PROJECT ON NETFLIX DATA ANALYSIS

A minor project report submitted to

By

Ms. KANDRU BLESSY	**Y21CSE065**
Ms. KODURIKOUSALYA	**Y21CSE075**
Mr.KOTTAPALLIBRAHMAM	**Y21CSE085**
Ms.MALLAVARPUSUSMITHA	**Y21CSE095**
Mr. MEKALA SIVA	**Y21CSE105**
Ms. NALLURI KAVYA	**Y21CSE115**

Under the Esteemed Guidance of

Er. Y VIJAYA DURGA CHANDRA SEKHAR ***(Hons. In IT)***

Founder & Chief Executive Officer, CS CODENZ

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CHALAPATHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

(Approved by A.I.C.T.E, Affiliated To Acharya Nagarjuna University)

GUNTUR – 522 034

2023 - 2024

CHALAPATHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

(Approved by A.I.C.T.E, Affiliated To Acharya Nagarjuna University)

CHALAPATHI NAGAR, LAM, GUNTUR

CERTIFICATE

This is to certify that the Minor Project entitled as **"PROJECT ON NETFLIX DATA ANALYSIS"** submitted by **KANDRU BLESSY (Y21CSE065)** in partial fulfillment for the award of the Minor Project (Data Analytics Using Python For Machine Learning) is a record of bonafied work carried out under my guidance.

UNDER THE GUIDANCE OF

Er Y V D CHANDRA SEKHAR *Hons. In IT*
Founder & CEO , CS CODENZ

HEAD OF THE DEPARTMENT

Dr A Balaji *M.E , Ph.D*
Professor & HOD , CSE

CHALAPATHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

(Approved by A.I.C.T.E, Affiliated To Acharya Nagarjuna University)

CHALAPATHI NAGAR, LAM, GUNTUR

CERTIFICATE

This is to certify that the Minor Project entitled as **"PROJECT ON NETFLIX DATA ANALYSIS"** submitted by **KODURI KRISHNA KOUSALYA(Y21CSE075)** in partial fulfillment for the award of the Minor Project (Data Analytics Using Python For Machine Learning) is a record of bonafied work carried out under my guidance.

UNDER THE GUIDANCE OF

Er Y V D CHANDRA SEKHAR *Hons. In IT*
Founder & CEO , CS CODENZ

HEAD OF THE DEPARTMENT

Dr A Balaji *M.E , Ph.D*
Professor & HOD , CSE

CHALAPATHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

(Approved by A.I.C.T.E, Affiliated To Acharya Nagarjuna University)

CHALAPATHI NAGAR, LAM, GUNTUR

CERTIFICATE

This is to certify that the Minor Project entitled as **"PROJECT ON NETFLIX DATA ANALYSIS"** submitted by **KOTTAPALLI BRAHMAM (Y21CSE085)** in partial fulfillment for the award of the Minor Project (Data Analytics Using Python For Machine Learning) is a record of bonafied work carried out under my guidance.

UNDER THE GUIDANCE OF

Er Y V D CHANDRA SEKHAR *Hons. In IT*
Founder & CEO , CS CODENZ

HEAD OF THE DEPARTMENT

Dr A Balaji *M.E , Ph.D*
Professor & HOD , CSE

CHALAPATHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

(Approved by A.I.C.T.E, Affiliated To Acharya Nagarjuna University)

CHALAPATHI NAGAR, LAM, GUNTUR

CERTIFICATE

This is to certify that the Minor Project entitled as **"PROJECT ON NETFLIX DATA ANALYSIS"** submitted by **MALLAVARPU SUSMITHA (Y21CSE095)** in partial fulfillment for the award of the Minor Project (Data Analytics Using Python For Machine Learning) is a record of bonafied work carried out under my guidance.

UNDER THE GUIDANCE OF

Er Y V D CHANDRA SEKHAR *Hons. In IT*
Founder & CEO , CS CODENZ

HEAD OF THE DEPARTMENT

Dr A Balaji *M.E , Ph.D*
Professor & HOD , CSE

CHALAPATHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

(Approved by A.I.C.T.E, Affiliated To Acharya Nagarjuna University)

CHALAPATHI NAGAR, LAM, GUNTUR

CERTIFICATE

This is to certify that the Minor Project entitled as **"PROJECT ON NETFLIX DATA ANALYSIS"** submitted by **MEKALA SIVA (Y21CSE105)** in partial fulfillment for the award of the Minor Project (Data Analytics Using Python For Machine Learning) is a record of bonafied work carried out under my guidance.

UNDER THE GUIDANCE OF

Er Y V D CHANDRA SEKHAR *Hons. In IT*
Founder & CEO , CS CODENZ

HEAD OF THE DEPARTMENT

Dr A Balaji *M.E , Ph.D*
Professor & HOD , CSE

CHALAPATHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

(Approved by A.I.C.T.E, Affiliated To Acharya Nagarjuna University)

CHALAPATHI NAGAR, LAM, GUNTUR

CERTIFICATE

This is to certify that the Minor Project entitled as **"PROJECT ON NETFLIX DATA ANALYSIS"** submitted by **NALLURI KAVYA (Y21CSE115)** in partial fulfillment for the award of the Minor Project (Data Analytics Using Python For Machine Learning) is a record of bonafied work carried out under my guidance.

UNDER THE GUIDANCE OF

Er Y V D CHANDRA SEKHAR *Hons. In IT*
Founder & CEO , CS CODENZ

HEAD OF THE DEPARTMENT

Dr A Balaji *M.E , Ph.D*
Professor & HOD , CSE

DECLARATION

I **KANDRU BLESSY (Y21CSE065)** declared that the dissertation report entitled **" PROJECT ON NETFLIX DATA ANALYSIS"** is no more than 1,00,000 words in length including quotes and exclusive of tables, figures, bibliography, and references. This dissertation contains no material that has been submitted previously, In whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated this dissertation in our own work.

Roll No	**Name**	**Signature**
Y21CSE065	KANDRU BLESSY	

Date :

Place :

DECLARATION

I **KODURI KRISHNA KOUSALYA (Y21CSE075)** declared that the dissertation report entitled **" PROJECT ON NETFLIX DATA ANALYSIS"** is no more than 1,00,000 words in length including quotes and exclusive of tables, figures, bibliography, and references. This dissertation contains no material that has been submitted previously, In whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated this dissertation in our own work.

Roll No	**Name**	**Signature**
Y21CSE075	KODURI KRISHNA KOUSALYA	

Date :

Place :

DECLARATION

I **KOTTAPALLI BRAHMAM (Y21CSE085)** declared that the dissertation report entitled **" PROJECT ON NETFLIX DATA ANALYSIS"** is no more than 1,00,000 words in length including quotes and exclusive of tables, figures, bibliography, and references. This dissertation contains no material that has been submitted previously, In whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated this dissertation in our own work.

