
         31     1833
         41     1805
         33     1789
         30     1770
         32     1740
         43     1645
         28     1641
```

Figure 4.1.1 (ii) Analyzing the months on books column

In figure-4.1.1(ii) above I analyzed the Months on books and found two things:
Firstly, the unique months of books in the column. The analysis shows that the unique months on books numbers start from 36 and go up to 55.
Secondly, the count of each age number that how much time a month on book repeated is there in the data. For example, there are 2151 customers that have 37 months on books.

Analysing the dependent count column

Dependent Count means that how many people are dependents on customer like their parents, children, wives etc.



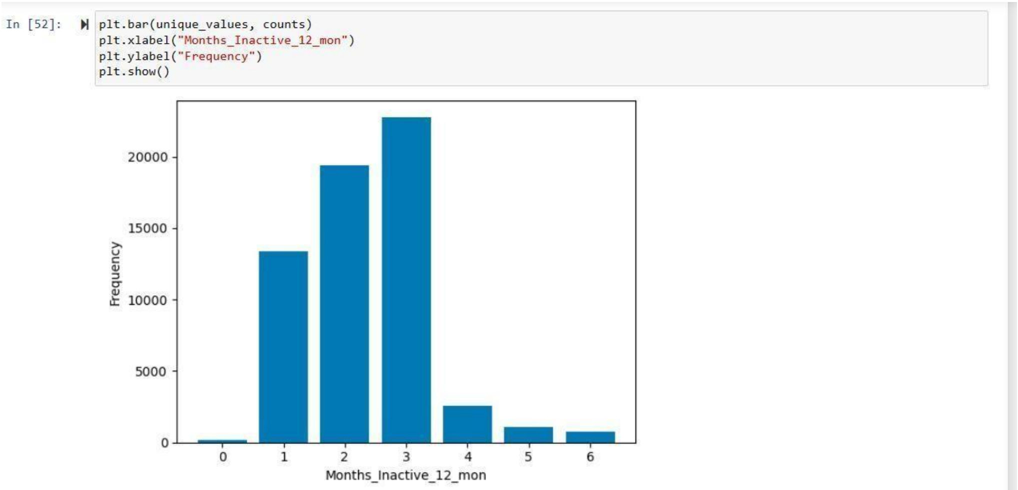
```
In [50]: plt.bar(unique_values, counts)
         plt.xlabel("Dependent Count")
         plt.ylabel("Frequency")
         plt.show()
```

Figure 4.1.1 (iii) Analyzing the dependent count column

In figure-4.1.1(iii) I have analyzed the Dependent_Count column which means how many dependents a customer has. Most of the customers have only one dependent and then second most of the customers have two dependents. Further, I plotted their numbers by using a bar plot.

Analyzing dependent_Count columns can help identify, which features or variables are most strongly associated with churn. This information can be used to develop more accurate churn prediction models and to identify strategies for reducing customer churn. Additionally, analyzing dependent count columns can help identify patterns in customer behavior that may be indicative of churn such as changes in usage frequency or decreases in the number of interactions, with customer service.

Analyzing the contact count in last 12 months



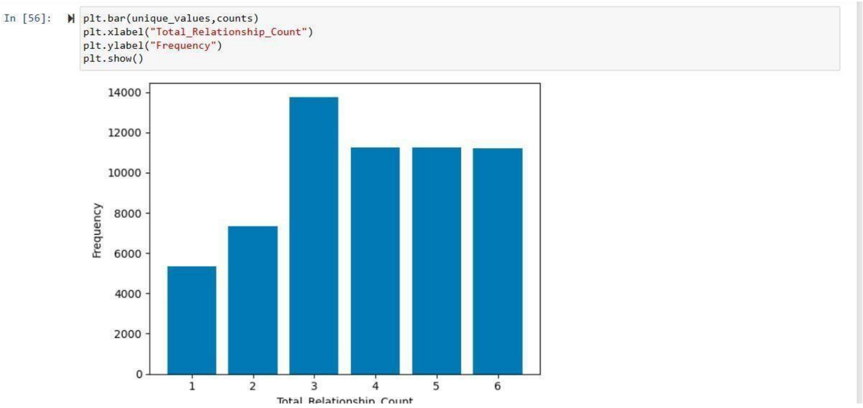
```
In [52]:  plt.bar(unique_values, counts)
          plt.xlabel("Months_Inactive_12_mon")
          plt.ylabel("Frequency")
          plt.show()
```

Figure 4.1.1 (iv) Analyzing the contact count in last 12 months column

In figure-4.1.1(iv) I analyzed the Months_Inactive_12_mon column which means how much time a customer is not attached to the organization or has not made a transaction.
We can find clients that are likely to churn by looking at the Month_Inactive_12_mon column. Customers may be more prone to churn than those who have been regularly engaged if they have been inactive for a sizable chunk of the prior 12 months.
In addition, looking at the Month_Inactive_12_mon column might reveal customer behavior patterns that can be linked to churn.

Analyzing the Total Relationship Column

The entire number of goods or services, a customer has with a certain business is indicated in the "Total_Relationship_Count" column in a customer churn prediction report. This column offers details on a customer's degree of involvement and loyalty to a business.



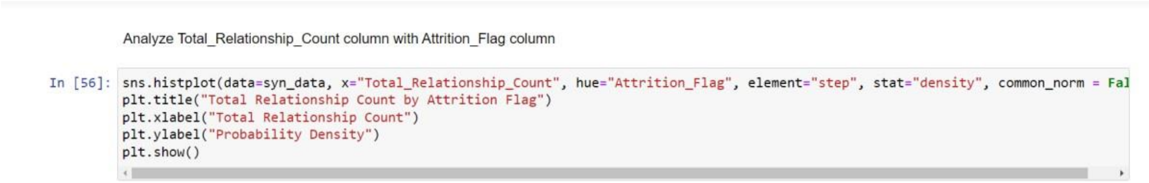
```
In [56]:  plt.bar(unique_values,counts)
          plt.xlabel("Total_Relationship_Count")
          plt.ylabel("Frequency")
          plt.show()
```

Figure 4.1.1 (v) Analyzing the Total relationship count column

In the figure-4.1.1(v) I have analyzed the Total_relationship_Count column which is about to represent the total number of financial relationships or accounts that a customer has with the bank or organization. The number 1 to 5 simultaneously displays the customer's relationships.

Analyzing the Total_Relationship_Count column can help identity which customers are more or less likely to churn. For instance, clients who have a high total connection count may be more devoted and less likely to leave than those who have a low total relationship count.

Additionally, analyzing the Total_Relationship_Count column can help identify opportunities to increase customer loyalty and engagement.



Analyze Total_Relationship_Count column with Attrition_Flag column

```
In [56]: sns.histplot(data=syn_data, x="Total_Relationship_Count", hue="Attrition_Flag", element="step", stat="density", common_norm = Fal
         plt.title("Total Relationship Count by Attrition Flag")
         plt.xlabel("Total Relationship Count")
         plt.ylabel("Probability Density")
         plt.show()
```

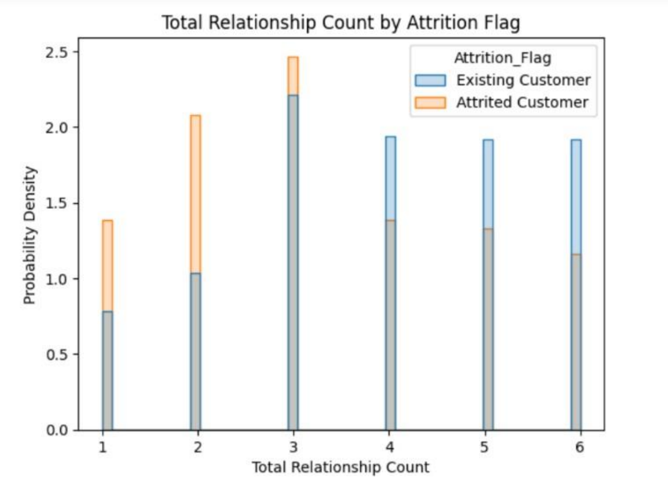
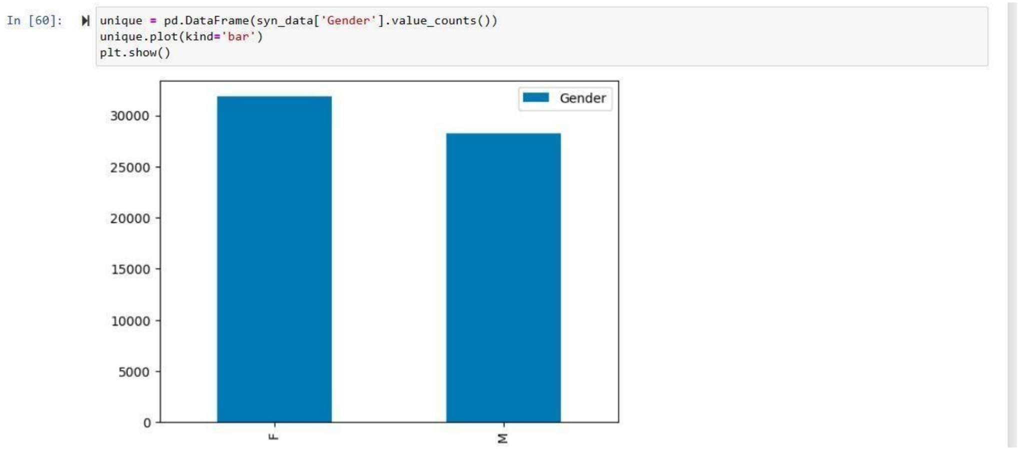


Figure 4.1.1 (vi) Analyzing the Relationship count column probability

We drew another graph from the figure-4.1.1(v) graph above that illustrates the chance of consumers churning with regard to the relationship count. If a customer has a big number of relationships, it implies that the consumer is an existing customer. For example, on the graph most people have three goods indicating that they are more likely to stay with the firm.

Analyzing the Gender Column



In [60]:

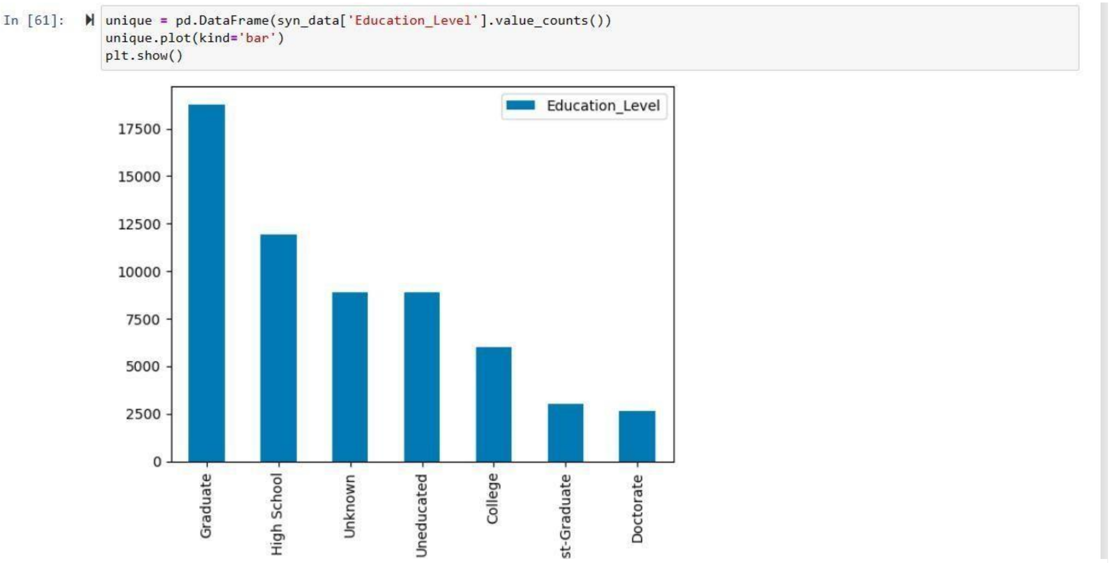
```
unique = pd.DataFrame(syn_data['Gender'].value_counts())
unique.plot(kind='bar')
plt.show()
```

Figure 4.1.1 (vii) Analyzing the gender column

In the figure-4.1.1(vii) above I have done the analysis of the Gender column. It shows how many Males and Females are in the dataset. With the analysis there are a lot of Females than Males. Females numbers are up to 35000 and males numbers are up to 25000. By using Barplot this insight is clearly shown above.

The proportion of gender count is almost equally distributed (52.9% male and 47.1%) compare to proportion of existing and attributed customer count (83.9% and 16.1%) which is highly imbalanced. The proportion of Attrited customers by gender there are 14.4% more male than female who have churned.

Analysing the Education Level Column:



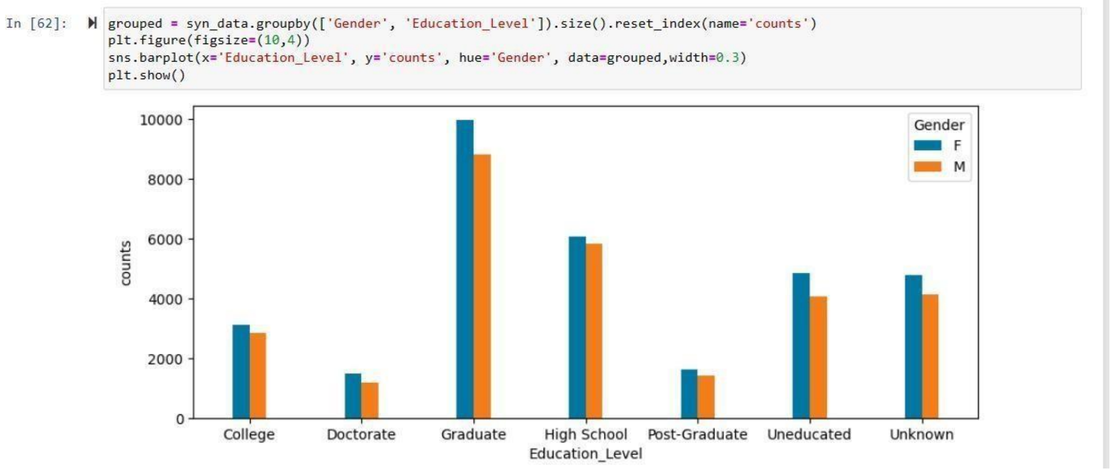
In [61]:

