ANP-D0449

DATA ANALYTICS USING PYTHON

SUBSCRIPTION BASED STREAMING SERVICE USAGE ANALYSIS

SUBMITTED BY:

R.VijayaKrishnan

S.V.Rajalakshmi

Abstract:

In today's digital age, streaming services like Netflix, Amazon Prime, Disney+ have transformed the way people consume entertainment. With millions of users worldwide, understanding their watching behavior, subscription type, and spending habits is crucial for optimizing user experience and business growth.

This analysis focuses on evaluating key factors such as watching hours, monthly spending, preferred content types, and subscription plans. By leveraging data analytics and visualization techniques, we can uncover patterns in user engagement, identify customer segments, and predict future subscription trends.

The insights from this study will help streaming platforms improve content recommendations, optimize pricing strategies, and enhance user retention. Additionally, the analysis can assist in reducing customer churn and increasing revenue by offering personalized experiences.

Problem Statement: Streaming platforms struggle with user retention, content optimization, and pricing strategies, leading to high churn rates and declining engagement. The challenge is to analyze user behavior and preferences to improve content recommendations, pricing models, and overall user experience.

Libraries used:

Pandas (import pandas as pd)

Pandas is a powerful data analysis library designed for handling and manipulating structured data, such as tables and spreadsheets. It provides efficient tools for data cleaning, transformation, and aggregation.

Key Features:

1.Works with DataFrames (tabular data) and Series (1D data).
2. Provides functions for data cleaning, handling missing values, and filtering.
3.Supports grouping and aggregation for summarizing data.
4.Allows merging and joining multiple datasets.
5.Reads and writes data in various formats like CSV, Excel, and JSON.

Matplotlib (import matplotlib.pyplot as plt)

Matplotlib is a basic visualization library used to create static, animated, and interactive plots. It provides extensive control over graph elements, making it highly customizable for various types of charts.

Key Features:

1.Supports line charts, bar plots, scatter plots, histograms, and more.
2. Allows full customization of titles, labels, legends, and colors.
3.Enables the creation of subplots for multiple charts in one figure.
4.Can export charts in multiple formats, including PNG, JPG, and PDF.

Seaborn (import seaborn as sns)

Seaborn is a statistical data visualization library built on top of Matplotlib. It provides elegant and easy-to-use functions for creating visually appealing and informative charts.

Key Features:

1.Offers built-in themes and color palettes for aesthetic charts.
2. Supports statistical plots like heatmaps, boxplots, violin plots, and regression plots.
3.Integrates seamlessly with Pandas DataFrames for efficient visualization.
4.Automatically handles data aggregation and categorical plotting.

Objective:

This analysis aims to understand user behaviour in subscription-based streaming services to enhance engagement, reduce churn, and optimize pricing. By examining watching patterns, content preferences, and spending habits, we can predict churn and implement retention strategies like personalized recommendations and targeted offers. Additionally, optimizing pricing models and improving content discovery will enhance user satisfaction. Comparing market trends and competitor strategies will provide insights for staying competitive, maximizing revenue, and delivering a more personalized streaming experience.

Code:

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

1. #Reading the CSV file

```
data=pd.read_csv("NetflixUsers.csv")
```

2.#Reading and viewing the dataset

```
data.head(5)
```

3.#Displaying columns

```
data.columns
```

4.#Finding null values

```
data.isnull().sum()
```

5.#Finding duplicate values

```
data.duplicated().sum()
```

6.#Deriving statistical information

```
data.describe()
```

7.#Deriving statistical information of object columns

```
data.describe(include='object')
```

8.#Pivot tables to find the monthly revenue usage based on age

```
pivot_table = df.pivot_table(values='Monthly Revenue', index='Age', columns='Subscription Type',
aggfunc='mean')
# Print Pivot Table
print(pivot_table)
```

9.#Visualizing the data using plots

```
age=data['Age']
transamt=data['Transaction_Amount']

plt.scatter(age,transamt,color='m')
plt.xlabel('Age')
plt.ylabel('Transaction_Amount')
plt.title('Transactions based on Age')
plt.show()
```

10.#Pivot tables to find the credit card usage based on Gender

```
table2=pd.pivot_table(data,values='Transaction_Amount',index='Gender',aggfunc=np.sum)
table2
```

11.#Visualizing of data using piechart(Subscription Type)

```
counts = data['Subscription Type'].value_counts()
myexplode=[0.07,0.02,0.06]
plt.pie(counts,labels=counts.index, autopct='%1.1f%%',
startangle=140,explode=myexplode,colors=['red','green','purple'])
plt.title("Subscription Type Distribution using piechart")
plt.show()
```