Roll No	**Name**	**Signature**
Y21CSE085	KOTTAPALLI BRAHMAM	

Date :

Place :

DECLARATION

I **MALLAVARPU SUSMITHA (Y21CSE095)** declared that the dissertation report entitled **"PROJECT ON NETFLIX DATA ANALYSIS"** is no more than 1,00,000 words in length including quotes and exclusive of tables, figures, bibliography, and references. This dissertation contains no material that has been submitted previously, In whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated this dissertation in our own work.

Roll No	**Name**	**Signature**
Y21CSE095	MALLAVARPU SUSMITHA	

Date :

Place :

DECLARATION

I **MEKALA SIVA (Y21CSE0105)** declared that the dissertation report entitled **" PROJECT ON NETFLIX DATA ANALYSIS"** is no more than 1,00,000 words in length including quotes and exclusive of tables, figures, bibliography, and references. This dissertation contains no material that has been submitted previously, In whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated this dissertation in our own work.

Roll No	**Name**	**Signature**
Y21CSE0105	MEKALA SIVA	

Date :

Place :

DECLARATION

I **NALLURI KAVYA (Y21CSE115)** declared that the dissertation report entitled **" PROJECT ON NETFLIX DATA ANALYSIS"** is no more than 1,00,000 words in length including quotes and exclusive of tables, figures, bibliography, and references. This dissertation contains no material that has been submitted previously, In whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated this dissertation in our own work.

Roll No	**Name**	**Signature**
Y21CSE115	NALLURI KAVYA	

Date :

Place :

ACKNOWLEDGMENT

We express our sincere thanks to our beloved Chairman sir , **Shri. Y V ANJANEYULU** for providing support and simulating environment for developing the project.

We express deep sense of reverence and profound gratitude to **Dr. M CHANDRA SEKHAR , Ph.D , Principal** for providing us the great support in carrying out the project.

It plunges us in exhilaration in taking privilege in expressing our heartfelt gratitude to **, Dr A Balaji , M.E , Ph.D ,** HOD - CSE for providing us every facility and for constant supervision.

We are thankful to our guide **Er Y Vijaya Durga Chandra Sekhar , Founder & CEO , CS CODENZ** for his encouragement, suggestions , supervision and abundant support throughout the project

Thanks to all the teaching and non-teaching staff and lab technicians for their support and also to our team mates for their valuable Co-operation.

Roll No	Name of the Student
Y21CSE065	KANDRU BLESSY
Y21CSE075	KODURI KRISHNAKOUSALYA
Y21CSE085	KOTTAPALLI BRAHMAM
Y21CSE095	MALLAVARPU SUSMITHA
Y21CSE105	MEKALA SIVA
Y21CSE115	NALLURI KAVYA

TABLE OF CONTENT

Abstract

Problem Statement

1. Introduction
2. Motivation & Objective
    - 2.1 Motivation
    - 2.2 Objective
3. Software and Hardware Requirements
    - 3.1 Software Requirements
    - 3.2 Hardware Requirements
4. Data Analysis
    - 4.1 Defining a Question ?
    - 4.2 Data Set Generation
    - 4.3 CRUD Operations
    - 4.4 Multi-dimensional Data Models
    - 4.5 Data Pre-Processing Techniques
    - 4.6 Apriori Algorithm Implementation
    - 4.7 Correlation Calculation
    - 4.8 Data Visualization
5. Result
6. Conclusion

ABSTRACT

This project delves into the world of Netflix through comprehensive data analysis. Leveraging a rich dataset comprising user interactions and content metadata, we employ advanced analytical techniques to uncover intricate patterns and insights. Through exploratory data analysis, we unveil viewer preferences, content popularity trends, binge-watching behaviors, and regional disparities in usage patterns. By applying statistical modeling and machine learning algorithms, we delve deeper into understanding factors influencing user engagement and content consumption. Additionally, we investigate the impact of variables such as genre, release year, and user demographics on content popularity and viewer retention. This project not only sheds light on the intricacies of Netflix usage but also provides actionable insights for content creators, marketers, and platform strategists to enhance user experience and optimize content offerings.

PROBLEM STATEMENT

The exponential growth of streaming platforms like Netflix, understanding user behavior and preferences has become paramount for content creators and platform strategists. This project aims to address the following key questions through data analysis:

1. Viewer Preferences: What are the most popular genres and types of content among Netflix users? Are there discernible patterns in viewing habits based on demographics or geographic regions?

2. Content Popularity: Which movies and TV shows are the most popular on Netflix? Can we identify factors such as release year, genre, or cast that contribute to a show's popularity?

3. Binge-watching Behaviours: How prevalent is binge-watching among Netflix users? Are there specific genres or types of content that are more likely to be binge-watched?

4. User Engagement: What factors influence user engagement on the platform? Are there certain features or promotional strategies that drive increased user activity?

5. Content Retention: How long do users typically engage with a particular show or movie on Netflix? Can we identify factors that contribute to viewer retention or drop-off?

CHAPTER 1

1. INTRODUCTION

The advent of streaming services has fundamentally transformed the entertainment industry, reshaping how audiences consume and interact with content. Among the vanguard of this digital revolution stands Netflix, an omnipresent force in the global media landscape. Boasting millions of subscribers worldwide and a sprawling library of movies, TV shows, documentaries, and original content, Netflix has not only redefined the concept of on-demand viewing but has also become synonymous with the modern era of entertainment consumption.

In this era of abundance, understanding the intricacies of viewer behaviour and preferences on Netflix has emerged as a critical endeavor for content creators, marketers, and platform strategists alike. The vast reservoir of user data generated by Netflix presents an unparalleled opportunity to gain insights into audience demographics, consumption patterns, and content preferences, thereby unlocking the keys to success in an increasingly competitive market.

Against this backdrop, this data analysis project embarks on a comprehensive exploration of the Netflix ecosystem, aiming to dissect and decode the multifaceted dynamics that underpin its success. By leveraging a rich dataset encompassing user interactions, viewing histories, and content metadata, we endeavor to unravel the complexities of Netflix usage, uncovering hidden trends, patterns, and correlations that illuminate the inner workings of the platform.

Central to our inquiry are a series of fundamental questions that lie at the intersection of data science and entertainment:

1. Viewer Preferences: What genres, themes, and formats resonate most strongly with Netflix audiences? How do viewer preferences vary across different demographics, such as age, gender, and geographic location?

2. Binge-Watching Behaviours: To what extent do Netflix users engage in binge-watching, and which types of content are most conducive to this behaviour? Are there discernible patterns in binge-watching habits across different demographic groups?

3. User Engagement: What factors drive user engagement and retention on the Netflix platform? How do features such as recommendation algorithms, user interfaces, and promotional strategies influence viewer behaviour?

4. Content Retention: How long do viewers typically engage with a particular title on Netflix, and what factors contribute to sustained interest or drop-off? Are there specific attributes of content that correlate with higher retention rates?

By interrogating these questions through the lens of data analysis, we seek not only to unravel the mysteries of Netflix but also to provide actionable insights and strategic recommendations for stakeholders across the entertainment ecosystem. Whether it be content creators seeking to tailor their productions to audience preferences, marketers devising targeted promotional campaigns, or platform strategists optimizing the user experience.