```
unique = pd.DataFrame(syn_data['Education_Level'].value_counts())
unique.plot(kind='bar')
plt.show()
```

Figure 4.1.1 (viii) Analyzing the Education column

In the Figure-4.1.1(viii) analysis of the Education_level column shows that most of the customers are Graduates and then the High School Customers. The lowest numbers of customers are from the Doctorate level.

From the two graphs above one final graph is plotted for the education level and gender.

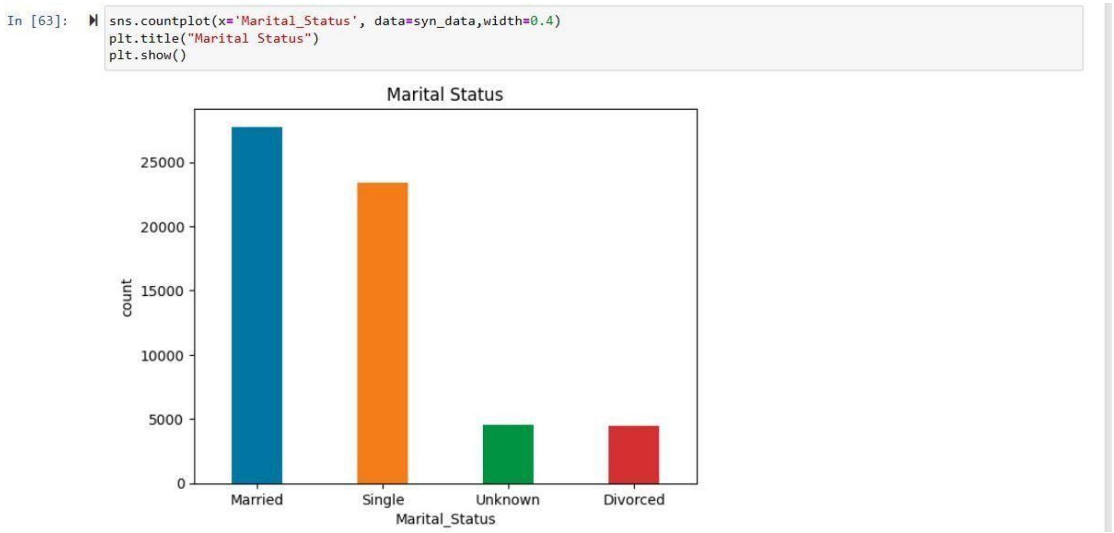


```
In [62]:  grouped = syn_data.groupby(['Gender', 'Education_Level']).size().reset_index(name='counts')
           plt.figure(figsize=(10,4))
           sns.barplot(x='Education_Level', y='counts', hue='Gender', data=grouped,width=0.3)
           plt.show()
```

Figure 4.1.1 (ix) Analyzing the education and gender columns

In the figure-4.1.1(ix) I have done the analysis of Education_Level with gender and compared the Education of both genders. It can be clearly seen that there are more numbers of Males and Females of graduate customers and in every education level there are more numbers of Females than Males.

Analyzing the Marital status column



```
In [63]:  sns.countplot(x='Marital_Status', data=syn_data,width=0.4)
           plt.title("Marital Status")
           plt.show()
```

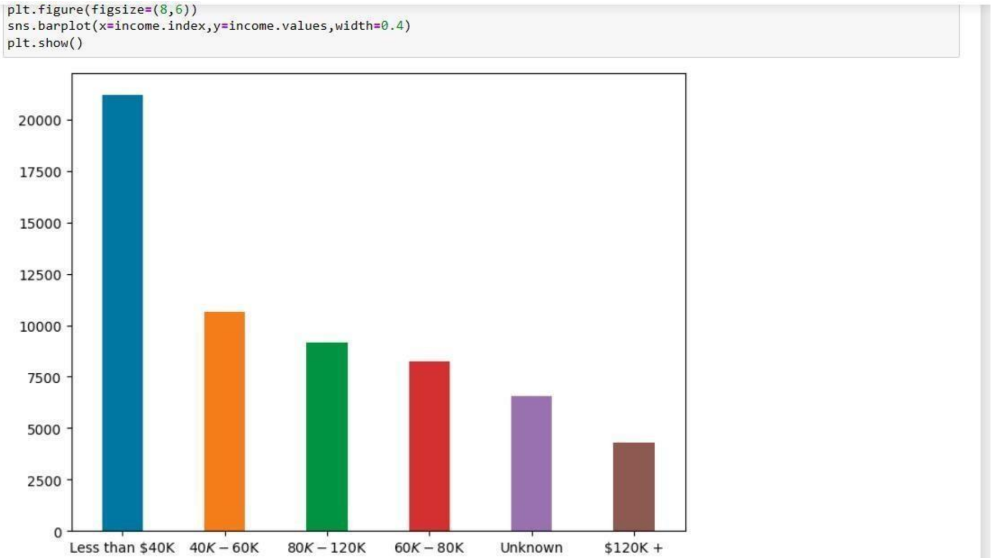
Figure 4.1.1 (x) Analyzing the Marital status column

According to the dataset the Martial_Status column in the data above, there are about 25,000 married customers followed by around 20,000 single customers. Customers in the third group have an uncertain marital status which means their marriage status is unknown. Finally, the study indicates, that some consumers are divorced.

A high proportion of marital status of customers who have churned is Married (43.6%), followed by Single (41.1%) compared to Divorced (7.4%) and Unknown (7.9%) status - Marital status of the attributed customers are highly clustered in Married status and Single.

Analyzing the income category Column

The income bracket of consumers is indicated in the "Income_Category" column of the customer churn forecast. This column contains data about a customer's financial standing and purchasing power.



```
plt.figure(figsize=(8,6))
sns.barplot(x=income.index,y=income.values,width=0.4)
plt.show()
```

Figure 4.1.1 (xi) Analyzing the Income category column

I used a bar plot to illustrate how many clients had how much money in the figure-4.1.1(xi) after analyzing the Income_Category column. This graph shows that more than 20000 of the company's clients earn less than $40k annually.

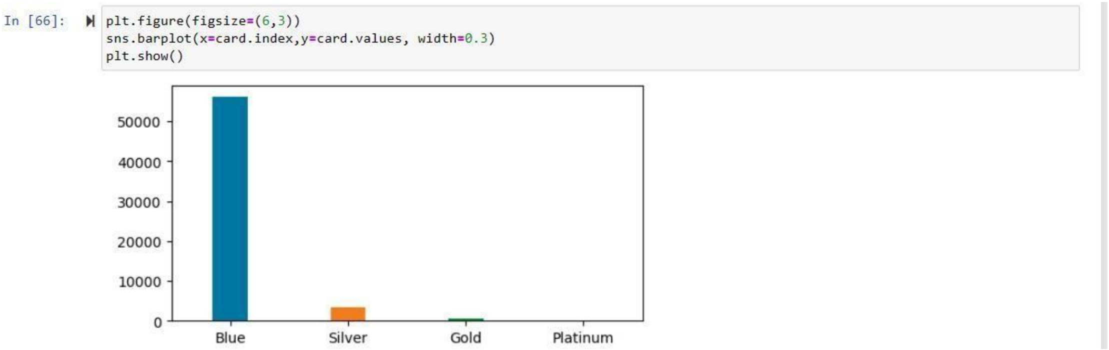
Which clients are more or less likely to churn may be determined by looking at the Income_Category field. Customers who earn more money for example can be more important to a business and less, inclined to leave than those who earn less money.

Additionally, looking at the Income_Category column might reveal ways to raise sales and improve customer satisfaction.

As you can see from the proportion of income category of attrited customer, it is highly concentrated around 60K−80K income (37.6%), followed by Less than 40K income(16.7120K + (11.5%). I assume that customers with higher income doesn't likely to leave their credit card services than meddle-income customer.

Analyzing the Credit Limit Column

The level or kind of credit card a client has, such as gold, platinum, or black, is often indicated by the Card_Category field in customer churn forecast.



```
In [66]:  plt.figure(figsize=(6,3))
          sns.barplot(x=card.index,y=card.values, width=0.3)
          plt.show()
```

Figure 4.1.1 (xii) Analyzing the card category column

The study of the Card_Category column in the preceding figure reveals that there are four different card category types: Blue, Silver, Gold, and Platinum. Nearly 90% of clients have a blue card.

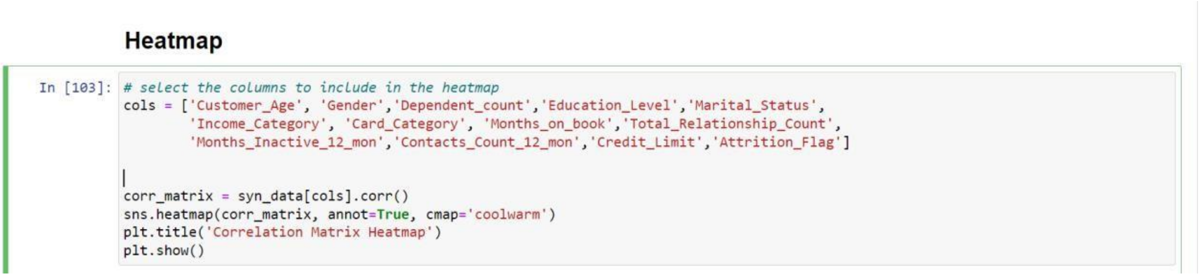
Finding similarities or variances in consumer behavior and preferences based on their card type may be accomplished by analyzing this column. Customers with higher-level cards, for instance, can be more lucrative or face various churn risks than customers with lower-level cards. Having an understanding of these variations may help retention initiatives be more focused, increase overall customer happiness, and boost profitability.

4.2 HEATMAPS

Heatmaps are graphical representations of data, where the values of a matrix are represented by colors. Each matrix element has a value that correlates to a color, typically a gradient of one color or a selection of colors. For analyzing massive datasets, spotting trends, and making decisions, heatmaps are helpful. [17]

Heatmaps are used to explore and comprehend patterns in sizable datasets in data analysis projects. A wide variety of data types, including gene expression, survey results, web traffic, and more, can be represented using heatmaps. Finding trends, clusters, and outliers is made simpler by plotting data in a heatmap. When working with sizable datasets that contain a lot of variables or dimensions, heatmaps are especially helpful.

Heatmaps are helpful in a number of fields including biology, finance, marketing & sports. For instance, heatmaps are often used to show gene expression data in biology in order to identify gene sets with comparable expression patterns. Financial analysts can identify patterns and correlations between numerous firms by using heatmaps to depict stock movements. Heatmaps may be used to illustrate customer behavior in marketing making it simpler, to identify popular items or areas of a website that receive a lot of traffic. Heatmaps may be used in sports to visualize player performance and show coaches where their players are strong and weak. [18]



Heatmap

```
In [103]: # select the columns to include in the heatmap
          cols = ['Customer_Age', 'Gender','Dependent_count','Education_Level','Marital_Status',
                  'Income_Category', 'Card_Category', 'Months_on_book','Total_Relationship_Count',
                  'Months_Inactive_12_mon','Contacts_Count_12_mon','Credit_Limit','Attrition_Flag']