12.# Visualizing of data using piechart(Country)

```
counts = data['Country'].value_counts()# Replace 'Category' with the actual column name
myexplode=[0.05,0.02,0.03,0.02,0.02,0.02,0.02,0.02,0.02,0.02]
plt.pie(counts, labels=counts.index, autopct='%1.1f%%', startangle=140,explode=myexplode)
plt.title("Country Distribution in Dataset")
plt.show()
```

13.# Visualizing of data using piechart(Gender)

```
counts = data['Gender'].value_counts()# Replace 'Category' with the actual column name
myexplode=[0.05,0.02]
plt.pie(counts, labels=counts.index, autopct='%1.1f%%', startangle=140, colors=['red',
'blue'],explode=myexplode)
plt.title("Category Distribution in Dataset")
plt.legend(title='Gender Distribution',loc='upper left')
plt.show()
```

14.#Visualizing of data using Barplot(Subscription Type)

```
 plt.figure(figsize=(8, 5))
sns.countplot(x='Subscription Type', data=df, palette='coolwarm')  # Replace with actual column
name
plt.title("Number of Users per Subscription Type")
plt.xlabel("Subscription Type")
plt.ylabel("Count")
plt.show()
```

15.#Visualizing of data using ScatterPlot(AGE vs COUNTRY)

```
x=data['Age']
y=data['Country']
plt.scatter(x,y,color='m',alpha=0.2)
plt.xlabel('Age')
plt.ylabel('Country')
plt.title('Scatter Distribution of AGE vs COUNTRY')
plt.show()
```

16.#Visualizing of data using Histogram(Distribution of Age)

```
plt.figure(figsize=(8, 5))
plt.hist(df['Age'], bins=15, color='blue', edgecolor='black', alpha=0.7)  # Adjust bins as needed
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.title("Age Distribution of Netflix Users")
plt.show()
```

17.# Visualizing of data using Histogram(Distribution of Monthly Revenue)

```
plt.figure(figsize=(8, 5))
plt.hist(df['Monthly Revenue'],bins=10, color='green', edgecolor='black', alpha=0.7)
plt.xlabel("Monthly Revenue ($)")
plt.title("Monthly Revenue Distribution of Netflix Users")
plt.show()
```

18.# Visualizing of data using Histogram(Distribution of Country)

```
plt.figure(figsize=(20,10))
plt.hist(df['Country'], bins=30, color='purple', edgecolor='black', alpha=0.8)
plt.xlabel("Country")
plt.ylabel("Frequency")
plt.title("Watching Hours Distribution by Country")
plt.show()
```

19.#Visualizing of data using violin plot:

```
import seaborn as sns
plt.figure(figsize=(8, 5))
sns.violinplot(y=df['Age'], color='purple')
# Add labels and title
plt.title("Violin Plot for Age Distribution")
plt.ylabel("Age")
# Show plot
plt.show()
```

20.#visualising of data using violin plot different values:

```
plt.figure(figsize=(8, 5))
sns.violinplot(x=df['Age'], y=df['Subscription Type'], palette='coolwarm')  # Replace column names

# Add labels and title
plt.xlabel("Subscription Type")
plt.ylabel("Monthly Revenue")
plt.title("Violin Plot for Monthly Revenue per Subscription Type")

# Show plot
plt.show()
```

21#.Drop a column:

```
Dummy=data.drop(columns=['Monthly Revenue'])
Dummy
```

22.#Line plot using seaborn

```
sns.lineplot(data=data,x='Subscription Type',y='Device')
```

23.#Random integer plotted using Scatter plot

```
a=np.random.randint(100,size=(100))
b=np.random.randint(100,size=(100))
colors=np.random.randint(100,size=(100))
sizes=10*np.random.randint(100,size=(100))

plt.scatter(a,b,c=colors,cmap='nipy_spectral',s=sizes,alpha=0.6)
plt.xlabel('Monthly Revenue')
plt.ylabel('Subscription Type')
plt.colorbar()
plt.title('Monthly revenue VS Subscription Type')
plt.show()
```

24.# Getting inbuilt Datasets using Seaborn

```
datasets=sns.get_dataset_names()
print(datasets)
```