CHAPTER 2

2. MOTIVATION & OBJECTIVE

The motivation behind analyzing Netflix data stems from the desire to understand how people interact with the platform and what content they prefer. By examining user behaviour and preferences, we can uncover valuable insights that inform content creation, marketing strategies, and platform improvements. Our objective is to delve into patterns of engagement, content popularity, and viewing habits to optimize the Netflix experience for users. Ultimately, we aim to use data-driven insights to enhance the quality of content offerings and improve user satisfaction on the platform.

2.1 MOTIVATION

The motivation behind conducting a data analysis project on Netflix lies in its prominent position within the entertainment industry. With its widespread popularity and vast user base, Netflix offers a rich source of data that can provide valuable insights into consumer behaviour, content preferences, and market trends. By analyzing this data, researchers and stakeholders can gain a deeper understanding of how users interact with the platform, identify patterns and trends, and make informed decisions to enhance user experience and drive business growth.

2.2 OBJECTIVE

The objective of the Netflix data analysis project is to extract valuable insights from user interactions and content metadata to understand viewer preferences, content popularity, binge-watching behaviours, and factors influencing user engagement. By employing statistical modeling and machine learning algorithms, the project aims to identify patterns, trends, and correlations within the data. Ultimately, the goal is to provide actionable recommendations for content creators, marketers, and platform strategists to optimize content offerings, enhance user experience, and drive growth on the Netflix platform.

CHAPTER 3

3 SOFTWARE & HARDWARE REQURIMENTS

3.1 SOFTWARE REQURIMENTS

Operating System	**:** Windows
Programming Language	**:** Python
Modules Required	**:** Pandas , Matplotlib
Modules	**:** Create own Dataset and perform all Data pre-processing operations and visualize the data.
IDE's	**:** Python Google Colab & Spyder

3.2 HARDWARE REQURIMENTS

Processor	**:** 11th Gen Intel(R) core (TM) i5-1155G7@ 2.50GH
RAM	**:** 8.00GB
Version	**:** 22H2

CHAPTER 4

4 DATA ANALYSIS

4.1 Defining a Question ?

The exponential growth of streaming platforms like Netflix, understanding user behaviour and preferences has become paramount for content creators and platform strategists. This project aims to address the following key questions through data analysis:

1. Viewer Preferences: What are the most popular genres and types of content among Netflix users? Are there discernible patterns in viewing habits based on demographics or geographic regions?
2. Content Popularity: Which movies and TV shows are the most popular on Netflix? Can we identify factors such as release year, genre, or cast that contribute to a show's popularity?
3. Binge-watching Behaviours: How prevalent is binge-watching among Netflix users? Are there specific genres or types of content that are more likely to be binge-watched?
4. User Engagement: What factors influence user engagement on the platform? Are there certain features or promotional strategies that drive increased user activity?
5. Content Retention: How long do users typically engage with a particular show or movie on Netflix? Can we identify factors that contribute to viewer retention or drop-off?

4.2 Data Set Generation

4.2.1

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
a = {
   "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101]),
   "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF", "Radhe Syam", "Uppena"]),
   "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]),
   "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700]),
   "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140]),
   "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840]),
   "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110]),
}
b = pd.DataFrame(a)
print("DataFrame 'b' before modification:")
print()
print(b)
print()
print("\n---- The table is created ----\n")
```

```
DataFrame 'b' before modification:

   Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost
0       101      Bahubali          2013     900     180.0  800.0  200.0
1       102           RRR          2014     700     120.0  100.0  180.0
2       103     Wednesday          2015     970     110.0  120.0  120.0
3       104          Loki          2016     780     100.0    NaN  100.0
4       105      Dj Tillu          2017     800     120.0  200.0  200.0
5       106  The Avengers          2018     950       NaN  150.0  120.0
6       107  The end Game          2019     860     120.0  150.0    NaN
7       108           KGF          2020     980     120.0  120.0  300.0
8       109    Radhe Syam          2021    1000     110.0  600.0  100.0
9       101        Uppena          2022     700     140.0  840.0  110.0


---- The table is created ----
```

4.3 CRUD Operations

4.3.1 Create:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
a = {
   "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101]),
   "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF",   "Radhe Syam", "Uppena"]),
   "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]),
   "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700]),
   "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140]),
   "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840]),
   "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110]),
}
b = pd.DataFrame(a)
print("DataFrame 'b' before modification:")
print()
print(b)
print()
print("\n---- The table is created ----\n")
print()
b['Total'] = b['Duration'] + b['Views'] + b['Cost']
print("DataFrame 'b' after modification:")
print()
print(b)
print()
print("\n---- Total column added ----\n")
```

Before Creation:

```
DataFrame 'b' before modification:

   Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost
0       101      Bahubali          2013     900     180.0  800.0  200.0
1       102           RRR          2014     700     120.0  100.0  180.0
2       103     Wednesday          2015     970     110.0  120.0  120.0
3       104          Loki          2016     780     100.0    NaN  100.0
4       105      Dj Tillu          2017     800     120.0  200.0  200.0
5       106  The Avengers          2018     950       NaN  150.0  120.0
6       107  The end Game          2019     860     120.0  150.0    NaN
7       108           KGF          2020     980     120.0  120.0  300.0
8       109    Radhe Syam          2021    1000     110.0  600.0  100.0
9       101        Uppena          2022     700     140.0  840.0  110.0
```

After Creation:

```
DataFrame 'b' after modification:

   Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost   Total
0       101      Bahubali          2013     900     180.0  800.0  200.0  1180.0
1       102           RRR          2014     700     120.0  100.0  180.0   400.0
2       103     Wednesday          2015     970     110.0  120.0  120.0   350.0
3       104          Loki          2016     780     100.0    NaN  100.0     NaN
4       105      Dj Tillu          2017     800     120.0  200.0  200.0   520.0
5       106  The Avengers          2018     950       NaN  150.0  120.0     NaN
6       107  The end Game          2019     860     120.0  150.0    NaN     NaN
7       108           KGF          2020     980     120.0  120.0  300.0   540.0
8       109    Radhe Syam          2021    1000     110.0  600.0  100.0   810.0
9       101        Uppena          2022     700     140.0  840.0  110.0  1090.0
```

4.3.2 Read

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
a = {
   "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101]),
   "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF", "Radhe Syam", "Uppena"]),
   "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]),
   "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700]),
   "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140]),
   "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840]),
   "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110]),
}
b = pd.DataFrame(a)
print("DataFrame 'b' before modification:")
print()
```

```
print(b)
print()
print("\n---- The table is created ----\n")
print()
b['Total'] = b['Duration'] + b['Views'] + b['Cost']
print("DataFrame 'b' after modification:")
print()
print(b)
print()
print("\n---- Total column added ----\n")
print("Accessing the 'Title_id' column:")
print()
print(b['Title_id'])
print()
print("\n---- 'Title_id' column displayed ----\n")
print("Accessing the entire DataFrame:")
print()
print(b)
print()
print("\n---- Entire DataFrame displayed ----\n")
print("Accessing row at index 2:")
print()
print(b.loc[2])
print()
print("\n---- Row at index 2 displayed ----\n")
print("Accessing value at row 2, column 3:")
print()
print(b.loc[2][3])
print()
print("\n---- Value at row 2, column 3 displayed ----\n")
```