          |
          corr_matrix = syn_data[cols].corr()
          sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
          plt.title('Correlation Matrix Heatmap')
          plt.show()
```

Figure 4.2 Heatmap created

4.3 Correlation Matrix

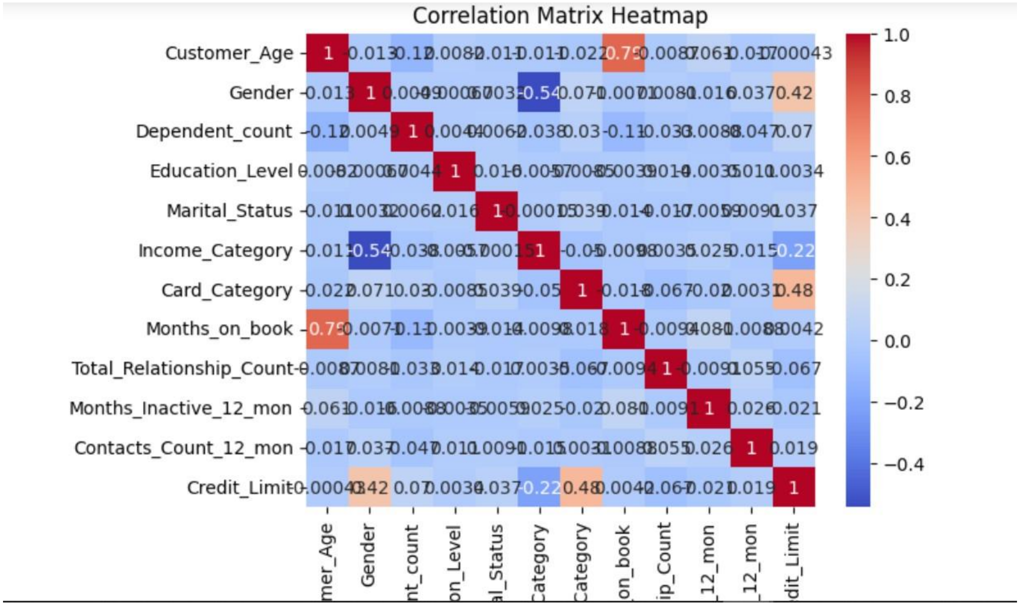


Figure 4.3 Correlation matrix

A heatmap of the correlation matrix of a subset of columns from a dataset can be seen in the graph above. The correlation matrix, where each cell stands for the correlation coefficient between two variables displays the pairwise correlations between the chosen columns. [19]

The intensity of the correlation coefficients is depicted on the heatmap using a color scale. From blue (negative correlation) through white (no correlation) and finally red (positive correlation). The correlation coefficient is stronger the darker the color. The heatmap also features annotations for the correlation coefficients found, within each cell.

A few examples of the several variables in this heatmap that are substantially associated with one another are "Total Relationship Count" and "Months on Book," both of which have a strong positive correlation coefficient (dark red cell). For instance, there is no association between the "Customer Age" and "Contacts Count 12 mon" variables. On the other side, we can also see certain factors that have no correlation or only a weak correlation, with one another (white cell).

Overall, this heatmap offers a quick and simple way to spot the patterns and connections, between the chosen variables which can aid in data analysis and help with decision-making.

4.3.1 Correlation Matrix in our project

The effectiveness of a classification algorithm is assessed using a confusion matrix. False positives and false negatives are produced by the algorithm along with real positives and true negatives. Accuracy, precision, recall, and F1 score are computed by the confusion matrix.

For binary classification problems, a confusion matrix has two rows and two columns, with one column representing positive instances and the other column representing negative instances. Rows show positive and negative predictions. Four confusion matrix cells represent:

True positives (TP): The algorithm's correct classification of positive instances.

False positives (FP): The algorithm incorrectly classifies negative instances as positive. True

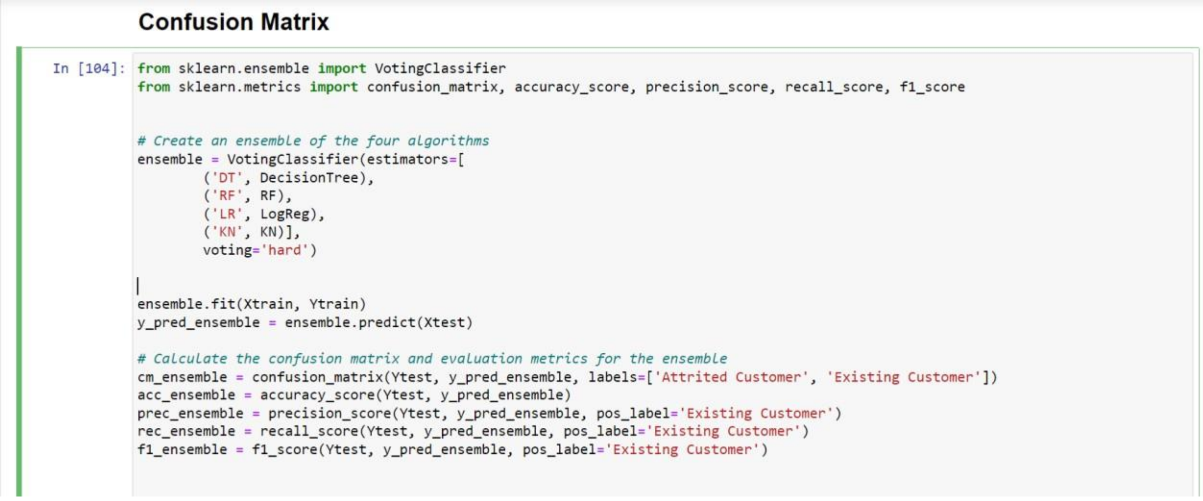
negatives (TN): Instances that the algorithm correctly classifies as negative.

False negatives (FN): The algorithm misclassifies positive instances as negative.

The confusion matrix shows a classification algorithm's strengths and weaknesses. If the algorithm has many false negatives, it may miss positive instances, which can be costly in some applications. However, a high number of false positives means the algorithm is misidentifying negative instances as positive, which can waste money and time.

We can calculate accuracy, precision, recall, and F1 score from the confusion matrix to evaluate the algorithm's performance.

This code selects specific columns from the "syn data" pandas DataFrame and then uses the "corr" function to create a correlation matrix. The Seaborn library is then used to create a heatmap that shows the relationships between the selected columns. The heatmap makes use of the "coolwarm" colormap and annotates the correlation coefficients. The plot is called "Correlation Matrix Heatmap". The plot is then shown using the "show" method from the matplotlib.pyplot package.



Confusion Matrix

```
In [104]: from sklearn.ensemble import VotingClassifier
          from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score


          # Create an ensemble of the four algorithms
          ensemble = VotingClassifier(estimators=[
                  ('DT', DecisionTree),
                  ('RF', RF),
                  ('LR', LogReg),
                  ('KN', KN)],
                  voting='hard')


          ensemble.fit(Xtrain, Ytrain)
          y_pred_ensemble = ensemble.predict(Xtest)

          # Calculate the confusion matrix and evaluation metrics for the ensemble
          cm_ensemble = confusion_matrix(Ytest, y_pred_ensemble, labels=['Attrited Customer', 'Existing Customer'])
          acc_ensemble = accuracy_score(Ytest, y_pred_ensemble)
          prec_ensemble = precision_score(Ytest, y_pred_ensemble, pos_label='Existing Customer')
          rec_ensemble = recall_score(Ytest, y_pred_ensemble, pos_label='Existing Customer')
          f1_ensemble = f1_score(Ytest, y_pred_ensemble, pos_label='Existing Customer')
```

Figure 4.3.1 Correlation Matrix in our project

4.4 Data Preparation

Cleaning, converting, and organizing raw data into a format appropriate for analysis and modeling is known as data preparation. Every machine learning (ML) model needs it, because the applicability and quality, of the training data, have a significant impact, on a model's performance.

Data preparation is particularly crucial in the case of customer churn prediction since the data used to train the model must precisely reflect the patterns and trends in customer behavior that generate churn. This may entail processing and cleaning the data to eliminate invalid or missing values, aggregating the data at the proper degree of detail, and choosing pertinent attributes that are most indicative of churn. In order to effectively account for unbalanced classes, the data may also need to be balanced. Additionally, variables may need to be standardized or scaled in order to guarantee that their influence on the model is taken into consideration. In general, accurate and trustworthy customer churn prediction models must be developed using good data preparation.

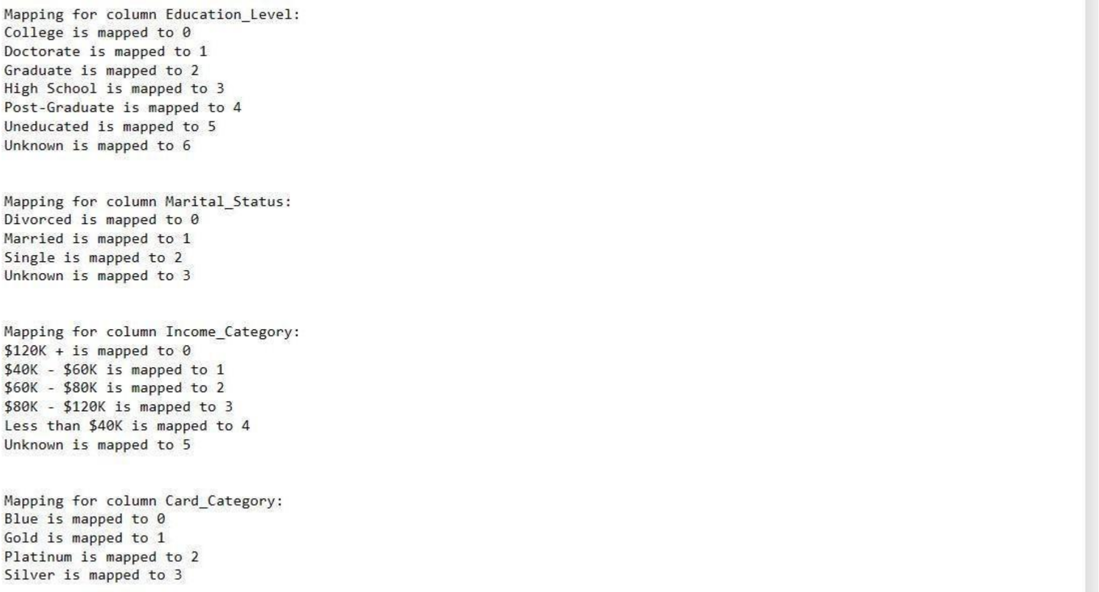
4.4.1 Mapping Categorial values into numerical values

To provide machine learning models the ability to meaningfully use categorical data as inputs, category values must be mapped into numerical values. Categorical values can be integrated into the calculations of the model by turning them into numerical values as most machine learning algorithms operate on numerical data.

Label encoding and one-hot encoding are two methods for converting category variables to numerical values. Each category value is given a distinct numerical value via label encoding, whereas one-hot encoding constructs a binary vector for each categorical value with a value of 1 in the appropriate element and 0s everywhere.

By converting categorical values into numerical values, the model is able to recognize and understand, the connections between various category values which is helpful for producing precise predictions. For instance, transforming categorical information like customer demographics or product categories into numerical values can assist find trends and correlations that lead to churn, enabling the model to make more precise predictions.

.



```
Mapping for column Education_Level:
College is mapped to 0
Doctorate is mapped to 1
Graduate is mapped to 2
High School is mapped to 3
Post-Graduate is mapped to 4
Uneducated is mapped to 5
Unknown is mapped to 6


Mapping for column Marital_Status:
Divorced is mapped to 0
Married is mapped to 1
Single is mapped to 2
Unknown is mapped to 3


Mapping for column Income_Category:
$120K + is mapped to 0
$40K - $60K is mapped to 1
$60K - $80K is mapped to 2
$80K - $120K is mapped to 3
Less than $40K is mapped to 4
Unknown is mapped to 5


Mapping for column Card_Category:
Blue is mapped to 0
Gold is mapped to 1
Platinum is mapped to 2
Silver is mapped to 3
```

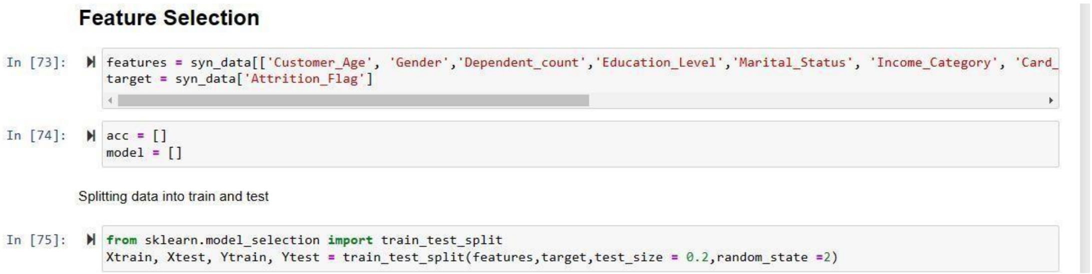
Figure 4.4.1 Mapping categorical values into numerical values

Since we need to provide the machine with some meaningful data so that it can train the model and make predictions all category categories in the aforementioned graphic have been converted into numerical values.

4.4.2 Feature Selection

The process of choosing a subset of pertinent features or variables to include in a machine learning model, or feature selection, is done from a larger collection of features. The purpose of feature selection is to enhance the performance of the model by lowering the amount of duplicate or irrelevant features, which can cause overfitting and limit model generalization.

Feature selection is important, in machine learning for several reasons. First, it can increase the model's effectiveness and computational complexity since using fewer features will need less processing power to train and execute the model. Second, it can make the model more interpretable since it will be simpler to find and comprehend, the aspects that are most predictive of the outcome when dealing, with a smaller collection of features.



Feature Selection

```
In [73]:  features = syn_data[['Customer_Age', 'Gender','Dependent_count','Education_Level','Marital_Status', 'Income_Category', 'Card_
           target = syn_data['Attrition_Flag']
```

```
In [74]:  acc = []
           model = []
```

Splitting data into train and test

