Project Title: Subscription-Based Streaming Service Usage Analysis

Team members: S.Manimaran, R.Praveenkumar

Abstract: Subscription-Based Streaming Service Usage Analysis

This project analyzes user engagement patterns within subscription-based streaming platforms by leveraging comprehensive data processing, statistical analysis, and machine learning techniques. The analysis addresses missing values in viewing history and user ratings, ensuring data completeness for reliable insights. Data aggregation methods summarize total watch time per genre, while statistical summarization captures average session duration, providing a snapshot of user engagement.

The project implements pattern detection to identify binge-watching behaviors and segment users by watch time, unveiling engagement clusters. Visualizations present top binge-watched content, genre trends, and user segment distributions to simplify complex behavior patterns. Correlation analysis uncovers the relationship between user ratings and watch time, while trend detection visualizes genre popularity shifts.

Advanced techniques such as cosine similarity generate personalized content recommendations, enhancing user experiences. Predictive modeling with linear regression forecasts future watch time trends, supporting data-driven content strategies. Additionally, churn risk prediction identifies at-risk users, enabling proactive retention efforts. Time-based trend tracking and genre preference evolution analyses offer insights into shifting viewer interests and emerging content trends.

This project demonstrates how data-driven methodologies can provide actionable insights into user behavior, supporting streaming services in enhancing engagement, retention, and content optimization strategies.

Problem Statement:

With the rapid expansion of subscription-based streaming platforms, understanding user engagement patterns is crucial for sustaining viewer retention, optimizing content strategies, and driving platform growth. However, streaming services face several challenges in analyzing user behavior effectively:

- **Incomplete Data:** Missing viewing history and user ratings hinder accurate engagement analysis.
- **Unclear Engagement Patterns:** Traditional metrics fail to capture binge-watching habits, content preferences, and session behaviors.
- **Content Overload:** With vast libraries, identifying top-performing content and recommending personalized content remains complex.
- **User Churn:** Predicting users at risk of disengagement is critical to improving retention rates.
- **Trend Adaptation:** Platforms struggle to track evolving genre preferences and predict trending content to stay ahead of viewer interests.

This project addresses these challenges by building a data-driven analysis system that processes incomplete datasets, detects engagement patterns, segments users, predicts future behavior, and visualizes content trends. The goal is to provide streaming platforms with actionable insights to enhance user experience, minimize churn, and improve content delivery strategies.

PROJECT CODE:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear_model import LinearRegression

from sklearn.metrics.pairwise import cosine_similarity


# Load the dataset

watch_data = pd.read_csv('streaming_data.csv')
```

```
# Handling missing values
watch_data['viewing_history'].fillna('No Data', inplace=True)
watch_data['user_ratings'].fillna(watch_data['user_ratings'].mean(), inplace=True)

# Data aggregation - Total watch time per genre
genre_watch_time = watch_data.groupby('genre')['watch_time'].sum().reset_index()

# Statistical summarization - Average session duration
average_session_duration = watch_data['session_duration'].mean()
print(f"Average Session Duration: {average_session_duration:.2f} minutes")

# Pattern detection - Identifying binge-watching habits
watch_data['binge_watch'] = watch_data['session_duration'] > 60
binge_watchers = watch_data['binge_watch'].sum()
print(f"Number of binge watchers: {binge_watchers}")

# User segmentation based on watch time
watch_data['user_segment'] = pd.qcut(watch_data['watch_time'], q=4, labels=['Low',
'Medium', 'High', 'Very High'])
segment_distribution = watch_data['user_segment'].value_counts()

# Data visualization - Total Watch Time per Genre
plt.figure(figsize=(10, 6))
sns.barplot(x='genre', y='watch_time', data=genre_watch_time)
plt.title('Total Watch Time per Genre')
plt.xticks(rotation=45)
```

```
plt.show()