25.#Loading Datasets

```
data_sets=sns.load_dataset('diamonds')
data_sets
```

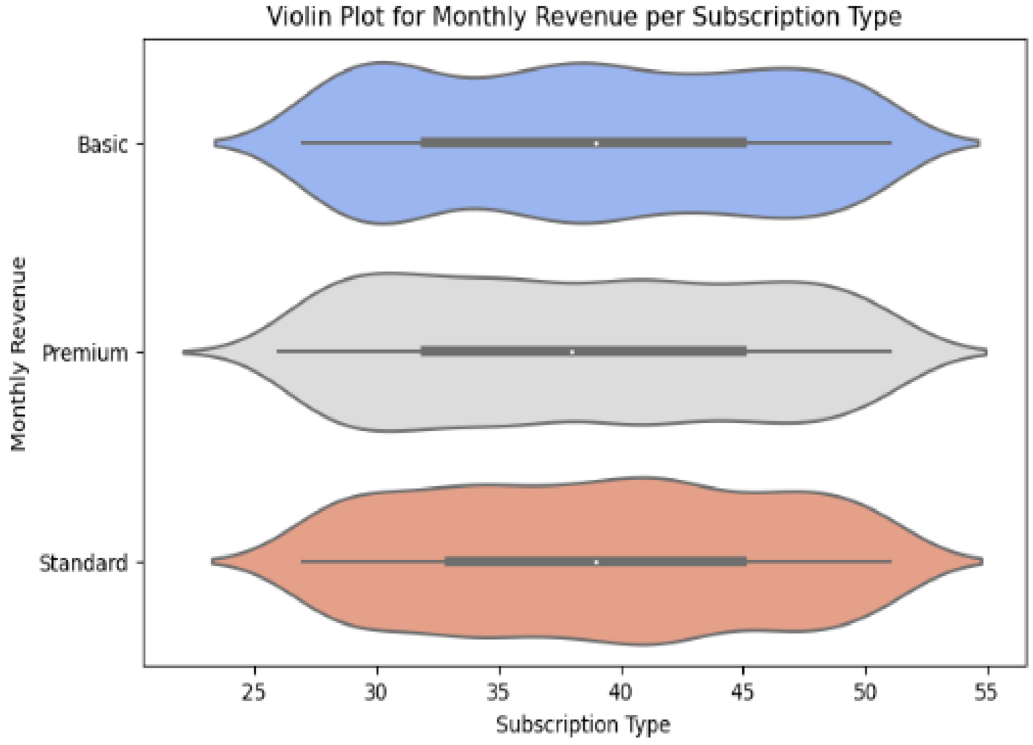
26.#Finding the Correlation

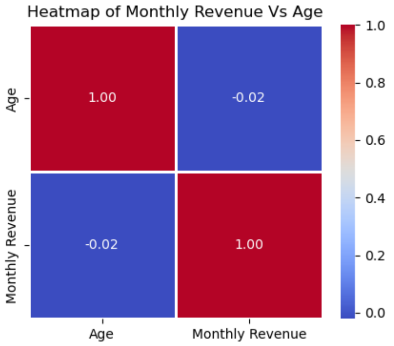
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Load dataset
df = pd.read_csv(r'C:\Users\raji0\OneDrive\Documents\anudip\Anudip Project\Netflix_userdatasets.csv')
# Select only two numerical columns (replace 'Column1' and 'Column2' with actual column names)
selected_columns = df[['Age','Monthly Revenue']]  # Example: Replace with your columns