```
In [75]:  from sklearn.model_selection import train_test_split
           Xtrain, Xtest, Ytrain, Ytest = train_test_split(features,target,test_size = 0.2,random_state =2)
```

Figure 4.4.2 Feature Selection

4.5 Modelling

Before Modelling the data we split the data into training and testing. We keep 80 % of data for training and 20 % for testing the model(Training and testing phase). We have used this split ratio because it gives a very proper output and more data is required for training at first. Once this is done we start to apply different algorithms on the split data. I have applied five algorithms that are Decision Tree, Random Forest, Logistic Regression, K-Nearest Neighbours, LazyPredict Algorithm.

4.5.1 Decision trees

What are Decision Trees?

In the area of machine learning and data analysis, the decision tree method is a well-liked and effective technique for tackling classification and regression issues. A decision tree is a representation of choices and potential outcomes, such as utility resource costs and chance event outcomes. Making predictions by segmenting a dataset into smaller and smaller subsets based on specific traits until the subsets are homogeneous or we reach a predetermined stopping threshold is a straightforward but efficient method.

The method iterates over each subset until it reaches a leaf node dividing the data recursively into subgroups based on the value of one of the input characteristics. Each node's property that delivers the most information gain, entropy reduction, or attribute that most clearly distinguishes the d The method iterates over each subset until it reaches a leaf node, dividing the data recursively into subgroups based on the value of one of the input characteristics. The feature that best separates the data into homogenous subsets, or one that delivers the highest information gain or entropy reduction at each node, is chosen by the algorithm.

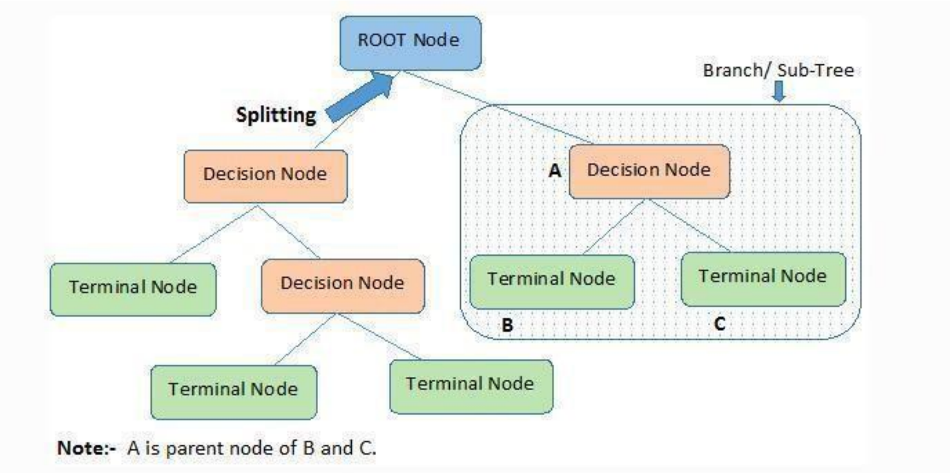


Figure 4.5.1(i) Algorithm Decision trees

The main advantage of the decision tree method is that it handles both continuous and categorical data is easy to comprehend & can be represented graphically. In addition, it is immune to outliers and can manage missing data. However, decision trees can overfit which happens when they get very complex and start, to fit the noise in the data instead of the underlying patterns. To address this issue, we can employ strategies like regularization, ensemble techniques, or pruning.

Overall, the decision tree technique is a strong and adaptable tool, with several applications in statistics data mining, and machine learning.

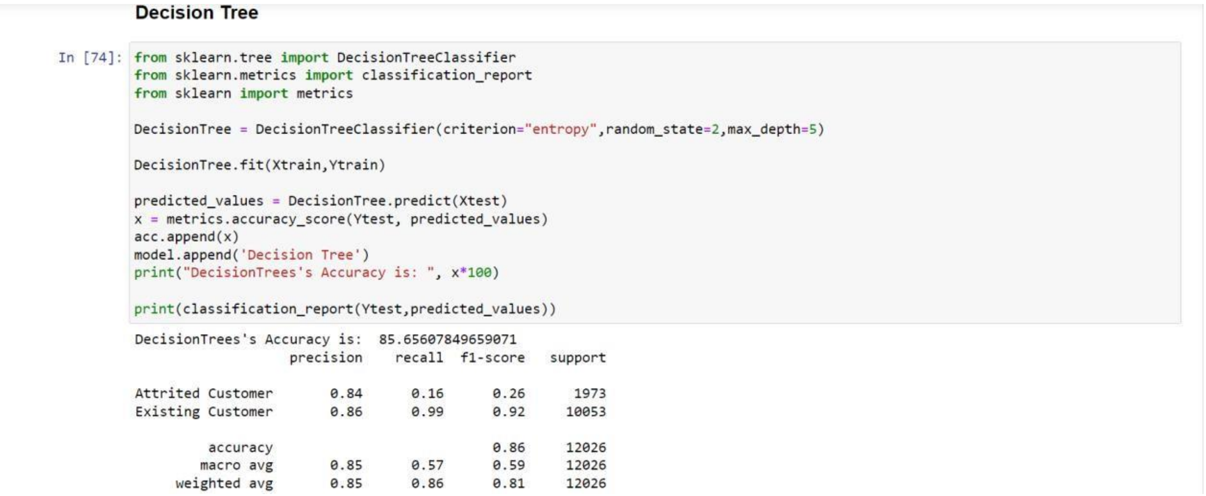
How we have used it in this project?

We may create a dataset that replicates customer behavior and their chance of churning in order to use fake data to train a decision tree for predicting customer turnover. This dataset may contain a variety of variables such as demographic data use trends purchase history, customer support interactions, and so on.

Finally, we can identify the critical factors that affect customer turnover, by utilizing this dataset to train a decision tree algorithm. The decision tree will recursively partition the dataset based on the values of these attributes assigning each leaf node a probability score, that indicates the likelihood of churn.

We can use the decision tree to forecast the likelihood of churn for new customers based on their feature values once it has been trained. We can identify a customer as being in danger of churning and take the necessary steps to keep them if the chance of churn reaches a certain level.

In general, organizations can identify the reasons that cause churn and take proactive steps to retain consumers by employing a decision tree for customer churn prediction using synthetic data.



Decision Tree

```
In [74]: from sklearn.tree import DecisionTreeClassifier
         from sklearn.metrics import classification_report
         from sklearn import metrics

         DecisionTree = DecisionTreeClassifier(criterion="entropy",random_state=2,max_depth=5)

         DecisionTree.fit(Xtrain,Ytrain)

         predicted_values = DecisionTree.predict(Xtest)
         x = metrics.accuracy_score(Ytest, predicted_values)
         acc.append(x)
         model.append('Decision Tree')
         print("DecisionTrees's Accuracy is: ", x*100)

         print(classification_report(Ytest,predicted_values))
```

```
DecisionTrees's Accuracy is:  85.65607849659071
                   precision    recall  f1-score   support

Attrited Customer       0.84      0.16      0.26      1973
Existing Customer       0.86      0.99      0.92     10053

         accuracy                           0.86     12026
        macro avg       0.85      0.57      0.59     12026
     weighted avg       0.85      0.86      0.81     12026
```

Figure 4.5.1(ii) Applied Decision trees

According to the figure above we get an accuracy of **85.6%** for our dataset.

4.5.2 Random Forest

What is Random Forest?

The popular and effective random forest approach is used in machine learning and data analysis to solve classification and regression issues. Several decision trees are combined in an ensemble learning technique to increase accuracy and lessen the overfitting of the model [20].

Using a collection of randomly chosen subsets of the data and features the method builds a series of decision trees. In order to increase the variety and decrease, the correlation between the trees each decision tree is trained on a fraction of the data using a random subset of characteristics. The output of the random forest is then determined, by averaging the forecasts made by each individual tree which aids in lowering variance and enhancing, the model's accuracy.

Additionally, the random forest method offers a feature importance metric that may be used to pinpoint the most crucial features that influence the prediction (Breiman, 2001). When a specific characteristic is used to separate the data the impurity of the nodes in the decision tree decreases which is the basis for how this is determined.

The random forest algorithm's key benefits are that it can handle categorical and continuous data is resistant to outliers and can deal with missing data. It is also easier to parallelize for faster training, on big datasets and less prone, to overfitting than a single decision tree.
Generally speaking, the random forest algorithm is a strong and adaptable tool, that has a wide range of uses in statistics data mining and machine learning.

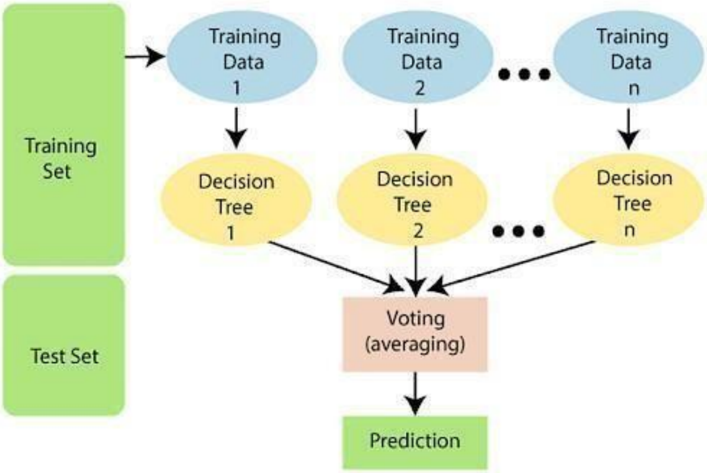


Figure 4.5.2(i) Random Forest Algorithm

How it is used in our project?

Following are the steps you may take to use random forest for artificial data to predict customer churn:

Create fictitious data: We can produce a dataset that resembles the actions and patterns of your customers' churn by using a data generating tool or approach. For instance, you may create data, using a statistical model or a tool like Faker.

To ensure that the synthetic dataset is in a machine learning-friendly state we should clean and preprocess it. This could involve category variable encoding resolving missing values and deleting duplicates.

To divide the dataset: For the purpose of evaluating the algorithm's performance divide the dataset into training & testing sets. The model should be trained using the training set and its effectiveness is assessed using the testing set.

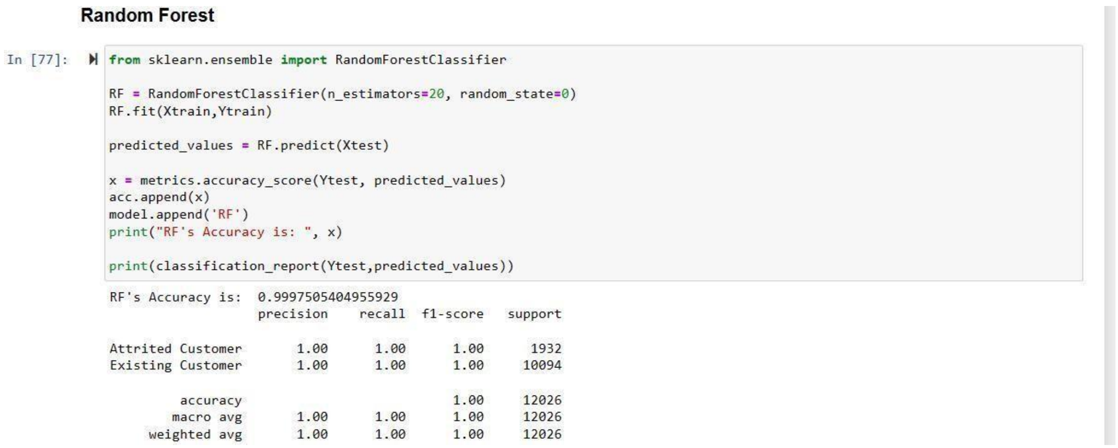
Training the random forest model: The random forest model should be trained using the training set. In order to do this, relevant characteristics must be chosen the number of trees must be specified and additional, hyperparameters must be provided.

Evaluation of the model: The testing set should be used to gauge how well the random forest model is working. To evaluate the model's performance, you can compute metrics like accuracy, precision, recall, and F1 score.

Modify the model: To enhance the performance of the random forest model, modify its hyperparameters. To determine the ideal hyperparameters you can employ strategies like grid search or randomized search.

Use the model: Based on new data we can use the model to estimate the likelihood, of customer churn after it has been trained and modified. This can assist you in taking preventative steps to stop customer churn and keep your current customers.

Overall, adopting random forest for artificial data-based customer churn prediction can help to retain customers. You can make a realistic training set to train the model by producing synthetic data that resembles the behavior and churn patterns of your customers. This can assist you in predicting customer turnover and implementing preventative strategies to keep clients.



Random Forest

```
In [77]:  from sklearn.ensemble import RandomForestClassifier

          RF = RandomForestClassifier(n_estimators=20, random_state=0)
          RF.fit(Xtrain,Ytrain)

          predicted_values = RF.predict(Xtest)

          x = metrics.accuracy_score(Ytest, predicted_values)
          acc.append(x)
          model.append('RF')
          print("RF's Accuracy is: ", x)

          print(classification_report(Ytest,predicted_values))
```

```
RF's Accuracy is:  0.9997505404955929
                   precision    recall  f1-score   support

Attrited Customer       1.00      1.00      1.00      1932
Existing Customer       1.00      1.00      1.00     10094

         accuracy                           1.00     12026
        macro avg       1.00      1.00      1.00     12026
     weighted avg       1.00      1.00      1.00     12026
```

Figure 4.5.2(ii) Applying Random Forest Algorithm

According to the figure above we get an accuracy of **99.9%** for our dataset.

4.5.3 Logistic Regression

What is Logistic Regression?

The popular technique known as logistic regression is used for binary classification situations where the output variable, may only take one of two potential values such as 0 or 1.

The technique works by modeling the connection between the input characteristics and the output variable using a logistic function. The logistic function may convert any input value, into a number between 0 and 1 which reflects, the likelihood that the output variable will be 1.

The approach determines the logistic function's parameters to train the logistic regression model, using a training dataset with known input features & output values. Following that the model makes use of these parameters to predict, the possibility that a fresh set of input data would produce an output variable of 1.

The procedure is known as logistic regression because it uses regression techniques to estimate the function and models, the likelihood of the output variable as 1 as a function of the input characteristics. By employing techniques like polynomial regression or feature engineering it is feasible to loosen the logistic regression model's requirement, that there be a linear connection between the input characteristics & the output variable.

Logistic regression's key advantage is that it is clear-cut easy to comprehend and it can handle both, categorical and continuous input data. Additionally, it is less susceptible to overfitting than other complicated models, like neural networks. It does, however, rely on the suppositions of the independence of the input characteristics & the linearity of their connection, to the output variable.

Overall, logistic regression is a powerful and well-liked approach, for binary classification problems in statistics and machine learning.

How we have used it in our project?

In reality, the logistic regression procedure is employed to look at correlations between variables. It assigns probabilities to discrete events, using the Sigmoid function which converts numerical outcomes, into an expression of probability between 0 and 1.0. Probability varies from 0 to 1 depending, on whether the event happens or not. For binary predictions, you can split the population, into two groups using a cut-off of 0.5. Every element in Group A exceeds, 0.5 but every element in Group B is below 0.5.

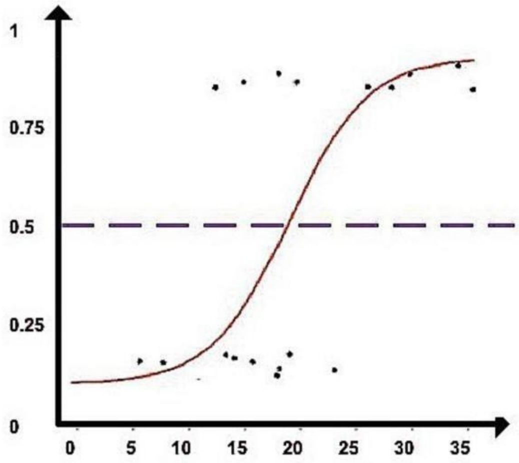


Figure 4.5.3 (i)Splitting the data

After utilizing the Sigmoid function to categorize data points, a hyperplane is employed (as much as possible) as a decision line to separate two groups. The nature of incoming data points may then be predicted using the decision boundary.

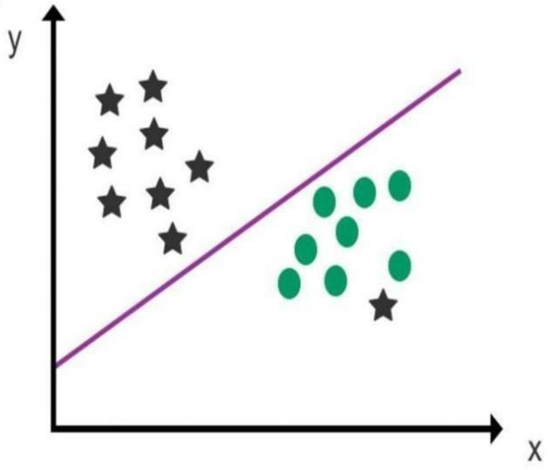
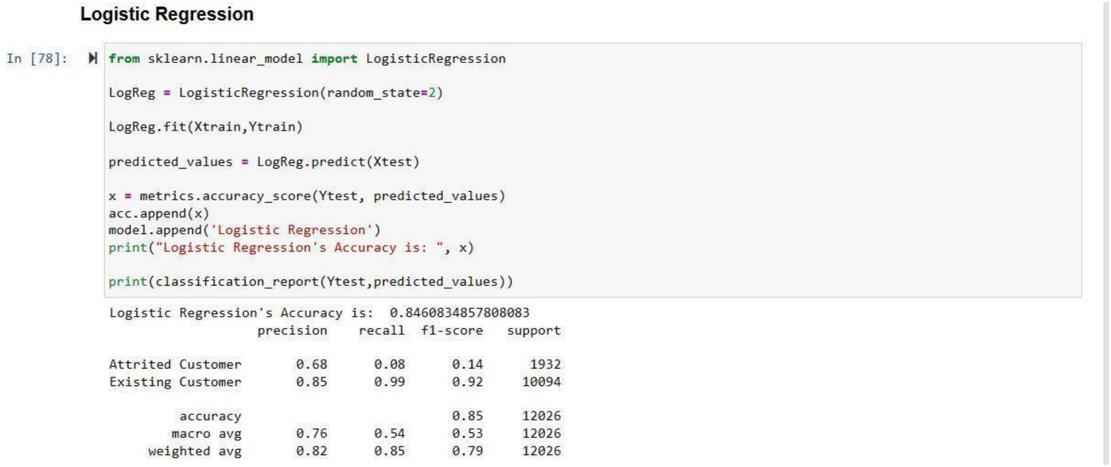


Figure 4.5.3(ii)Hyperplane formed

When using synthetic data for customer churn prediction, a predictive model that uses logistic regression can help identify the customers who are in danger of leaving. Synthetic data may be produced, using a variety of techniques such as bootstrapping resampling & oversampling. The produced synthetic data may be used to train the logistic regression model.

The logistic regression model forecasts the chance that a client will depart by using, the attributes of the consumer as input. By rating their customers according to this probability and utilizing, that information to identify which ones are most likely to depart businesses may focus, their retention efforts on the customers who are most likely to do so.

Always exercise caution when utilizing synthetic data since it occasionally results in false representations of the underlying patterns and distributions, in the real data. Therefore, it is suggested to evaluate, the logistic regression model's performance using, real data before applying it in a practical situation.



Logistic Regression

