

Title: - Telegram Analysis Project

Name: Abhishek Guttedar

Reg No: P18BR23S126015

Department: MCA (BCU)

Semester: Third

† Objective: -

The primary objective of the Telegram Data Analysis project is to utilize data-driven insights to improve content strategies, enhance user experience, and optimize platform performance. This involves analysing Telegram's extensive content catalog and user interaction data to achieve the following specific objectives:

- Understand User Preferences and Behaviour
- Analyse Content Trends
- Recommendation Systems

† **Problem statement: -**

- With the rapid growth of the streaming industry, Telegram has amassed a vast collection of content and user data.
- However, effectively leveraging this data to enhance user satisfaction, improve content recommendations, and drive strategic decisions remains a challenge.

1. Content Trends Analysis
2. Recommendation System Enhancement
3. User Behaviour Insights

† **Solution: -**

To address the challenges and achieve the objectives, a comprehensive solution is proposed, combining advanced data analysis, machine learning, and visualization techniques. Below are the detailed components of the solution:

1. Data Collection and Preprocessing
2. Content Trends Analysis
3. User Behavior Analysis

✚ **Implementation: -**

```
import pandas as pd import seaborn as sns

import matplotlib.pyplot as plt import

warnings warnings.filterwarnings("ignore")

df =

pd.read_csv("E:/Telegram_Userbase.csv")

sns.set_theme(style="whitegrid")

# 1. Distribution of Monthly Revenue by Subscription Type
plt.figure(figsize=(12, 6)) sns.boxplot(data=df, x='Subscription
Type', y='Monthly Revenue', palette="Set2")
plt.title('Distribution of Monthly Revenue by Subscription Type',
fontsize=16) plt.xlabel('Subscription Type', fontsize=12)
plt.ylabel('Monthly Revenue', fontsize=12)
plt.xticks(rotation=45) plt.show()

# 2. Count of Users by Country (Top 10 Countries)

plt.figure(figsize=(12, 6))

top_countries = df['Country'].value_counts().head(10)

sns.barplot(x=top_countries.index, y=top_countries.values,
palette="viridis")
plt.title('Top 10 Countries by User Count', fontsize=16)

plt.xlabel('Country', fontsize=12) plt.ylabel('Number
```

```
of Users', fontsize=12) plt.xticks(rotation=45)
```

```
plt.show()
```

```
# 3. Age Distribution by Gender plt.figure(figsize=(12,  
6))
```

```
sns.histplot(data=df, x='Age', hue='Gender', multiple='stack',  
palette="coolwarm", bins=20) plt.title('Age Distribution by  
Gender', fontsize=16) plt.xlabel('Age', fontsize=12)  
plt.ylabel('Count', fontsize=12) plt.show()
```

```
# 4. Monthly Revenue Trends by Join Date
```

```
plt.figure(figsize=(12, 6)) df['Join Date'] =  
pd.to_datetime(df['Join Date']) df['Join Month'] =  
df['Join Date'].dt.to_period('M')
```

```
monthly_revenue = df.groupby('Join Month')['Monthly  
Revenue'].sum().reset_index()
```

```
monthly_revenue['Join Month'] = monthly_revenue['Join  
Month'].astype(str)
```

```
sns.lineplot(data=monthly_revenue, x='Join Month', y='Monthly  
Revenue', marker='o', color="purple") plt.title('Monthly  
Revenue Trends by Join Date', fontsize=16) plt.xlabel('Join  
Month', fontsize=12)
```

```
plt.ylabel('Monthly Revenue', fontsize=12)
```

```
plt.xticks(rotation=45) plt.show()
```

✚ Output: -