```
Accessing the 'Title_id' column:

0    101
1    102
2    103
3    104
4    105
5    106
6    107
7    108
8    109
9    101
Name: Title_id, dtype: int64


---- 'Title_id' column displayed ----

Accessing row at index 2:

Title_id              103
Movie_name      Wednesday
Release_year         2015
Rating                970
Duration            110.0
Views               120.0
Cost                120.0
Total               350.0
Name: 2, dtype: object


---- Row at index 2 displayed ----
```

4.3.3 Update:

```
  import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
a = {
   "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101]),
   "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF", "Radhe Syam", "Uppena"]),
   "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]),
   "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700]),
   "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140]),
   "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840]),
   "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110]),
}
b = pd.DataFrame(a)
print("DataFrame 'b' before modification:")
print()
print(b)
```

```
print()
print("\n---- The table is created ----\n")
print()
b['Total'] = b['Duration'] + b['Views'] + b['Cost']
print("DataFrame 'b' after modification:")
print()
print(b)
print()
print("\n---- Total column added ----\n")
print("Accessing the 'Title_id' column:")
print()
print(b['Title_id'])
print()
print("\n---- 'Title_id' column displayed ----\n")
print("Accessing the entire DataFrame:")
print()
print(b)
print()
print("\n---- Entire DataFrame displayed ----\n")
print("Accessing row at index 2:")
print()
print(b.loc[2])
print()
print("\n---- Row at index 2 displayed ----\n")
print("Accessing value at row 2, column 3:")
print()
print(b.loc[2][3])
print()
print("\n---- Value at row 2, column 3 displayed ----\n")
print("Column update - updating 'Views' column at index 0:")
print()
b['Views'][0] = 180
print()
print(b)
print()
print("\n---- 'Views' column updated ----\n")
print("Row update - updating 'Movie_name' for index 0:")
print()
b.loc[0, "Movie_name"] = "Racha"
print()
print(b)
print()
print("\n---- 'Movie_name' updated for index 0 ----\n")
```

OUTPUT:

Before Updation:

```
DataFrame 'b' after modification:

   Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost   Total
0       101      Bahubali          2013     900     180.0  800.0  200.0  1180.0
1       102           RRR          2014     700     120.0  100.0  180.0   400.0
2       103     Wednesday          2015     970     110.0  120.0  120.0   350.0
3       104          Loki          2016     780     100.0    NaN  100.0     NaN
4       105      Dj Tillu          2017     800     120.0  200.0  200.0   520.0
5       106  The Avengers          2018     950       NaN  150.0  120.0     NaN
6       107  The end Game          2019     860     120.0  150.0    NaN     NaN
7       108           KGF          2020     980     120.0  120.0  300.0   540.0
8       109    Radhe Syam          2021    1000     110.0  600.0  100.0   810.0
9       101        Uppena          2022     700     140.0  840.0  110.0  1090.0
```

After Column Updation:

```
Column update - updating 'Views' column at index 0:


   Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost   Total
0       101      Bahubali          2013     900     180.0  180.0  200.0  1180.0
1       102           RRR          2014     700     120.0  100.0  180.0   400.0
2       103     Wednesday          2015     970     110.0  120.0  120.0   350.0
3       104          Loki          2016     780     100.0    NaN  100.0     NaN
4       105      Dj Tillu          2017     800     120.0  200.0  200.0   520.0
5       106  The Avengers          2018     950       NaN  150.0  120.0     NaN
6       107  The end Game          2019     860     120.0  150.0    NaN     NaN
7       108           KGF          2020     980     120.0  120.0  300.0   540.0
8       109    Radhe Syam          2021    1000     110.0  600.0  100.0   810.0
9       101        Uppena          2022     700     140.0  840.0  110.0  1090.0


---- 'Views' column updated ----
```

After Row Updation :

```
Row update - updating 'Movie_name' for index 0:


   Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost   Total
0       101         Racha          2013     900     180.0  180.0  200.0  1180.0
1       102           RRR          2014     700     120.0  100.0  180.0   400.0
2       103     Wednesday          2015     970     110.0  120.0  120.0   350.0
3       104          Loki          2016     780     100.0    NaN  100.0     NaN
4       105      Dj Tillu          2017     800     120.0  200.0  200.0   520.0
5       106  The Avengers          2018     950       NaN  150.0  120.0     NaN
6       107  The end Game          2019     860     120.0  150.0    NaN     NaN
7       108           KGF          2020     980     120.0  120.0  300.0   540.0
8       109    Radhe Syam          2021    1000     110.0  600.0  100.0   810.0
9       101        Uppena          2022     700     140.0  840.0  110.0  1090.0


---- 'Movie_name' updated for index 0 ----
```

4.3.4 Delete:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
a = {
   "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101]),
   "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF", "Radhe Syam", "Uppena"]),
   "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]),
   "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700]),
   "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140]),
   "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840]),
   "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110]),
}
b = pd.DataFrame(a)
print("DataFrame 'b' before modification:")
print()
print(b)
print()
print("\n---- The table is created ----\n")
print()
b['Total'] = b['Duration'] + b['Views'] + b['Cost']
print("DataFrame 'b' after modification:")
print()
print(b)
print()
print("\n---- Total column added ----\n")
print("Accessing the 'Title_id' column:")
print()
print(b['Title_id'])
print()
```

```
print("\n---- 'Title_id' column displayed ----\n")
print("Accessing the entire DataFrame:")
print()
print(b)
print()
print("\n---- Entire DataFrame displayed ----\n")
print("Accessing row at index 2:")
print()
print(b.loc[2])
print()
print("\n---- Row at index 2 displayed ----\n")
print("Accessing value at row 2, column 3:")
print()
print(b.loc[2][3])
print()
print("\n---- Value at row 2, column 3 displayed ----\n")
print("Column update - updating 'Views' column at index 0:")
print()
b['Views'][0] = 180
print()
print(b)
print()
print("\n---- 'Views' column updated ----\n")
print("Row update - updating 'Movie_name' for index 0:")
print()
b.loc[0, "Movie_name"] = "Racha"
print()
print(b)
print()
print("\n---- 'Movie_name' updated for index 0 ----\n")
print("Column deletion - deleting 'Total' column:")
print()
b.pop('Total')
print()
print(b)
print()
print("\n---- 'Total' column deleted ----\n")
print("Row deletion - deleting row with index 9:")
print()
x = b.drop(9)
print()
print(x)
print()
print("\n---- Row with index 9 deleted ----\n")
```

OUTPUT:

Before Delete:

	Title_id	Movie_name	Release_year	Rating	Duration	Views	Cost	Total
0	101	Racha	2013	900	180.0	180.0	200.0	1180.0
1	102	RRR	2014	700	120.0	100.0	180.0	400.0
2	103	Wednesday	2015	970	110.0	120.0	120.0	350.0
3	104	Loki	2016	780	100.0	NaN	100.0	NaN
4	105	Dj Tillu	2017	800	120.0	200.0	200.0	520.0
5	106	The Avengers	2018	950	NaN	150.0	120.0	NaN
6	107	The end Game	2019	860	120.0	150.0	NaN	NaN
7	108	KGF	2020	980	120.0	120.0	300.0	540.0
8	109	Radhe Syam	2021	1000	110.0	600.0	100.0	810.0
9	101	Uppena	2022	700	140.0	840.0	110.0	1090.0