# Compute correlation
correlation_matrix = selected_columns.corr()
# Print correlation matrix
print(correlation_matrix)
```

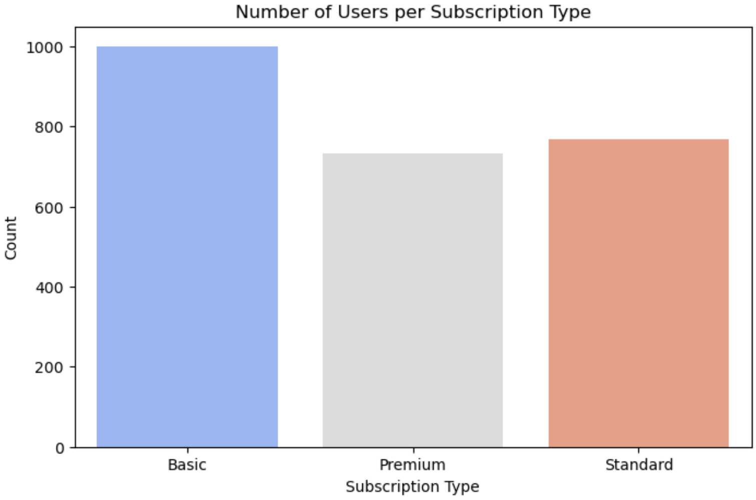
27.# Creating Heatmap

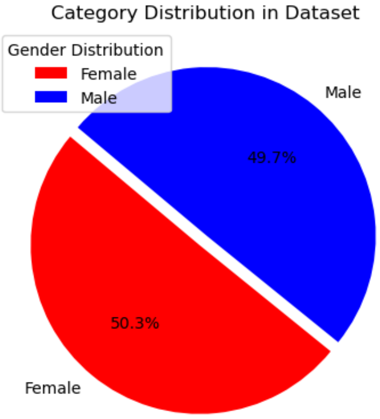
```
plt.figure(figsize=(5, 4))  # Set figure size
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=1, square=True)
# Add title
plt.title("Heatmap of Monthly Revenue Vs Age")
plt.show()
```

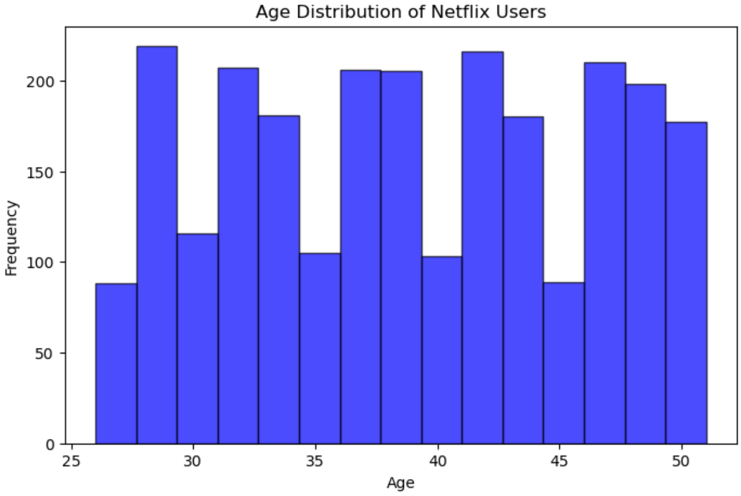
VISUALIZATIONS:

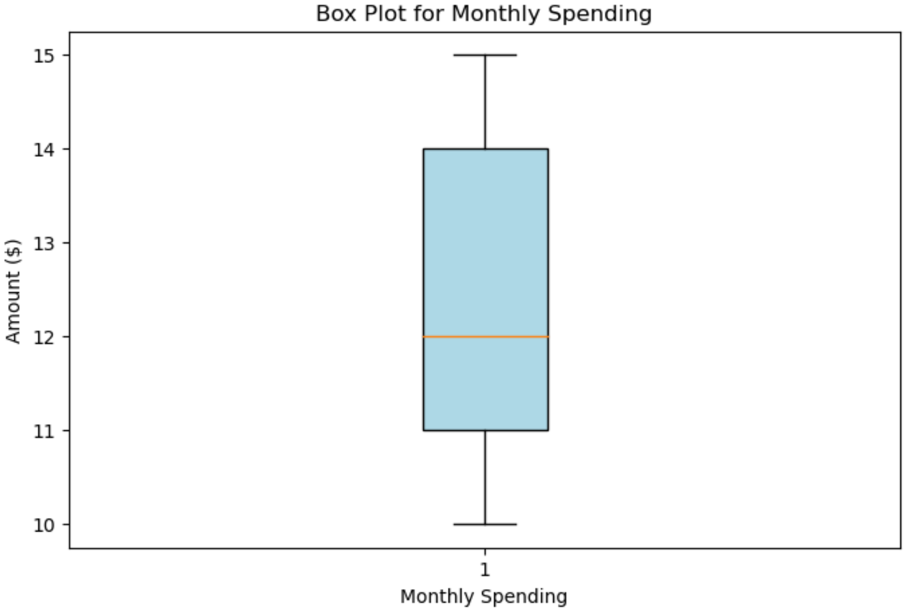