```
In [78]:  from sklearn.linear_model import LogisticRegression

          LogReg = LogisticRegression(random_state=2)

          LogReg.fit(Xtrain,Ytrain)

          predicted_values = LogReg.predict(Xtest)

          x = metrics.accuracy_score(Ytest, predicted_values)
          acc.append(x)
          model.append('Logistic Regression')
          print("Logistic Regression's Accuracy is: ", x)

          print(classification_report(Ytest,predicted_values))
```

```
Logistic Regression's Accuracy is:  0.8460834857808083
                   precision    recall  f1-score   support

Attrited Customer       0.68      0.08      0.14      1932
Existing Customer       0.85      0.99      0.92     10094

         accuracy                           0.85     12026
        macro avg       0.76      0.54      0.53     12026
     weighted avg       0.82      0.85      0.79     12026
```

Figure 4.5.3 (iii) Applying Logistic Regression Algorithm

According to the figure above we get an accuracy of **84.6%** for our dataset.

4.5.4 K Nearest Neighbors (KNN) ALGORITHM

What is KNN Algorithm?

K-nearest neighbors (KNN) is a simple yet powerful technique used in machine learning and data analysis to resolve classification and regression problems. The KNN method initially determines the K data points in the training set that are most similar to a particular test point in order to forecast its value. The technique is based on the assumption that data points with similar characteristics, would often belong to the same class or yield similar outcomes. The technique calculates the Manhattan distance or Euclidean distance, between each data point in the training set and the test point in order to identify the nearest neighbors. The next step is to select the K nearest neighbors based on the shortest distance.

The class that appears the most frequently among the K closest neighbors is the predicted class of the test point in classification problems. The average of the output values of the K closest neighbors serves as the predicted value of the test point for regression problems.

The key advantage of the KNN method is that it can handle categorical and continuous data and is clear and easy to use. However, it can be computationally expensive and necessitates careful K and distance metric selection.

Many classification and regression, issues in statistics, data mining & machine learning, can be solved using the KNN algorithm.

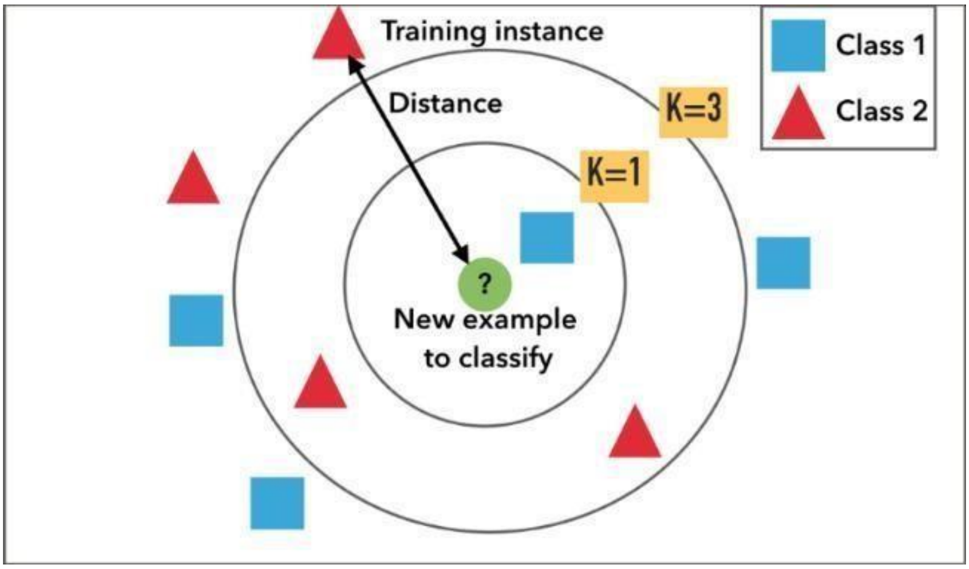


Figure 4.5.4 KNN ALGORITHM

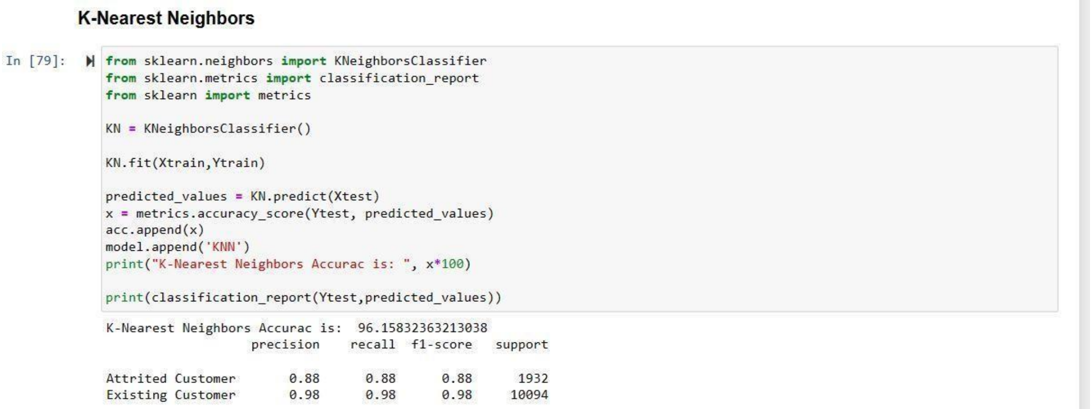
How we have used it in the project?

KNN (k-nearest neighbors) is a well-liked machine learning technique for classification and regression applications. Based on a number of input data such as demographics purchase history customer behavior & other factors KNN may be used to determine if a customer is likely to churn or not. To utilize KNN for customer churn prediction, we must first generate synthetic data that closely reflects the traits of your real-world customer data. We may use a variety of methods including data augmentation oversampling and generative models to produce synthetic data.

After dividing the synthetic data into training and testing sets the KNN technique may be used to train a classification model on the training set. The model may then be used to predict whether or not a test set client would depart using the input features.

When utilizing KNN it's important to pick the right value for k which specifies, how many nearest neighbors should be considered when producing a forecast. The model's performance can be significantly impacted by the choice of k which is frequently decided through trial and error or by using a cross-validation technique.

In conclusion, customer churn may be predicted using synthetic data by producing fake data that replicates the real-world data splitting the data into training and testing sets than using the KNN technique to train a classification model, on the training set.



K-Nearest Neighbors

In [79]:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report
from sklearn import metrics

KN = KNeighborsClassifier()

KN.fit(Xtrain,Ytrain)

predicted_values = KN.predict(Xtest)
x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('KNN')
print("K-Nearest Neighbors Accurac is: ", x*100)

print(classification_report(Ytest,predicted_values))
```

```
K-Nearest Neighbors Accurac is:  96.15832363213038
                   precision    recall  f1-score   support

Attrited Customer       0.88      0.88      0.88      1932
Existing Customer       0.98      0.98      0.98     10094
```

Figure 4.5.4 Applying KNN Algorithm

According to the figure above we get an accuracy of **96.1%** for our dataset.

4.5.5 LazyPredict Algorithm

Python's LazyPredict library automates model selection and training. By automating the model selection and training processes it was created to decrease the time and effort, needed to create and compare, various machine learning models. A single function, LazyClassifier from the LazyPredict library automatically trains and assesses, a variety of classification models on a labeled dataset. Logistic regression decision trees, random forests, support vector machines & many other classification models are supported, by the library.

In order to use the LazyClassifier function the input dataset must first be divided, into training and testing sets. The process then uses the training set, to train each classification model and the testing, set to assess each model's performance. Each model's performance is assessed, using a variety of metrics including accuracy, precision, recall, and F1 score.

The LazyClassifier function returns a summary of the models' performance after they have all been trained and evaluated, along with the model name, training time, and evaluation metrics. It is simple to determine which model performs, best because the function also ranks, the models according to performance.

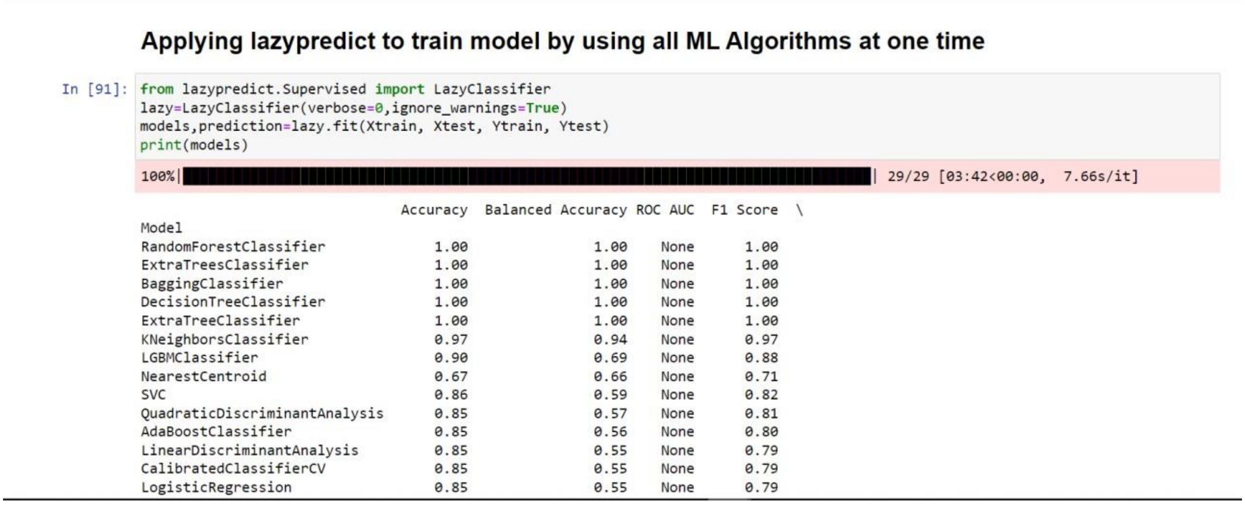
One of the key benefits of the LazyPredict library is its fast comparison of the performance of several models, on a given dataset. This can be particularly beneficial when it is uncertain, which model will be the most productive for a specific job or when there are insufficient resources to choose and train, the appropriate model.

You can develop and compare machine learning models more quickly with the aid of the powerful and user-friendly tool LazyPredict. Due to its automated model selection and training procedure, it is the ideal choice for academics & practitioners who wish to swiftly evaluate, the effectiveness of several models on a certain dataset.

How we have used it in this project?
Before utilizing LazyPredict to forecast customer attrition we first built the synthetic data using techniques like data augmentation oversampling or generative models. We may examine the effectiveness of several machine learning algorithms on simulated data using LazyPredict. The library supports a broad range of classification techniques, including KNN, Decision Trees, Random Forest, Gradient Boosting, and others.

We install the library and import it into your Python code before using LazyPredict. Your fictitious data would then be loaded into a pandas Data Frame and divided, into training and testing sets. The performance of various classification algorithms would then be assessed using LazyPredict on the training set and their performances, would be compared using evaluation metrics like accuracy, precision, recall, and F1 score.

Additionally, LazyPredict gives you the choice to use grid search to fine-tune the hyperparameters of the chosen algorithms enhancing the performance of the models even more.
In conclusion, by creating synthetic data, loading it into a pandas. DataFrame dividing it into training and testing sets & using LazyPredict, to assess the performance of various machine learning algorithms on the training set it is possible to use, synthetic data to predict customer churn.