After Column Deletion:

```
Column deletion - deleting 'Total' column:
```

	Title_id	Movie_name	Release_year	Rating	Duration	Views	Cost
0	101	Racha	2013	900	180.0	180.0	200.0
1	102	RRR	2014	700	120.0	100.0	180.0
2	103	Wednesday	2015	970	110.0	120.0	120.0
3	104	Loki	2016	780	100.0	NaN	100.0
4	105	Dj Tillu	2017	800	120.0	200.0	200.0
5	106	The Avengers	2018	950	NaN	150.0	120.0
6	107	The end Game	2019	860	120.0	150.0	NaN
7	108	KGF	2020	980	120.0	120.0	300.0
8	109	Radhe Syam	2021	1000	110.0	600.0	100.0
9	101	Uppena	2022	700	140.0	840.0	110.0

```
---- 'Total' column deleted ----
```

After Row Deletion:

```
Row deletion - deleting row with index 9:


   Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost
0       101         Racha          2013     900     180.0  180.0  200.0
1       102           RRR          2014     700     120.0  100.0  180.0
2       103     Wednesday          2015     970     110.0  120.0  120.0
3       104          Loki          2016     780     100.0    NaN  100.0
4       105      Dj Tillu          2017     800     120.0  200.0  200.0
5       106  The Avengers          2018     950       NaN  150.0  120.0
6       107  The end Game          2019     860     120.0  150.0    NaN
7       108           KGF          2020     980     120.0  120.0  300.0
8       109    Radhe Syam          2021    1000     110.0  600.0  100.0


---- Row with index 9 deleted ----
```

4.5 Data Pre-Processing Techniques:

4.5.1 Data Collection:

```
 import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
a = {
   "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101]),
   "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF", "Radhe Syam", "Uppena"]),
   "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]),
   "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700]),
   "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140]),
   "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840]),
   "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110]),
}
b = pd.DataFrame(a)
print()
print("------DATA PRE-PROCESSING TECHNIQES------")
print()
print("------------DATA COLLECTION--------")
print()
print(b)
```

OUTPUT:

```
------DATA PRE-PROCESSING TECHNIQES------

--------DATA COLLECTION--------

   Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost
0       101      Bahubali          2013     900     180.0  800.0  200.0
1       102           RRR          2014     700     120.0  100.0  180.0
2       103     Wednesday          2015     970     110.0  120.0  120.0
3       104          Loki          2016     780     100.0    NaN  100.0
4       105      Dj Tillu          2017     800     120.0  200.0  200.0
5       106  The Avengers          2018     950       NaN  150.0  120.0
6       107  The end Game          2019     860     120.0  150.0    NaN
7       108           KGF          2020     980     120.0  120.0  300.0
8       109    Radhe Syam          2021    1000     110.0  600.0  100.0
9       101        Uppena          2022     700     140.0  840.0  110.0
```

4.5.2 Data Cleaning:

```
  import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
a = {
    "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101]),
    "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF", "Radhe Syam", "Uppena"]),
    "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]),
    "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700]),
    "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140]),
    "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840]),
    "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110]),
}
b = pd.DataFrame(a)
print()
print("--------DATA CLEANING--------")
print()
print(b)
print()
print("Data Cleaning:")
print()
print()
print("\n ---- Data cleaning displayed ----\n")
print("Check for missing values using 'isnull()':")
```

```
print()
missing_values = b.isnull()
print(missing_values)
print()
print("\n---- Missing values checked ----\n")
print("Check for non-missing values using 'notnull()':")
print()
non_missing_values = b.notnull()
print(non_missing_values)
print()
print("\n---- Non-missing values checked ----\n")
```

OUTPUT:

```
Check for missing values using 'isnull()':

   Title_id  Movie_name  Release_year  Rating  Duration  Views   Cost
0     False       False         False   False     False  False  False
1     False       False         False   False     False  False  False
2     False       False         False   False     False  False  False
3     False       False         False   False     False   True  False
4     False       False         False   False     False  False  False
5     False       False         False   False      True  False  False
6     False       False         False   False     False  False   True
7     False       False         False   False     False  False  False
8     False       False         False   False     False  False  False
9     False       False         False   False     False  False  False


---- Missing values checked ----

Check for non-missing values using 'notnull()':

   Title_id  Movie_name  Release_year  Rating  Duration  Views   Cost
0      True        True          True    True      True   True   True
1      True        True          True    True      True   True   True
2      True        True          True    True      True   True   True
3      True        True          True    True      True  False   True
4      True        True          True    True      True   True   True
5      True        True          True    True     False   True   True
6      True        True          True    True      True   True  False
7      True        True          True    True      True   True   True
8      True        True          True    True      True   True   True
9      True        True          True    True      True   True   True


---- Non-missing values checked ----
```

4.5.3 Data Integration

```
 import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Data
a = {
    "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101]),
    "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF", "Radhe Syam", "Uppena"]),
    "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022]),
    "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700]),
    "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140]),
```

```
    "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840]),
    "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110]),
}

b = pd.DataFrame(a)

print()
print("--------DATA INTEGRATION--------")
print()
print(b)
print()
print("Data Integration - filling missing values using backward fill:")
print()
filled_backwards = b.fillna(method='bfill')
print()
print(filled_backwards)
print()
print("\n---- Missing values filled using backward fill ----\n")
print()
print("Data Integration - filling missing values using forward fill:")
print()
filled_forwards = b.fillna(method='pad')
print()
print(filled_forwards)
print()
```

OUTPUT:

```
--------DATA INTEGRATION--------

   Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost
0       101      Bahubali          2013     900     180.0  800.0  200.0
1       102           RRR          2014     700     120.0  100.0  180.0
2       103     Wednesday          2015     970     110.0  120.0  120.0
3       104          Loki          2016     780     100.0    NaN  100.0
4       105      Dj Tillu          2017     800     120.0  200.0  200.0
5       106  The Avengers          2018     950       NaN  150.0  120.0
6       107  The end Game          2019     860     120.0  150.0    NaN
7       108           KGF          2020     980     120.0  120.0  300.0
8       109    Radhe Syam          2021    1000     110.0  600.0  100.0
9       101        Uppena          2022     700     140.0  840.0  110.0

Data Integration - filling missing values using backward fill:


   Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost
0       101      Bahubali          2013     900     180.0  800.0  200.0
1       102           RRR          2014     700     120.0  100.0  180.0
2       103     Wednesday          2015     970     110.0  120.0  120.0
3       104          Loki          2016     780     100.0  200.0  100.0
4       105      Dj Tillu          2017     800     120.0  200.0  200.0
5       106  The Avengers          2018     950     120.0  150.0  120.0
6       107  The end Game          2019     860     120.0  150.0  300.0
7       108           KGF          2020     980     120.0  120.0  300.0
8       109    Radhe Syam          2021    1000     110.0  600.0  100.0
9       101        Uppena          2022     700     140.0  840.0  110.0


---- Missing values filled using backward fill ----
```