# Top binge-watched content
binge_content = watch_data[watch_data['binge_watch']].groupby('content')['session_duration'].sum().sort_values(ascending=False).head(10)
plt.figure(figsize=(10, 6))
sns.barplot(x=binge_content.values, y=binge_content.index)
plt.title('Top Binge-Watched Content')
plt.xlabel('Total Binge-Watch Duration (minutes)')
plt.show()


# Visualizing user segment distribution
plt.figure(figsize=(8, 6))
sns.barplot(x=segment_distribution.index, y=segment_distribution.values)
plt.title('User Segmentation Based on Watch Time')
plt.xlabel('User Segment')
plt.ylabel('Number of Users')
plt.show()


# Correlation analysis - Ratings vs Watch Time
plt.figure(figsize=(8, 6))
sns.scatterplot(x='user_ratings', y='watch_time', data=watch_data)
plt.title('Correlation between User Ratings and Watch Time')
plt.xlabel('User Ratings')
plt.ylabel('Total Watch Time (minutes)')
```

```
plt.show()


# Trend detection - Genre popularity over watch time

plt.figure(figsize=(10, 6))

sns.lineplot(x='genre', y='watch_time', data=genre_watch_time.sort_values('watch_time', ascending=False))

plt.title('Genre Popularity Trend Based on Watch Time')

plt.xticks(rotation=45)

plt.show()


# Content recommendation based on top genres

top_genre = genre_watch_time.sort_values('watch_time', ascending=False).iloc[0]['genre']

recommended_content = watch_data[watch_data['genre'] == top_genre]['content'].unique()[:5]

print(f"Top Genre: {top_genre}")

print("Recommended Content:", ', '.join(recommended_content))


# Personalized content recommendation using cosine similarity

user_profile = watch_data.pivot_table(index='user_id', columns='genre', values='watch_time', fill_value=0)

similarity_matrix = cosine_similarity(user_profile)

user_idx = 0  # Example user (first user in dataset)

similar_users = np.argsort(similarity_matrix[user_idx])[::-1][1:6]

recommended_for_user = watch_data[watch_data['user_id'].isin(similar_users)].groupby('content')['watch_time'].sum().sort_values(ascending=False).head(5).index.tolist()

print(f"Personalized Recommendations for User {watch_data['user_id'].iloc[user_idx]}: {', '.join(recommended_for_user)}")
```

```
# Time-based trend tracking
if 'date' in watch_data.columns:
    watch_data['date'] = pd.to_datetime(watch_data['date'])
    daily_watch_trend =
watch_data.groupby(watch_data['date'].dt.date)['watch_time'].sum()
    plt.figure(figsize=(10, 6))
    sns.lineplot(x=daily_watch_trend.index, y=daily_watch_trend.values)
    plt.title('Daily Watch Time Trend')
    plt.xlabel('Date')
    plt.ylabel('Total Watch Time (minutes)')
    plt.xticks(rotation=45)
    plt.show()
else:
    print("Date column not found in dataset. Time-based trend tracking skipped.")

# Churn risk prediction: Users who stop watching
watch_data['churn_risk'] = watch_data['watch_time'] < (watch_data['watch_time'].mean()
* 0.5)
churn_risk_users = watch_data[watch_data['churn_risk']].shape[0]
print(f"Number of users at risk of churn: {churn_risk_users}")

# Future watch time prediction
if 'date' in watch_data.columns:
    watch_data['day_number'] = (watch_data['date'] - watch_data['date'].min()).dt.days
    X = watch_data[['day_number']]
```

```
    y = watch_data['watch_time']
    model = LinearRegression()
    model.fit(X, y)
    future_days = np.array([[i] for i in range(watch_data['day_number'].max() + 1, watch_data['day_number'].max() + 8)])
    future_watch_time = model.predict(future_days)
    plt.figure(figsize=(10, 6))
    plt.plot(watch_data['day_number'], watch_data['watch_time'], label='Actual Watch Time')
    plt.plot(future_days.flatten(), future_watch_time, linestyle='dashed', color='red', label='Predicted Watch Time')
    plt.title('Future Watch Time Prediction (Next 7 Days)')
    plt.xlabel('Days from Start')
    plt.ylabel('Total Watch Time (minutes)')
    plt.legend()
    plt.show()
else:
    print("Date column not found in dataset. Future prediction skipped.")