Applying lazypredict to train model by using all ML Algorithms at one time

```
In [91]: from lazypredict.Supervised import LazyClassifier
         lazy=LazyClassifier(verbose=0,ignore_warnings=True)
         models,prediction=lazy.fit(Xtrain, Xtest, Ytrain, Ytest)
         print(models)
```

```
100%|██████████████████████████████████████████████████████████████████████████| 29/29 [03:42<00:00,  7.66s/it]
```

```
                               Accuracy  Balanced Accuracy ROC AUC  F1 Score  \
Model
RandomForestClassifier             1.00               1.00    None      1.00
ExtraTreesClassifier               1.00               1.00    None      1.00
BaggingClassifier                  1.00               1.00    None      1.00
DecisionTreeClassifier             1.00               1.00    None      1.00
ExtraTreeClassifier                1.00               1.00    None      1.00
KNeighborsClassifier               0.97               0.94    None      0.97
LGBMClassifier                     0.90               0.69    None      0.88
NearestCentroid                    0.67               0.66    None      0.71
SVC                                0.86               0.59    None      0.82
QuadraticDiscriminantAnalysis      0.85               0.57    None      0.81
AdaBoostClassifier                 0.85               0.56    None      0.80
LinearDiscriminantAnalysis         0.85               0.55    None      0.79
CalibratedClassifierCV             0.85               0.55    None      0.79
LogisticRegression                 0.85               0.55    None      0.79
```

Figure 4.5.5 Applying lazypredict Algorithm

Chapter 05 Evaluation

We start this chapter by measuring the accuracy of the results and knowing the difference between accuracy and cross validation. In this Chapter we will study how to create a webpage in flask and how to do customer churn prediction on that website.

5.1 Cross Validation

Cross-validation is a statistical method for assessing, how well a prediction model performs. It's a technique that's widely used in data analysis and machine learning and it's especially essential for dissertations, that require modeling or prediction [21].

A training set and a validation set are created from the given data, to allow for cross-validation. The predictive model's performance is assessed using, the validation set after it has been trained using the training data. The objective of cross main validation is to evaluate, the generalizability of the model not merely its ability to match, the training set of data.

Although there are many different cross-validation methods k fold cross-validation is one of the most popular. The data is split into k equal-sized subsets or folds for k-fold cross-validation. The model is then trained k times with each training iteration using the remaining k-1 folds as the training set and a different fold as the validation set. The model's performance is then averaged across the k folds.

Cross-validation is a potent technique because it enables us to gauge how well the predictive model performs on fresh untested data. Because it enables you to assess the reliability and generalizability of our model, this is crucial for dissertations. The model is likely to perform well on new data if it performs well on the validation data.

Cross-validation is a helpful method for predictive modeling dissertation research. Testing the effectiveness of your model on new, previously unexplored data will allow us to more precisely evaluate the validity and generalizability of your findings and reach more trustworthy conclusions. All algorithms were subjected to cross-validation, and I determined the algorithms' correctness (Kohavi, 1995).

5.2 Saving the models

Keeping a trained model in a file or format so that it may be used to generate predictions on new data later on is referred to as "saving a model" in machine learning. Since reuse is made possible without further training needed for each desired prediction, saving a model is crucial. This can save a lot of time and processing power particularly, if the model is complex and requires, a lot of information or processing power to train. In our project we have saved each model into a pickle file(.pkl file) which will be used for deployment later.

In machine learning saving a model provides a variety of benefits including:

Reusability: You can use a model repeatedly, without having to retrain it to generate predictions based, on fresh data by preserving it. This can save a lot of time & money especially if the model is sophisticated and requires, a lot of data or computing power to train.

Replicability: You may subsequently duplicate your study and conclusions by storing a model. This is particularly useful if you want to share, your analysis with others or if you need to go back & examine it.

Flexibility: You may utilize a saved model in a variety of settings and programs. You might be able to use the same model to generate predictions on different types of data or across different areas.
Collaboration: By storing a model you may share it, with others so they can use it, or build on it, to create new models. Machine learning model saving is an important phase in the model-building process, due to the potential to reuse a model, so saves time & money.

5.3 Accuracy of Algorithms

A statistic used in machine learning to evaluate, how effectively a classification algorithm is doing is accuracy [22]. It displays the percentage of cases in the dataset that were properly categorized in comparison, to all of the examples. The ability of a machine learning model to categorize instances in accordance with the input data is measured by accuracy, to put it another way. It is an essential indicator for evaluating a model's efficacy since it demonstrates how effectively the model is able to anticipate the future.

Finding accuracy is important since it demonstrates if a model is effective in solving the categorization problem that it was designed to address. Low accuracy could indicate, that the model is inefficient forcing additional model improvement or the selection of an alternative approach. On the other hand, high accuracy demonstrates the model's effectiveness and ability to be trusted to generate predictions based on new data.

The ideal measure to employ isn't always accurate it's crucial to keep this in mind. Other measures such as accuracy, recall, or F1 score could in certain cases be more appropriate depending on the particular problem being solved and the type of data.

5.3.1 Accuracy Comparison

The process of comparing the performance of various machine learning algorithms, based on their accuracy scores is known as accuracy comparison. It is essential because it aids in determining, which algorithm is most suitable, for a specific classification issue.

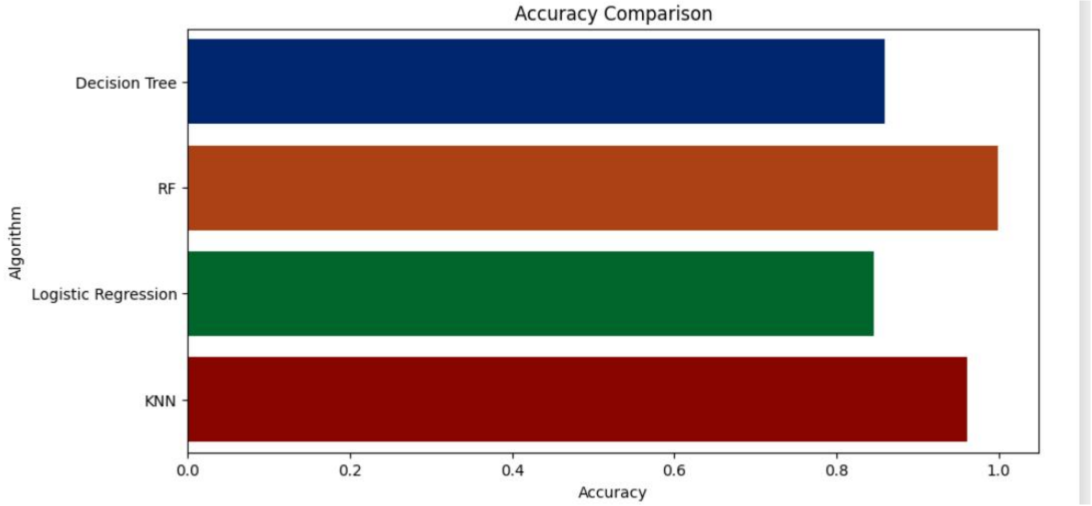


Figure 5.3.1 Accuracy Comparison

No.	MACHINE LEARNING ALGORITHM	ACCURACY PERCENTAGE
1	**DECISION TREES**	85.86 %
2	**RANDOM FOREST**	99.8%
3	**K NEAREST NEIGHBOURS**	96.14%
4	**LOGISTIC REGRESSION**	84.31%
5	**LAZYPREDICT**	Maximum Random forest

Decision Tree: 85%

In my case, the decision tree algorithm achieved an accuracy of 85%. This means that the model correctly classified 85% of the examples in the dataset. While this accuracy score is relatively high it may still leave room for improvement, depending on the specific problem being solved.

Random Forest: 99%

In my case, the random forest algorithm produced an extremely high accuracy of 99%. This shows that the model correctly classified, nearly all of the samples in the dataset.

Logistic Regression: 84%

In my case, the logistic regression technique attained an accuracy of 84% which is also pretty high. This implies that the model accurately categorized 84% of the items in the dataset. While this accuracy score is significantly lower than the other algorithms, I examined it is still a solid result.

K-Nearest Neighbors (KNN): 96%

The accuracy of the KNN algorithm in my case was 96% which is pretty good. This indicates that 96% of the cases in the dataset were properly categorized, by the model. KNN is renowned for being straightforward & simple to use in my case, it seems to have worked extremely well.

Overall, I scored all of the algorithms I tested with excellent accuracy which is a positive indicator. It's crucial to remember though that when assessing a model accuracy should not be the only factor to take into account. There may also be other crucial elements to take into accounts such as the algorithm's interpretability, resilience, and computational complexity.

5.4 Stacking

Instead of utilizing just one model, the machine learning approach known as stacking combines numerous models to give a single prediction. In the process of stacking, a sort of model aggregation, models are trained individually then their predictions are integrated using a meta-model also known as a blender. [23]

Stacking: Why

To enhance the predictive performance of the model stacking may need to be applied to the machine learning algorithms. By combining the qualities of several models, stacking may be used to overcome their flaws. By merging the outcomes of many models, we may get a prediction that is more accurate and trustworthy. There are several factors that might influence our decision to employ stacking:

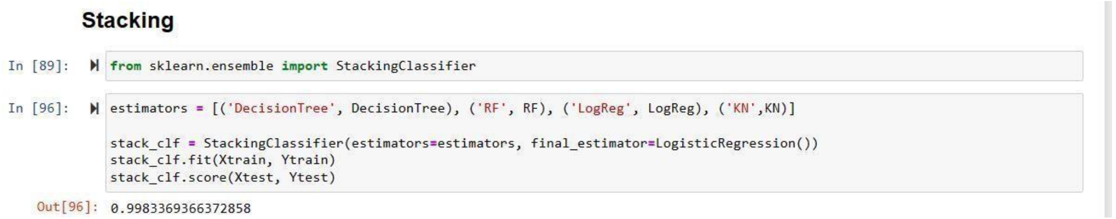
Enhanced prediction accuracy:
We can increase the precision of the final prediction by combining the predictions of various models. When the base models are complementary, or when they make errors on different portions of the dataset, stacking can be especially effective.

Robustness:
Using stacking can strengthen the model's resistance to adjustments to the dataset or the modeling procedure. We can lessen the effect of outliers or data noise by combining the predictions of various models.

Flexibility:
Because stacking works with any machine learning method, it is a flexible technique that may be applied to a wide range of issues.

Interpretation of a model:
Stacking can shed light on the relative importance of various data features. We can determine, which features are most crucial for the outcome prediction by looking at the weights, that the meta-model assigned to each base model.

Calibration of the model:
It is possible to calibrate the predictions of various models using stacking. We can adjust the probabilities given to each class and get a more accurate prediction by combining the predictions of various models with a meta-model.



Stacking

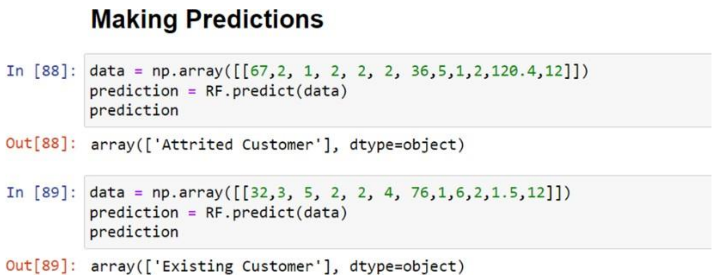
```
In [89]:  from sklearn.ensemble import StackingClassifier

In [96]:  estimators = [('DecisionTree', DecisionTree), ('RF', RF), ('LogReg', LogReg), ('KN',KN)]

          stack_clf = StackingClassifier(estimators=estimators, final_estimator=LogisticRegression())
          stack_clf.fit(Xtrain, Ytrain)
          stack_clf.score(Xtest, Ytest)

Out[96]:  0.9983369366372858
```

Figure 5.4 Applying Stacking

In the figure it can seen that stacking is applied on all four algorithms and it gives 99% Accuracy .

5.5 Test cases to check prediction

Passed an Array for each column and used the Random forest to predict if the customer is an Attrited customer or an existing customer.



Making Predictions

```
In [88]: data = np.array([[67,2, 1, 2, 2, 2, 36,5,1,2,120.4,12]])
         prediction = RF.predict(data)
         prediction

Out[88]: array(['Attrited Customer'], dtype=object)

In [89]: data = np.array([[32,3, 5, 2, 2, 4, 76,1,6,2,1.5,12]])
         prediction = RF.predict(data)
         prediction

Out[89]: array(['Existing Customer'], dtype=object)
```

Figure 5.5 Test cases

IMPLEMENTATION PHASE 3

5.6 Deployment

A Flask application is deployed when it is made accessible for use on a production server. While Flask applications are typically created and tested on a local development server deploying them to a real-world setting necessitates further preparation and thought. [24]

When deploying Flask application common tasks include:
Setting up the database defining environment variables and defining logging options are all parts of configuring the application environment.

Selecting a production server: A number of production servers including Apache, Nginx, & Gunicorn are compatible with Flask. The choice will depend on the particular requirements, of the application and the benefits and drawbacks of each server.
It must be configured so that it may work with the Flask application when a production server has been chosen. It is necessary to provide the location of the application code the configuration of the server to handle incoming requests & any security settings.
Setting up deployment automation: Because manually deploying a Flask application may be time-consuming & error-prone it's usually a good idea to set up an automated deployment procedure. Using technology like Jenkins or Travis CI the application may be automatically created and deployed, whenever changes are made to the code. Overall, Flask application deployment calls for meticulous planning and close attention, to detail but with the right tools and approaches, it can be a simple procedure.

5.7 Choosing a deployment environment

My project to estimate customer turnover will be deployed using Flask.
A popular lightweight and adaptable Python web framework Flask is known for its simplicity and ease of use. Flask is a fantastic choice for deployment for a variety of reasons:
It is easy to understand Flask is designed to be easy to understand and use even for programmers, with no prior experience with Python or web programming. Its basic approach to web development enables developers to get started fast and saves them time learning a lot of complicated ideas or technologies.

Flask is very adaptable and can be used to create a wide range of web applications from straightforward static websites to intricate ones, with dynamic content and intricate business logic.

Lightweight: Flask is a small-footprint web framework that can be installed on servers with limited resources. This makes it a good option for developing applications that must operate on little servers or in environments, with few resources like the cloud. Flask is very extensible and is simple, to customize to meet the unique requirements of an application. For Flask there are numerous 3rd party libraries & plugins that can increase functionality and simplify development.