sing values filled using forward fill ----\n")

```
Data Integration - filling missing values using forward fill:


   Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost
0       101      Bahubali          2013     900     180.0  800.0  200.0
1       102           RRR          2014     700     120.0  100.0  180.0
2       103     Wednesday          2015     970     110.0  120.0  120.0
3       104          Loki          2016     780     100.0  120.0  100.0
4       105      Dj Tillu          2017     800     120.0  200.0  200.0
5       106  The Avengers          2018     950     120.0  150.0  120.0
6       107  The end Game          2019     860     120.0  150.0  120.0
7       108           KGF          2020     980     120.0  120.0  300.0
8       109    Radhe Syam          2021    1000     110.0  600.0  100.0
9       101        Uppena          2022     700     140.0  840.0  110.0


---- Missing values filled using forward fill ----
```

4.5.4 Data Reduction:

```
import pandas as pd
import numpy as np
a = {
    "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101,101]),
    "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF", "Radhe Syam", "Uppena","Bahubali"]),
    "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,2022, 2013]),
    "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700, 900]),
    "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140, 180]),
    "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840, 800]),
    "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110, 200]),
}

b = pd.DataFrame(a)

print(b)
print()
print("Drop duplicates:")
print()
dropped_duplicates = b.drop_duplicates()
print()
print(dropped_duplicates)
print()
print("\n---- Duplicates dropped ----\n")
```

Output:

```
    Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost
0        101      Bahubali          2013     900     180.0  800.0  200.0
1        102           RRR          2014     700     120.0  100.0  180.0
2        103     Wednesday          2015     970     110.0  120.0  120.0
3        104          Loki          2016     780     100.0    NaN  100.0
4        105      Dj Tillu          2017     800     120.0  200.0  200.0
5        106  The Avengers          2018     950       NaN  150.0  120.0
6        107  The end Game          2019     860     120.0  150.0    NaN
7        108           KGF          2020     980     120.0  120.0  300.0
8        109    Radhe Syam          2021    1000     110.0  600.0  100.0
9        101        Uppena          2022     700     140.0  840.0  110.0
10       101      Bahubali          2013     900     180.0  800.0  200.0

Drop duplicates:


   Title_id    Movie_name  Release_year  Rating  Duration  Views   Cost
0       101      Bahubali          2013     900     180.0  800.0  200.0
1       102           RRR          2014     700     120.0  100.0  180.0
2       103     Wednesday          2015     970     110.0  120.0  120.0
3       104          Loki          2016     780     100.0    NaN  100.0
4       105      Dj Tillu          2017     800     120.0  200.0  200.0
5       106  The Avengers          2018     950       NaN  150.0  120.0
6       107  The end Game          2019     860     120.0  150.0    NaN
7       108           KGF          2020     980     120.0  120.0  300.0
8       109    Radhe Syam          2021    1000     110.0  600.0  100.0
9       101        Uppena          2022     700     140.0  840.0  110.0


---- Duplicates dropped ----
```

4.5.5 Data Transformation:

It is the process of converting, cleansing, and structuring data into a usable format that can be analyzed to support decision making processes.

4.5.6 Data Discretization:

It is the process of putting values into buckets so that there are a limited number of possible status.

4.7 Correlation Calculation:

```
 import pandas as pd
import numpy as np
a = {
   "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101,101]),
   "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF", "Radhe Syam", "Uppena","Bahubali"]),
   "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,2022, 2013]),
   "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700, 900]),
   "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140, 180]),
   "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840, 800]),
   "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110, 200]),
}

b = pd.DataFrame(a)

print(b)
print()
```

```
print("Correlation between 'Duration' and 'Views':")
print(dropped_duplicates['Duration'].corr(dropped_duplicates['Views']))
print("\nCorrelation between 'Views' and 'Cost':")
print(dropped_duplicates['Duration'].corr(dropped_duplicates['Cost']))
print("Correlation between 'Rating' and 'Duration':")
print(dropped_duplicates['Rating'].corr(dropped_duplicates['Duration']))
```

OUTPUT:

```
Correlation between 'Duration' and 'Views':
0.06882660575887038


Correlation between 'Views' and 'Cost':
0.28047390110304443

Correlation between 'Rating' and 'Duration':
-0.06742632793700229
```

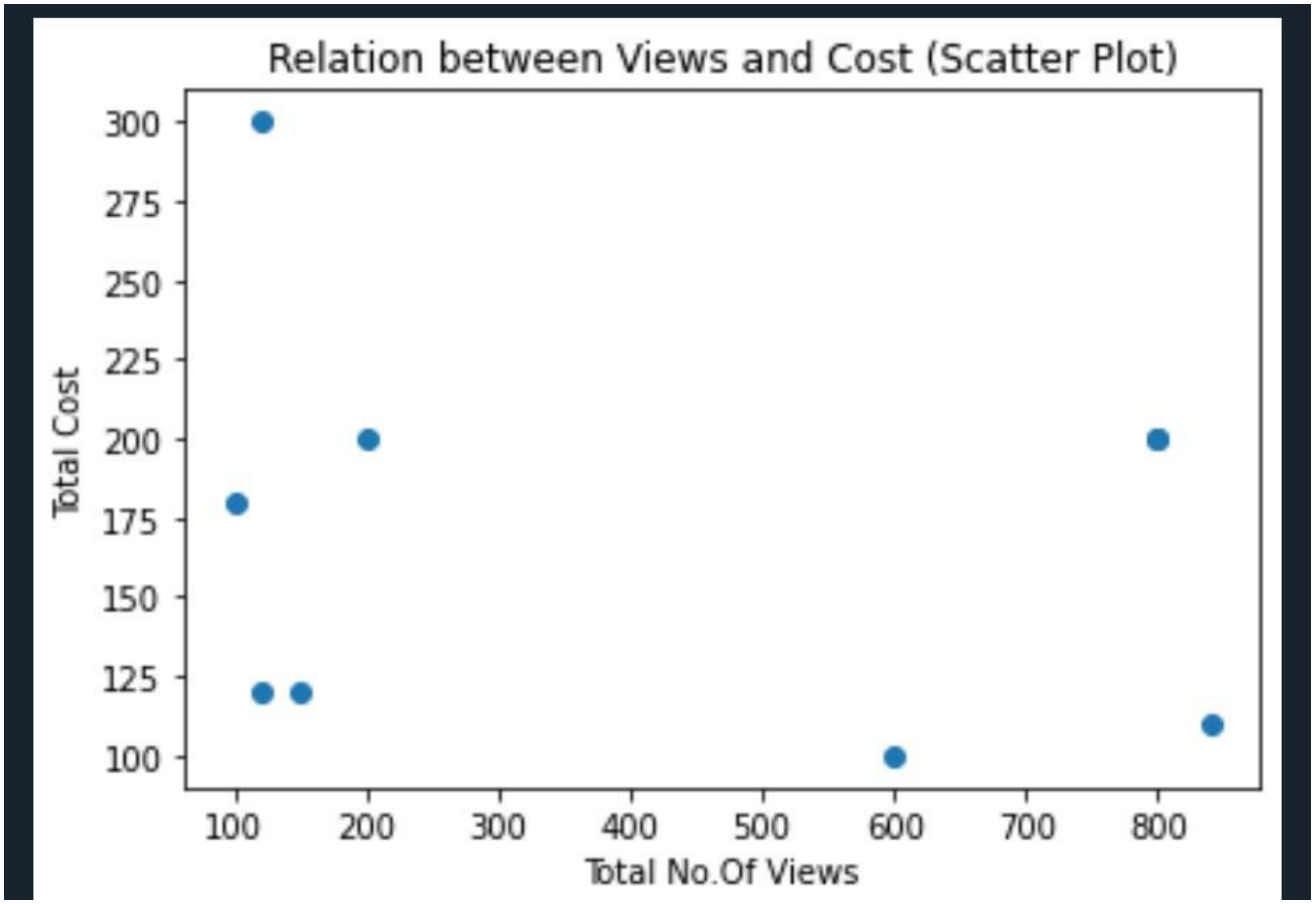
4.8 Data Visualization:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
a = {
    "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101,101]),
    "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF", "Radhe Syam", "Uppena","Bahubali"]),
    "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,2022, 2013]),
    "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700, 900]),
    "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140, 180]),
    "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840, 800]),
    "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110, 200]),
}
b = pd.DataFrame(a)