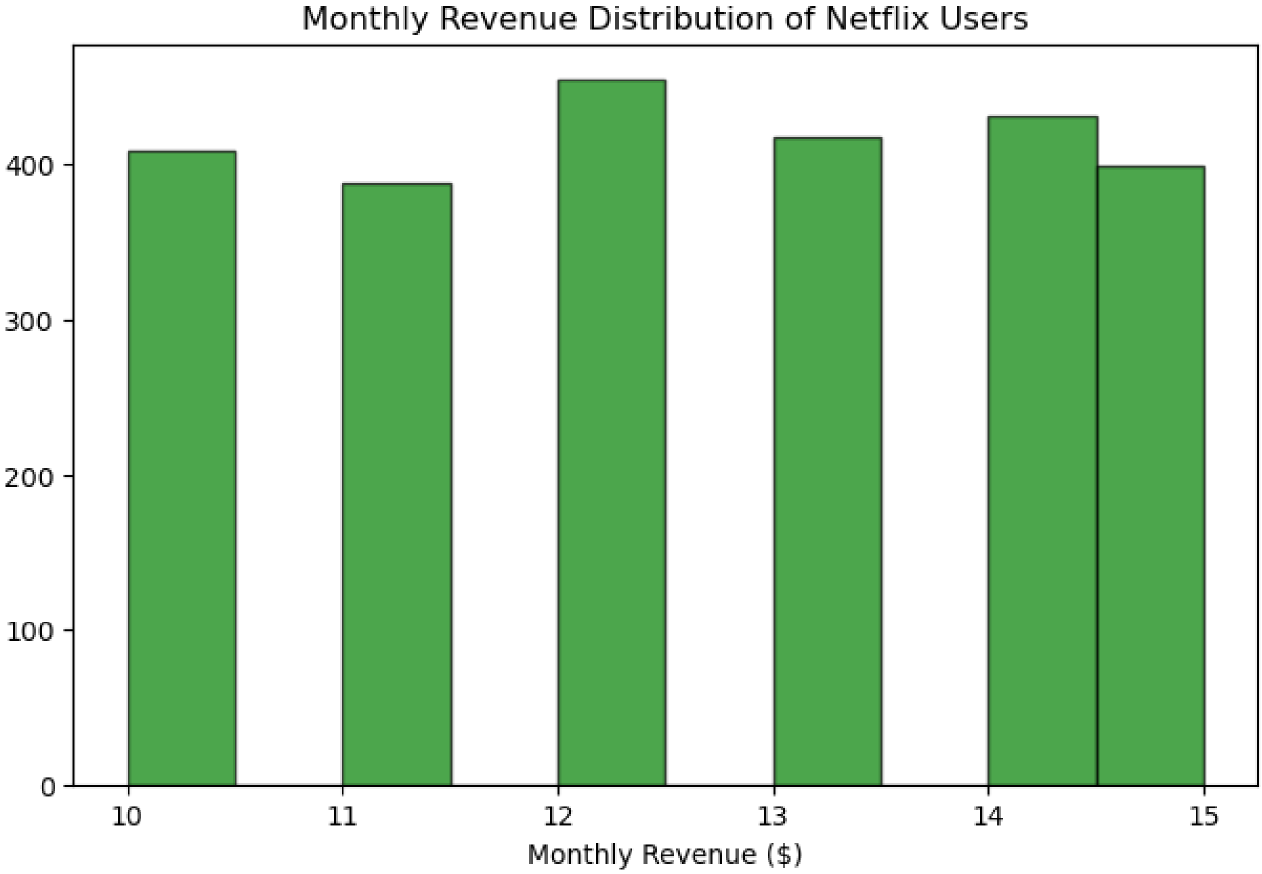

Heatmap of Monthly Revenue Vs Age

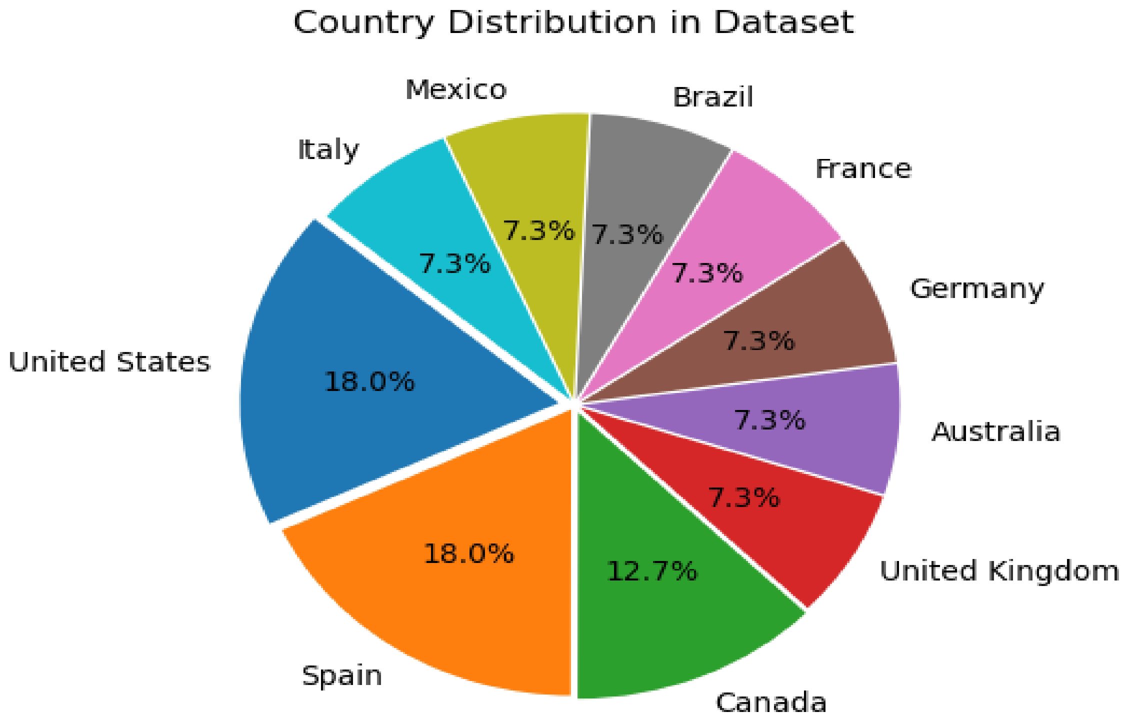

Number of Users per Subscription Type

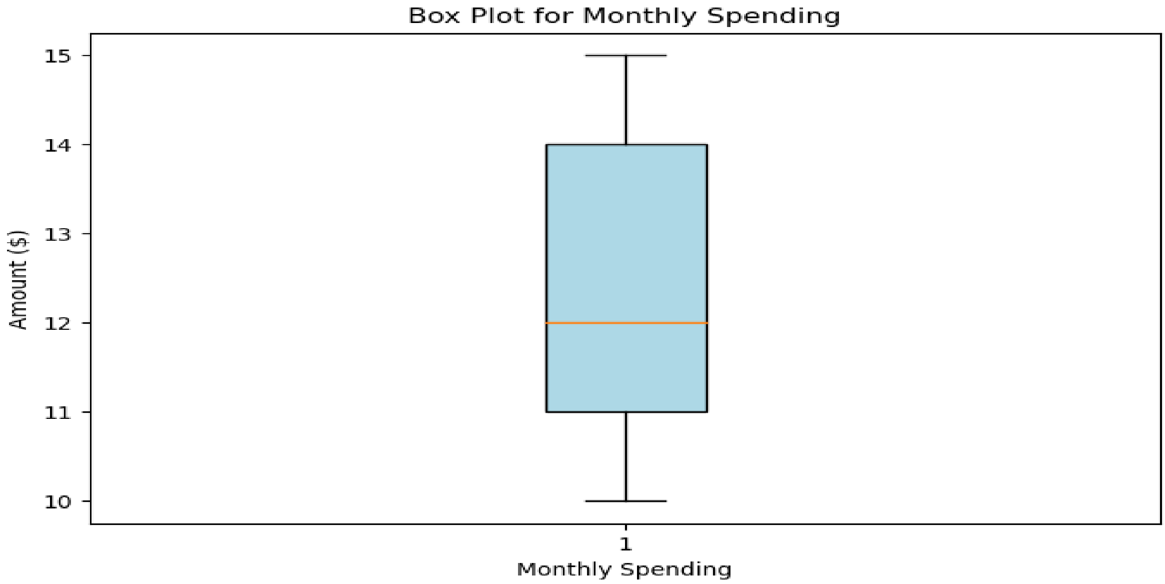

Category Distribution in Dataset

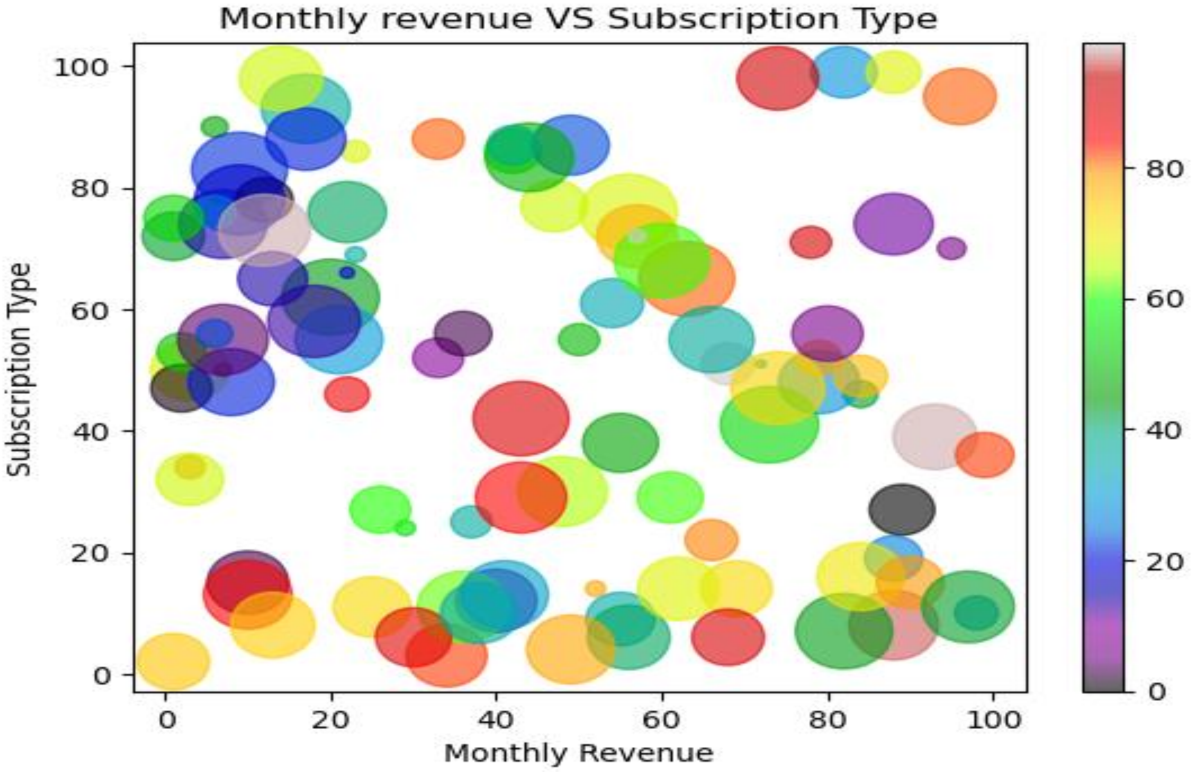

Age Distribution of Netflix Users

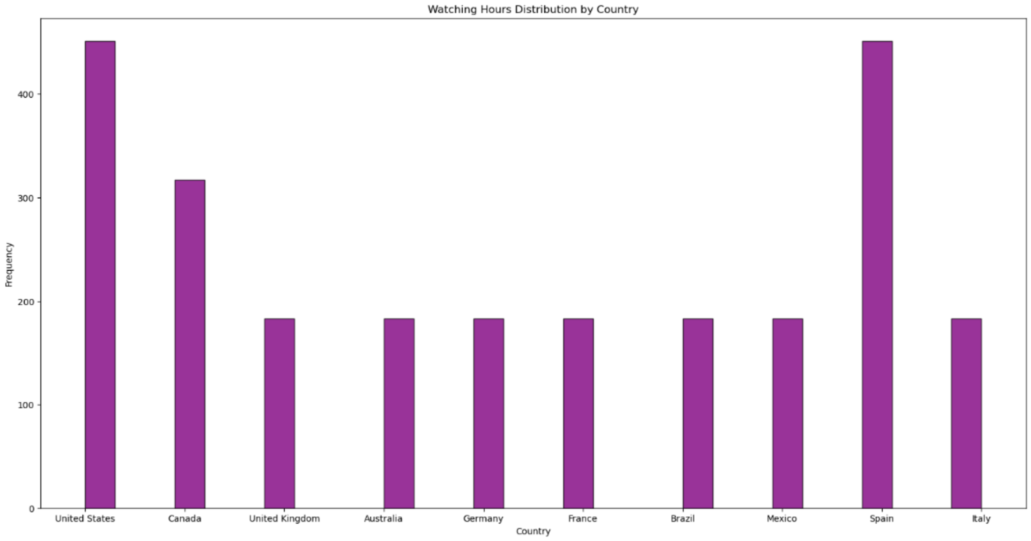

Box Plot for Monthly Spending
Amount ($)
15
14
13
12
11
10
1
Monthly Spending

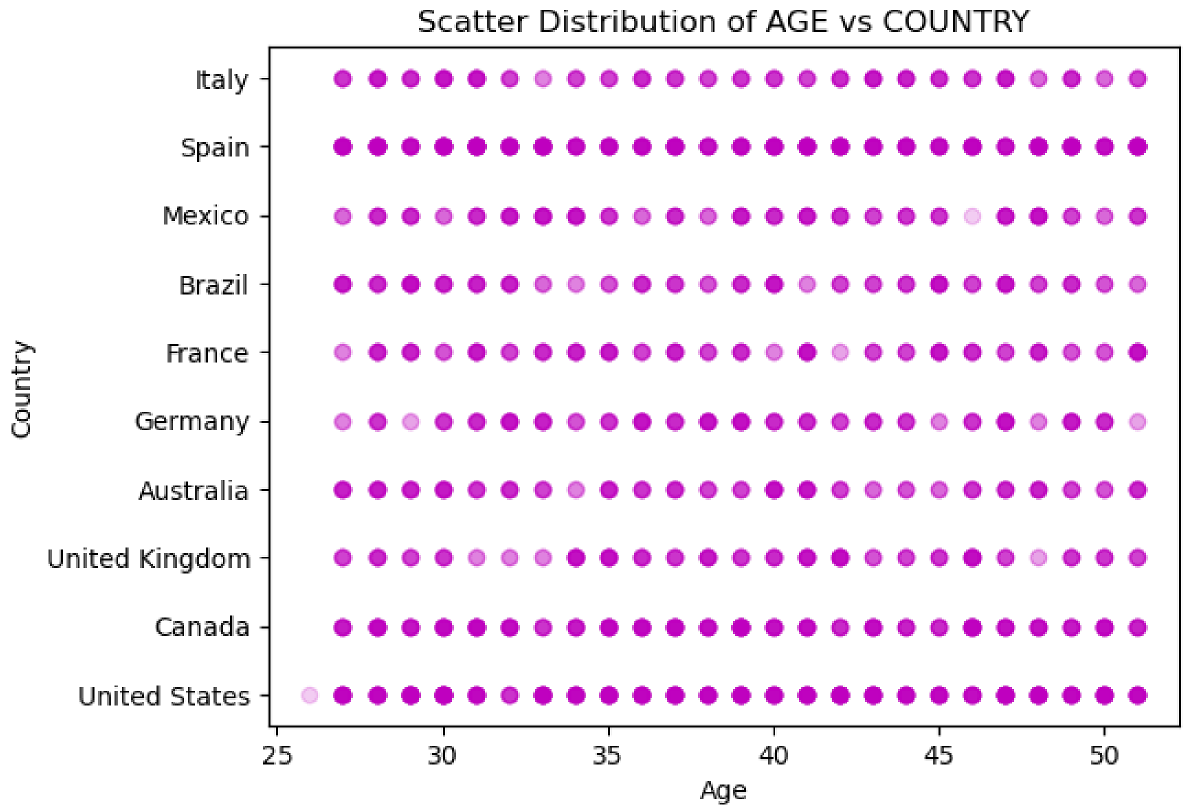

Monthly Revenue Distribution of Netflix Users
400
300
200
100
0
10
11