Popular: Flask is a very well-liked web framework, so there is a sizable developer and user community that can offer support and guidance. This also implies that a wealth of online materials such as documentation instructions and code samples are accessible.
In general, Flask is a great option for deployment because of its acceptance, usability, flexibility, minimal weight & adaptability. Because of these qualities, it is a wonderful framework for building online applications, that need to be delivered rapidly and successfully.

5.7.1 Flask setup

Before we are going to deploy our model, we need to set up FLASK in the Visual Studio code. There are some necessary steps for FLASK setup.

1)Create a virtual environment in our project folder.
2)Install FLASK in our project folder.
3)Create an app.py file that contains the basic function that redirects to the browser.
4)Create a Static folder for images and any designing files like CSS, and JS.
5)Create a Templates folder for HTML files.

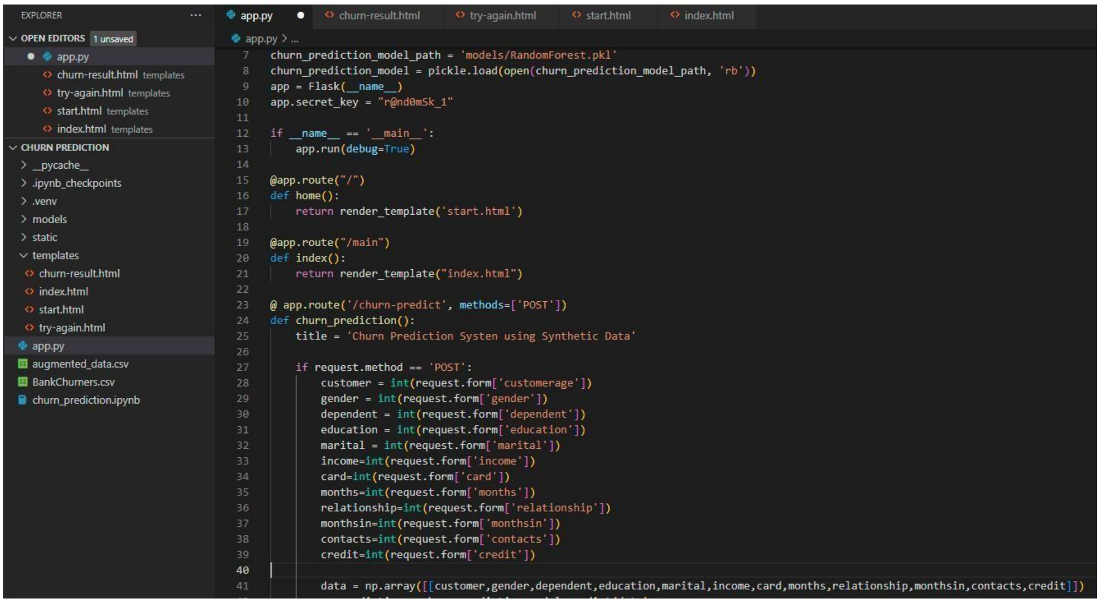
5.8 Deployment Sequence

To forecast client turnover, the Flask web application employs a trained machine learning model. The machine learning model is a Random Forest model that has been trained using fictional data.
The trained model is loaded by the code from the "RandomForest.pkl" file using the pickle module.
After that, three routes are established to handle HTTP requests & the Flask application is launched.
The "home" route returns a rendered HTML template for the web application's home page.

A rendered HTML template is returned by the index route of the web application's home page.
The 'churn prediction' route is triggered when a user accepts a form on the homepage asking for customer information. The route retrieves the form data from the POST request and uses it to feed the machine learning model which has previously been trained to forecast with form data. The projected result is then displayed to the user in the 'churn-result.html' template.
Overall, this code demonstrates how to integrate a trained machine learning model to forecast customer turnover into a Flask web application.

5.8.1 Setting up App.py file

For putting up any website in FLASK, this file serves as the foundation. The primary operations and file views are located in the app.py file. The model path is also in this file and there is a distinct function for each file that renders it to a certain page.



```python
churn_prediction_model_path = 'models/RandomForest.pkl'
churn_prediction_model = pickle.load(open(churn_prediction_model_path, 'rb'))
app = Flask(__name__)
app.secret_key = "r@nd0mSk_1"

if __name__ == '__main__':
    app.run(debug=True)

@app.route("/")
def home():
    return render_template('start.html')

@app.route("/main")
def index():
    return render_template("index.html")

@ app.route('/churn-predict', methods=['POST'])
def churn_prediction():
    title = 'Churn Prediction Systen using Synthetic Data'

    if request.method == 'POST':
        customer = int(request.form['customerage'])
        gender = int(request.form['gender'])
        dependent = int(request.form['dependent'])
        education = int(request.form['education'])
        marital = int(request.form['marital'])
        income=int(request.form['income'])
        card=int(request.form['card'])
        months=int(request.form['months'])
        relationship=int(request.form['relationship'])
        monthsin=int(request.form['monthsin'])
        contacts=int(request.form['contacts'])
        credit=int(request.form['credit'])

        data = np.array([[customer,gender,dependent,education,marital,income,card,months,relationship,monthsin,contacts,credit]])
```

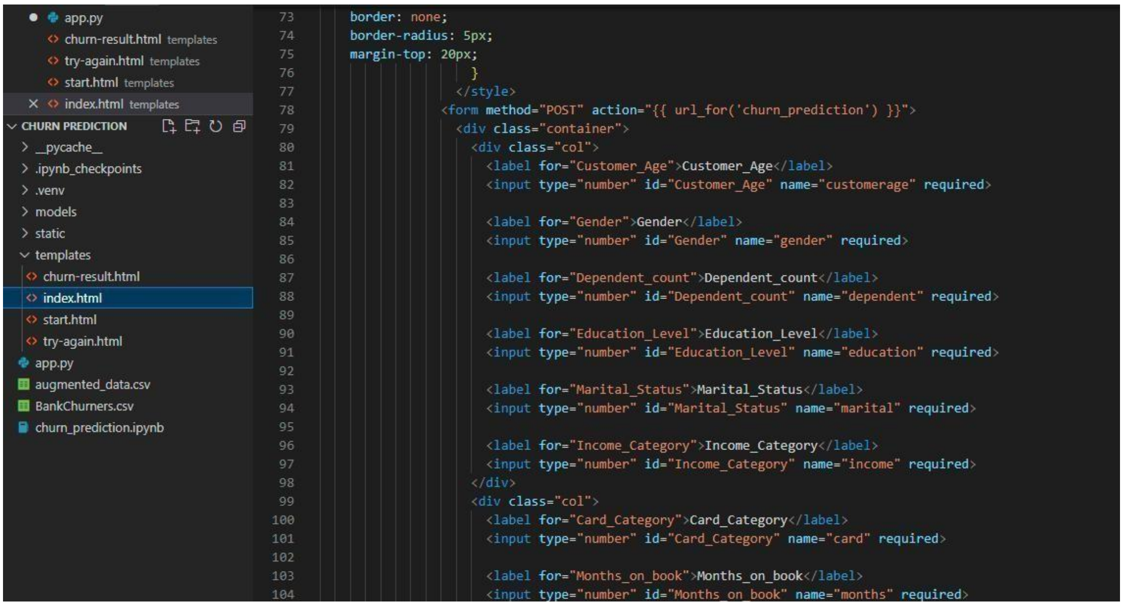
Figure 5.8.1 Initializing App.py file

In the figure-5.7.1 first of all the path of the model is specified where it can be read and used further for prediction. There are route functions that specify the root of a specific file where the exact file is and it's all content in that HTML file.

The main function is churn_prediction which is a form that used the POST method and takes input from users like customer age, gender, education, card category, dependent count, marital status, income, credit limit, etc. When the user enters this information about a user then the model predicts the value that it existing customer or attrited customer.

5.8.2 Setting up Index.html file

Index.html is the main file of the website which has input from content & all the information, related to the model.

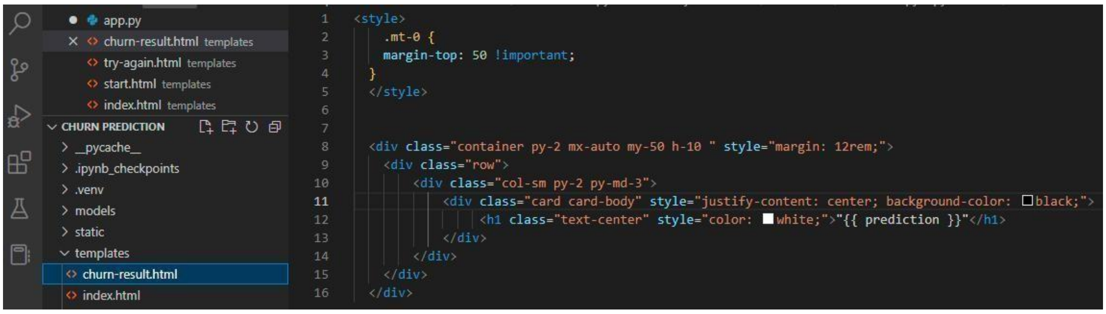


```html
border: none;
border-radius: 5px;
margin-top: 20px;
                }
            </style>
          <form method="POST" action="{{ url_for('churn_prediction') }}">
            <div class="container">
              <div class="col">
                <label for="Customer_Age">Customer_Age</label>
                <input type="number" id="Customer_Age" name="customerage" required>

                <label for="Gender">Gender</label>
                <input type="number" id="Gender" name="gender" required>

                <label for="Dependent_count">Dependent_count</label>
                <input type="number" id="Dependent_count" name="dependent" required>

                <label for="Education_Level">Education_Level</label>
                <input type="number" id="Education_Level" name="education" required>

                <label for="Marital_Status">Marital_Status</label>
                <input type="number" id="Marital_Status" name="marital" required>

                <label for="Income_Category">Income_Category</label>
                <input type="number" id="Income_Category" name="income" required>
              </div>
              <div class="col">
                <label for="Card_Category">Card_Category</label>
                <input type="number" id="Card_Category" name="card" required>

                <label for="Months_on_book">Months_on_book</label>
                <input type="number" id="Months_on_book" name="months" required>
```

Figure 5.8.2 Index.html file

5.8.3 Setting up churn-result.html file

The result of user input shown on this page and here is the code of this page.

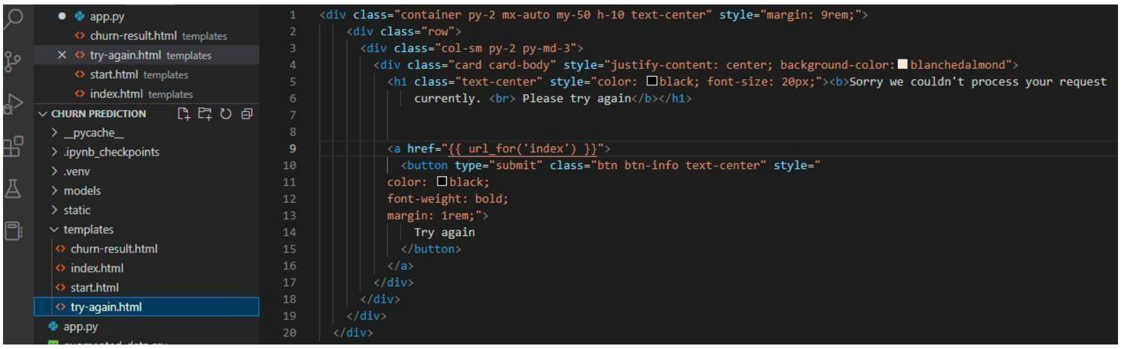


```
<style>
    .mt-0 {
    margin-top: 50 !important;
  }
  </style>


  <div class="container py-2 mx-auto my-50 h-10 " style="margin: 12rem;">
    <div class="row">
        <div class="col-sm py-2 py-md-3">
            <div class="card card-body" style="justify-content: center; background-color: black;">
                <h1 class="text-center" style="color: white;">"{{ prediction }}"</h1>
            </div>
        </div>
    </div>
  </div>
```

Figure 5.8.3 Churn-result.html file

5.8.4 Setting up try-again.html file

When the user inputs incorrect data or the model fails for whatever reason a try-again page will appear.



```
<div class="container py-2 mx-auto my-50 h-10 text-center" style="margin: 9rem;">
    <div class="row">
      <div class="col-sm py-2 py-md-3">
        <div class="card card-body" style="justify-content: center; background-color:blanchedalmond">
          <h1 class="text-center" style="color: black; font-size: 20px;"><b>Sorry we couldn't process your request
            currently. <br> Please try again</b></h1>


          <a href="{{ url_for('index') }}">
            <button type="submit" class="btn btn-info text-center" style="
          color: black;
          font-weight: bold;
          margin: 1rem;">
            Try again
            </button>
          </a>
        </div>
      </div>
    </div>
  </div>
```

Figure 5.8.4 Try Again.html page

5.9 Running these files

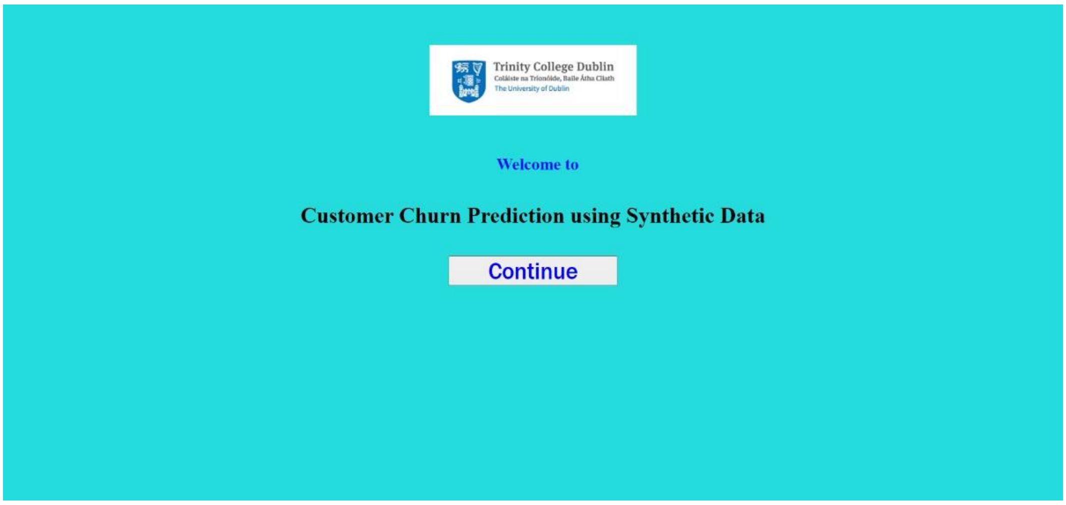
We begin by entering the command "python -m flask run" into the visual studio code terminal. The "python -m flask run" command starts a Flask application.
This message is often displayed when a local web server is started & is presently listening, on port 5000 at the IP address localhost (127.0.0.1). This indicates that the web server will handle any requests made, to that IP address and port.