print(b)
plt.scatter(b["Views"], b["Cost"])
plt.xlabel("Total No.Of Views")
plt.ylabel("Total Cost")
plt.title("Relation between Views and Cost (Scatter Plot)")
plt.show()
```

OUTPUT:

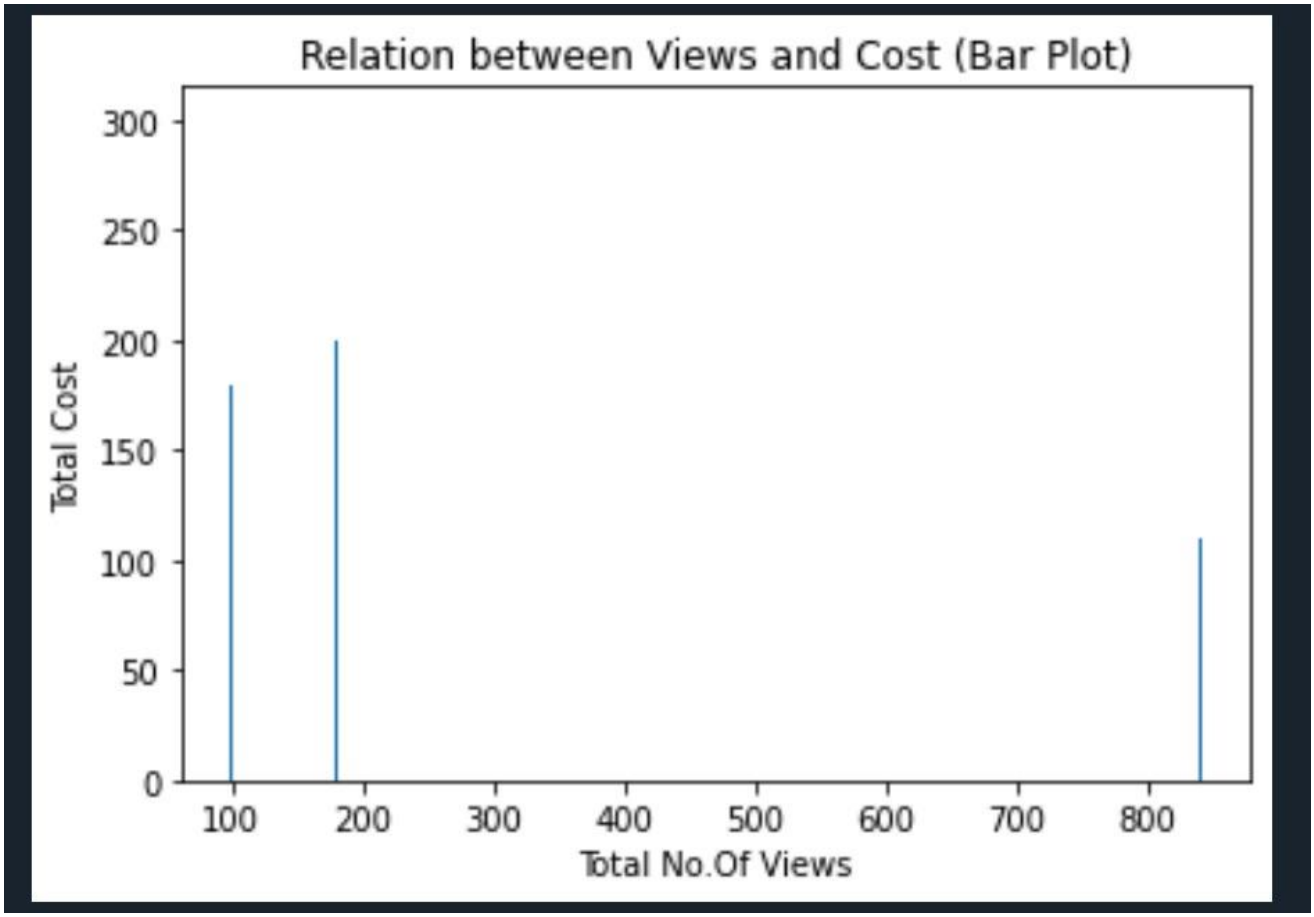
	Title_id	Movie_name	Release_year	Rating	Duration	Views	Cost
0	101	Bahubali	2013	900	180.0	800.0	200.0
1	102	RRR	2014	700	120.0	100.0	180.0
2	103	Wednesday	2015	970	110.0	120.0	120.0
3	104	Loki	2016	780	100.0	NaN	100.0
4	105	Dj Tillu	2017	800	120.0	200.0	200.0
5	106	The Avengers	2018	950	NaN	150.0	120.0
6	107	The end Game	2019	860	120.0	150.0	NaN
7	108	KGF	2020	980	120.0	120.0	300.0
8	109	Radhe Syam	2021	1000	110.0	600.0	100.0
9	101	Uppena	2022	700	140.0	840.0	110.0
10	101	Bahubali	2013	900	180.0	800.0	200.0



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
a = {
   "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101,101]),
   "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF", "Radhe Syam", "Uppena","Bahubali"]),
   "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,2022, 2013]),
   "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700, 900]),
   "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140, 180]),
   "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840, 800]),
   "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110, 200]),
}
b = pd.DataFrame(a)
print(b)
plt.bar(b["Views"], b["Cost"])
plt.xlabel("Total No.Of Views")
plt.ylabel("Total Cost")
plt.title("Relation between Views and Cost (Bar Plot)")
plt.show()
```

OUTPUT:

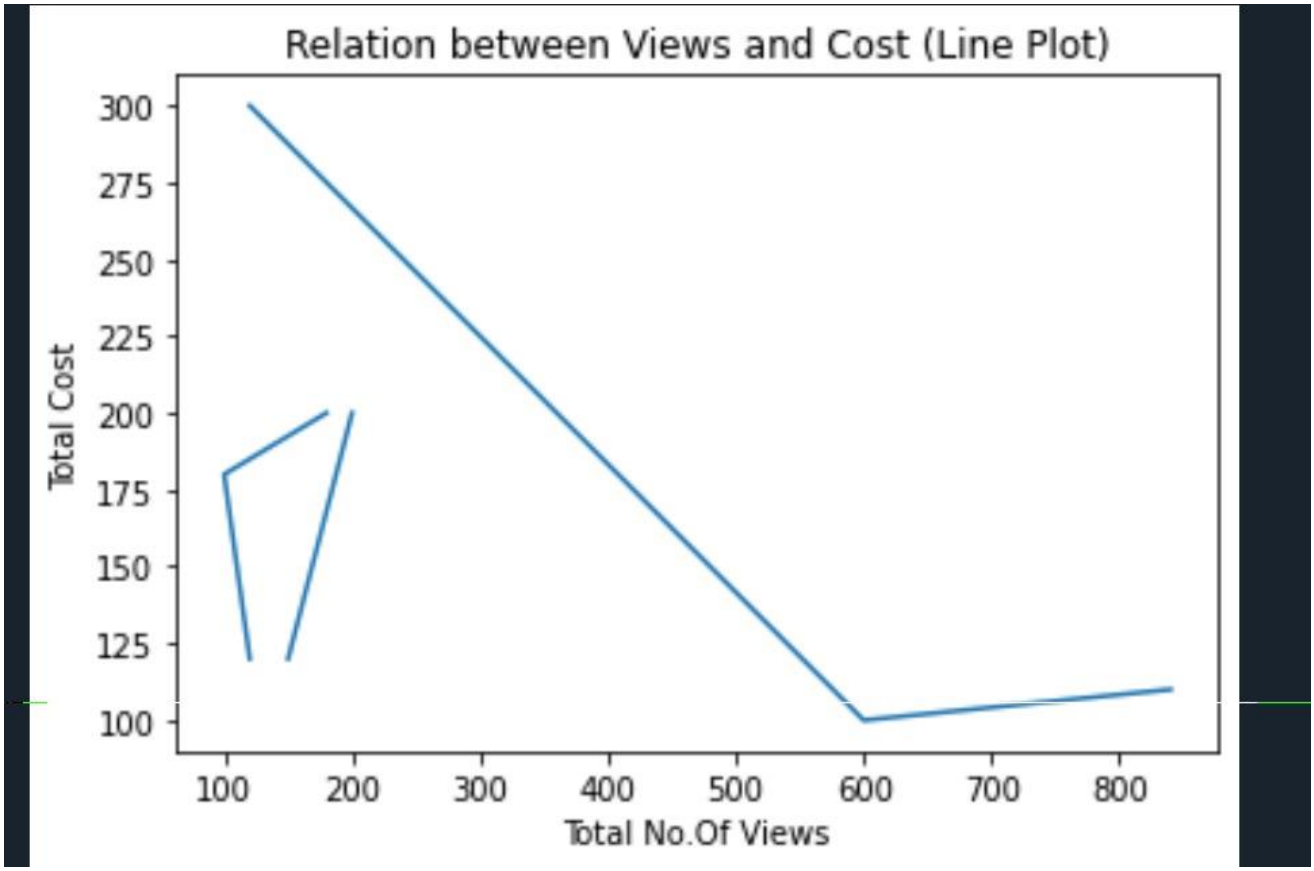
	Title_id	Movie_name	Release_year	Rating	Duration	Views	Cost
0	101	Bahubali	2013	900	180.0	800.0	200.0
1	102	RRR	2014	700	120.0	100.0	180.0
2	103	Wednesday	2015	970	110.0	120.0	120.0
3	104	Loki	2016	780	100.0	NaN	100.0
4	105	Dj Tillu	2017	800	120.0	200.0	200.0
5	106	The Avengers	2018	950	NaN	150.0	120.0
6	107	The end Game	2019	860	120.0	150.0	NaN
7	108	KGF	2020	980	120.0	120.0	300.0
8	109	Radhe Syam	2021	1000	110.0	600.0	100.0
9	101	Uppena	2022	700	140.0	840.0	110.0
10	101	Bahubali	2013	900	180.0	800.0	200.0


Relation between Views and Cost (Bar Plot)
Total Cost
300
250
200
150
100
50
0
100
200
300
400
500
600
700
800
Total No.Of Views

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
a = {
   "Title_id": pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 101,101]),
   "Movie_name": pd.Series(["Bahubali", "RRR", "Wednesday", "Loki", "Dj Tillu", "The Avengers",
"The end Game", "KGF", "Radhe Syam", "Uppena","Bahubali"]),
   "Release_year": pd.Series([2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,2022, 2013]),
   "Rating": pd.Series([900, 700, 970, 780, 800, 950, 860, 980, 1000, 700, 900]),
   "Duration": pd.Series([180, 120, 110, 100, 120, np.nan, 120, 120, 110, 140, 180]),
   "Views": pd.Series([800, 100, 120, np.nan, 200, 150, 150, 120, 600, 840, 800]),
   "Cost": pd.Series([200, 180, 120, 100, 200, 120, np.nan, 300, 100, 110, 200]),
}
b = pd.DataFrame(a)
print(b)
plt.plot(b["Views"], b["Cost"])
plt.xlabel("Total No.Of Views")
plt.ylabel("Total Cost")
plt.title("Relation between Views and Cost (Line Plot)")
plt.show()
```

OUTPUT:

	Title_id	Movie_name	Release_year	Rating	Duration	Views	Cost
0	101	Bahubali	2013	900	180.0	800.0	200.0
1	102	RRR	2014	700	120.0	100.0	180.0
2	103	Wednesday	2015	970	110.0	120.0	120.0
3	104	Loki	2016	780	100.0	NaN	100.0
4	105	Dj Tillu	2017	800	120.0	200.0	200.0
5	106	The Avengers	2018	950	NaN	150.0	120.0
6	107	The end Game	2019	860	120.0	150.0	NaN
7	108	KGF	2020	980	120.0	120.0	300.0
8	109	Radhe Syam	2021	1000	110.0	600.0	100.0
9	101	Uppena	2022	700	140.0	840.0	110.0
10	101	Bahubali	2013	900	180.0	800.0	200.0


Relation between Views and Cost (Line Plot)
Total Cost
300
275
250
225
200
175
150
125
100
100
200
300
400
500
600
700
800
Total No.Of Views

CHAPTER 5

5 RESULT

At the last after performing all the operations on the data set we had got the results.And we has also show the results by using three types of graphs and the graphs are shown above

CHAPTER 6

6 CONCLUSION

The conclusion of the project on Netflix Data Analysis is to how to indentify the customers who are willing to terminate their accounts from the banks.