12
13
14
15
Monthly Revenue ($)


Country Distribution in Dataset
Mexico
Brazil
Italy
France
Germany
United States
Australia
United Kingdom
Spain
Canada
7.3%
7.3%
7.3%
7.3%
7.3%
7.3%
7.3%
18.0%
18.0%
12.7%


Box Plot for Monthly Spending
15
14
13
12
11
10
Amount ($)
1
Monthly Spending


Monthly revenue VS Subscription Type
Subscription Type
Monthly Revenue
0
20
40
60
80
100
0
20
40
60
80
100
0
20
40
60
80


Watching Hours Distribution by Country
Frequency
Country
0
100
200
300
400
United States
Canada
United Kingdom
Australia
Germany
France
Brazil
Mexico
Spain
Italy



Results:

1. **User Behavior Insights** – Most users prefer specific genres, and peak watching hours vary by age group.
2. **Churn Prediction** – Users with **low watch time and high monthly spending** are more likely to cancel subscriptions.
3. **Content Optimization** – Popular genres and trending shows drive engagement, while poor recommendations lead to disinterest.
4. **Pricing Strategy** – Tiered subscription plans with discounts improve retention and revenue.
5. **Competitive Analysis** – Users switch platforms due to better content variety and exclusive shows.

Conclusion:

This analysis highlights that **personalized recommendations, optimized pricing plans, and engaging content** are key to **reducing churn and increasing user satisfaction**. By using data insights, streaming platforms can **improve content discovery, enhance user experience, and develop competitive pricing models**. Implementing these strategies will help platforms **retain users, boost engagement, and maximize revenue growth** in the long run.