Python is the name of the Python interpreter, that is required to run, the Flask application. "-m flask," tells Python to launch the "flask" module. Using the command "run," the Flask development server is started. The Flask application will start up and be served by default on port 5000 once you input this command into your terminal. The output in the terminal should now display the URL for the application, and the development server ought to have already begun.

Please be aware that this command assumes, that you have previously created a Flask application & that the Flask library has been set up, in your Python environment. The local URL that is automatically used, to serve the Flask application when you execute the "python -m flask run" command is "http://127.0.0.1:5000".
The loopback address for your local computer is "127.0.0.1" often known as its IP address "5000" which is the application-serving port that Flask uses by default.
To open a web browser & visit, the Flask application put, "http://127.0.0.1:5000" into, the address bar. The right HTML template will then be rendered by the server or the proper data will be sent back to the browser in response. The Flask development server running on your local PC will get a request from this.

5.10 Website

This is the first page that shows up after the running of the command.

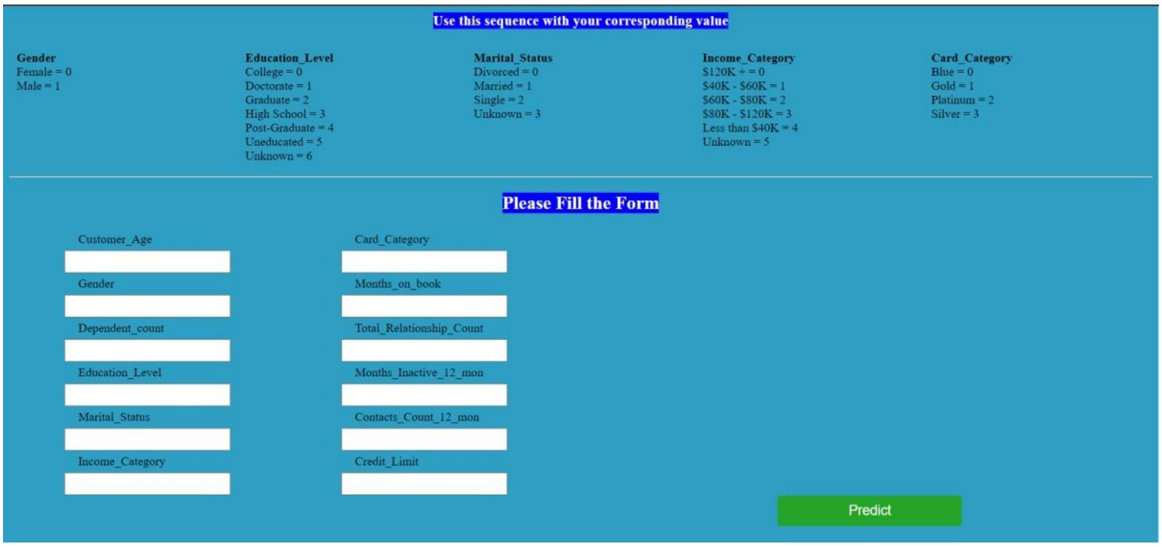


Welcome to

Customer Churn Prediction using Synthetic Data

Figure 5.10(i) First Page

The figure below is the deployed site of customer churn prediction. In the first section of the page, there are some labeling values of categorical columns like Gender, Education_level, Marital_Status, Income_Category, and Card_Category. Values of these columns are labeled in numerical values because the machine learning model only understands numerical values that’s why the user needs to enter numerical values as input and the model gives the output to the user.
Secondly, there is a form that has input fields of columns like Gender, Education_level, Dependent_Count, etc which the user needs to fill up, and then by pressing the predict button we will get the output based on the input values.



Use this sequence with your corresponding value

Gender	Education_Level	Marital_Status	Income_Category	Card_Category
Female = 0	College = 0	Divorced = 0	$120K + = 0	Blue = 0
Male = 1	Doctorate = 1	Married = 1	$40K - $60K = 1	Gold = 1
	Graduate = 2	Single = 2	$60K - $80K = 2	Platinum = 2
	High School = 3	Unknown = 3	$80K - $120K = 3	Silver = 3
	Post-Graduate = 4		Less than $40K = 4	
	Uneducated = 5		Unknown = 5	
	Unknown = 6			

Please Fill the Form

Customer_Age

Gender

Dependent_count

Education_Level

Marital_Status

Income_Category

Card_Category

Months_on_book

Total_Relationship_Count

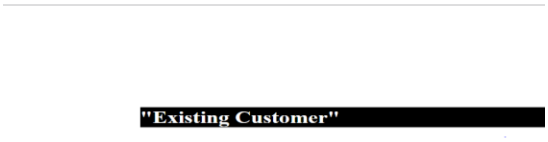
Months_Inactive_12_mon

Contacts_Count_12_mon

Credit_Limit

Figure 5.9 (ii) Second Page

The Results are shown on this page that can be Existing Customer and Attrited Customer.



"Existing Customer"

Chapter 06 Conclusions & Future Work

This chapter provides a summary of the dissertation report highlighting the major contributions made, by this project to the existing body of work as well as the system's limitations. It also suggests areas for improvement & further, research in the future.

6.1 Limitations of the project

Consumer churn prediction systems have certain drawbacks despite the fact that they can offer useful insights into consumer behavior and aid businesses in increasing customer retention. The following are a few of the customer churn prediction systems drawbacks:

The following disadvantages of utilizing artificial data to forecast client attrition should be taken into account:

Accuracy: Synthetic data may not always correctly, reflect the true distribution of the real data. Predictions based on manufactured data may thus be less exact than predictions based on genuine data.

Real-world influences: Synthetic data may fail to account for real-world influences on customer behavior such as economic fluctuations or competing company offerings.

Overfitting: If the synthetic data are not reflective of the real data the model may be overfitting preventing it from generalizing successfully to new data.

Ethics issues: It is probable that manufactured data is skewed and does not represent the true population. The use of such data to make choices that affect actual consumers raises ethical concerns. Synthetic data may only be used in circumstances when it provides an accurate depiction of the real data. For example, it may not be suitable for predicting unusual events or outliers.

Data complexity: Synthetic data might be unable to fully represent the complexity of real data such as the subtleties of customer behavior or the interactions between variables.

As a result, it's critical to assess the drawbacks of using synthetic data, to predict customer churn and to take into account any potential trade-offs between accuracy, moral propriety, and data complexity. Additionally, it is crucial to carefully interpret the model's results after validating them using actual data.

6.2 Future Work

It is important to compare the effectiveness of this approach to using real-world data given that I am training my model on synthetic data. My successor may start by comparing the performance of models created with synthetic data with those created using real data.

Investigate the results of several synthetic data generation techniques: Only a few techniques are available for creating synthetic data, like SMOTE and GANs. My successor can research how different approaches impact the churn prediction model's accuracy. I had attempted to use these models but at the time, I was still studying these courses.

Investigate various machine learning algorithms: Try out various machine learning techniques to see if they can enhance the performance of the churn prediction model, such as support vector machines.

Execute feature engineering in greater detail: Feature engineering is essential to the creation of a successful churn prediction model. My successor can investigate various feature engineering methodologies and determine which features are most crucial for churn prediction.

Examine the model's performance over time: Churn-inducing elements are dynamic phenomena that are subject to alter throughout time. My successor will be able to evaluate the model's performance over time to ensure that it continues to accurately anticipate churn.

Put the model to use in a real-world situation: Examine the churn prediction model's performance in a real-world environment and the effects it has on financial results like customer retention and revenue.

My successor should start by reading up on customer churn prediction and methods for creating synthetic data in order to know where to begin. They should also become acquainted with the dataset and machine learning techniques I used. From there, they can begin experimenting with various methods and algorithms to raise the churn prediction model's accuracy.

6.3 Security and privacy concerns

When using customer churn prediction using synthetic data, certain security and privacy issues need to be considered. These concerns consist of:

Data breach: It could contain private client information including personal information and transaction history using synthetic data raises the possibility of a data breach. In the case of a data breach, this information may be disclosed to uninvited parties which might result in identity theft financial loss & reputational harm.

Data privacy: Synthetic data could not provide the same level of privacy protection for customers as genuine data. This is because synthetic data might contain patterns and trends that are similar to those in actual data and can therefore be used to identify individual consumers. Synthetic data is created by algorithms that are trained on real data.

Bias: Synthetic data may be skewed against, particular consumer groups which might lead to unjust treatment or discrimination, against these groups. For instance, the synthetic data might not truly reflect the complete client base if just specific demographics or geographical, areas were utilized to construct it.

Accuracy: As was previously said synthetic data may occasionally not precisely reflect the true distribution of the real data, which might lead to incorrect forecasts or judgments that could harm customers.

Legal compliance: Using synthetic data to forecast customer turnover may offer legal compliance difficulties since there may be regulations or standards controlling the usage of consumer data in this circumstance. If we break these guidelines we might face financial or legal consequences.

To address these security and privacy concerns it is critical to employ appropriate data security mechanisms such as encryption and access restrictions, to secure synthetic data. It is also critical to guarantee that the synthetic data is developed in a fair and transparent way and tested against genuine data in order to assure correctness. Last but not least when utilizing synthetic data to anticipate customer attrition, it is vital to follow all applicable legal and regulatory obligations.

6.4 Conclusion

The project began by creating synthetic data from bank customer data. Machine learning algorithms are used to produce new data that mimics the original data to create synthetic data. When actual data is unavailable or additional data, is required to train machine learning models, this strategy might be effective.

Following the creation of synthetic data, data analysis & preprocessing were performed on the data to prepare it for usage in machine learning algorithms. To prepare data for modeling data preparation includes a variety of approaches such as data cleansing, feature selection, normalization, and transformation.

The preprocessed data were subjected to the decision tree, random forest, logistic regression, and KNN machine learning algorithms. These algorithms were selected because they are often used in predicting customer attrition and might offer a useful baseline for comparison.

To evaluate which method worked best the accuracy of each algorithm was compared after each algorithm had been applied to the preprocessed data. The algorithm with the best accuracy at 99% was discovered to be a random forest. With 84% and 96% accuracy respectively, logistic regression & KNN also fared well. With an accuracy rate of 85% the decision tree algorithm did less well.

A stacking approach was then used to combine the outputs of the four algorithms which involved training a meta-model on the outputs of the base models. This approach can help to improve, the overall accuracy of the model by combining the strengths of different algorithms.

Flask, a popular online framework for creating and delivering machine learning models was used, to deploy the model at the end. This made it possible for people to view and utilize the model via a web interface.

Finally, the project involved creating synthetic data from bank customer data, performing data analysis and pre-processing, implementing four different machine learning algorithms, comparing the accuracy of each algorithm, combining the algorithms using a stacking approach, and deploying the final model using Flask. The overall objective of the research was to anticipate customer attrition & the outcomes showed that this objective may be accomplished using a combination of machine learning techniques.

Bibliography

[1] J. &. B. L. L. Hadden, "Customer churn prediction using machine learning algorithms: A systematic review. Expert Systems with Applications," www.sciencedirect.com, 2017.

[2] S. &. K. V. Narasimhan, "Customer churn prediction and prevention: A systematic literature review. Expert Systems with Applications," www.sciencedirect.com, 2018.

[3] M. J. Zaki, "Synthetic data: A survey. ACM Computing Surveys (CSUR)," 2020.

[4] L. &. R. R. P. Torgo, "Synthetic data for fraud detection: A survey. ACM Computing Surveys (CSUR)," 2019.

[5] M. S. M. &. S. S. Kshirsagar, " Synthetic data generation using GANs for data augmentation: A survey. International Journal of Computer Applications," 2019.

[6] Altexsoft, "Customer Churn Prediction Using Machine Learning: Main Approaches and Models," 2019.

[7] W. e. al, "IEEE Transactions on Industrial Informatics,," 2021.

[8] S. &. M. Jain, "Customer churn prediction using synthetic data: An application of generative adversarial networks," 2020.

[9] L. a. Yoon, "Study," ieeexplore.ieee.org.

[10] T. Kimura, "Customer Churn Prediction with Hybrid Resampling and Ensemble Learning.," 2022.

[11] W. C. Z. C. Y. &. D. X. 2. Jiang, "Customer churn prediction in e-commerce with Gaussian mixture model and synthetic data. Journal of Ambient Intelligence and Humanized Computing," 2020.

[12] C. dilmegani, "Synthetic Data vs Real Data: Benefits, Challenges in 2023," 2022.

[13] K. (n.d.), "Bank Customer Churn Prediction," sakshi, 2021.

[14] hyperproof.io, "What Is Business Impact Analysis?," 2022.

[15] Maha, "Data Selection," 2021.

[16] itl.nist.gov, "Measures of Skewness and Kurtosis," 2019.

[17] ferdio, "Heat Map," 2022. [Online]. Available: https://datavizproject.com/data-type/heat-map/.

[18] plotly, "Heatmaps in Python," 2022.

[19] T. Bock, "displayr," [Online]. Available: https://www.displayr.com/what-is-a-correlation-matrix/.

[20] J. Brownlee, "How to Develop a Random Forest Ensemble in Python," 2020.

[21] J. Brownlee, "A Gentle Introduction to k-fold Cross-Validation," 2020.

[22] N. Bressler, "How to Check the Accuracy of Your Machine Learning Model," 2022.

[23] Y. Khandelwal, "Ensemble Stacking for Machine Learning and Deep Learning," 2021.

[24] Flask, "flask.palletsprojects," 2015. [Online]. Available: https://flask.palletsprojects.com/en/2.2.x/.