# Predicting trending content
trending_content = watch_data.groupby('content')['watch_time'].mean().sort_values(ascending=False).head(5)
print("Trending Content Predictions:", trending_content.index.tolist())

# Genre preference evolution
if 'date' in watch_data.columns:
```

```
    genre_trend_over_time = watch_data.groupby([watch_data['date'].dt.month,
'genre'])['watch_time'].sum().unstack().fillna(0)
    genre_trend_over_time.plot(figsize=(10, 6))
    plt.title('Genre Preference Evolution Over Time')
    plt.xlabel('Month')
    plt.ylabel('Total Watch Time (minutes)')
    plt.legend(title='Genre')
    plt.show()
else:
    print("Date column not found in dataset. Genre preference evolution skipped.")
```

Outputs:

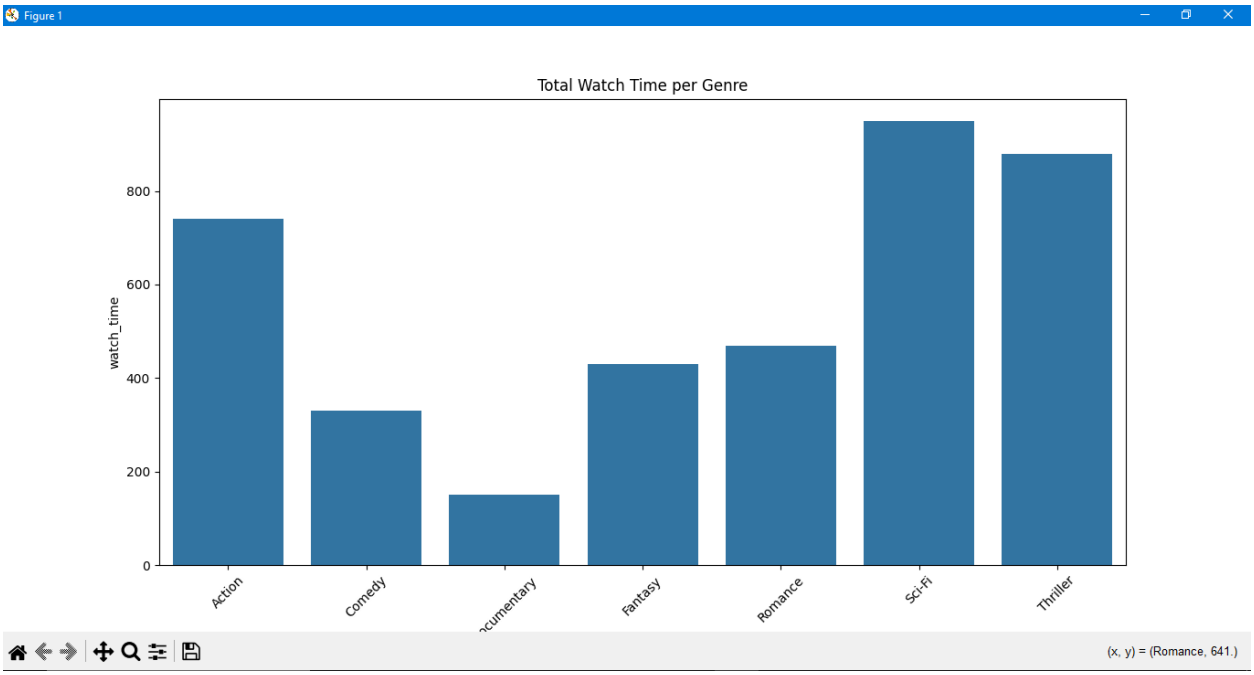
Figure 1
Total Watch Time per Genre
watch_time
0
200
400
600
800
Action
Comedy
Fantasy
Romance
Sci-Fi
Thriller
(x, y) = (Romance, 641.)

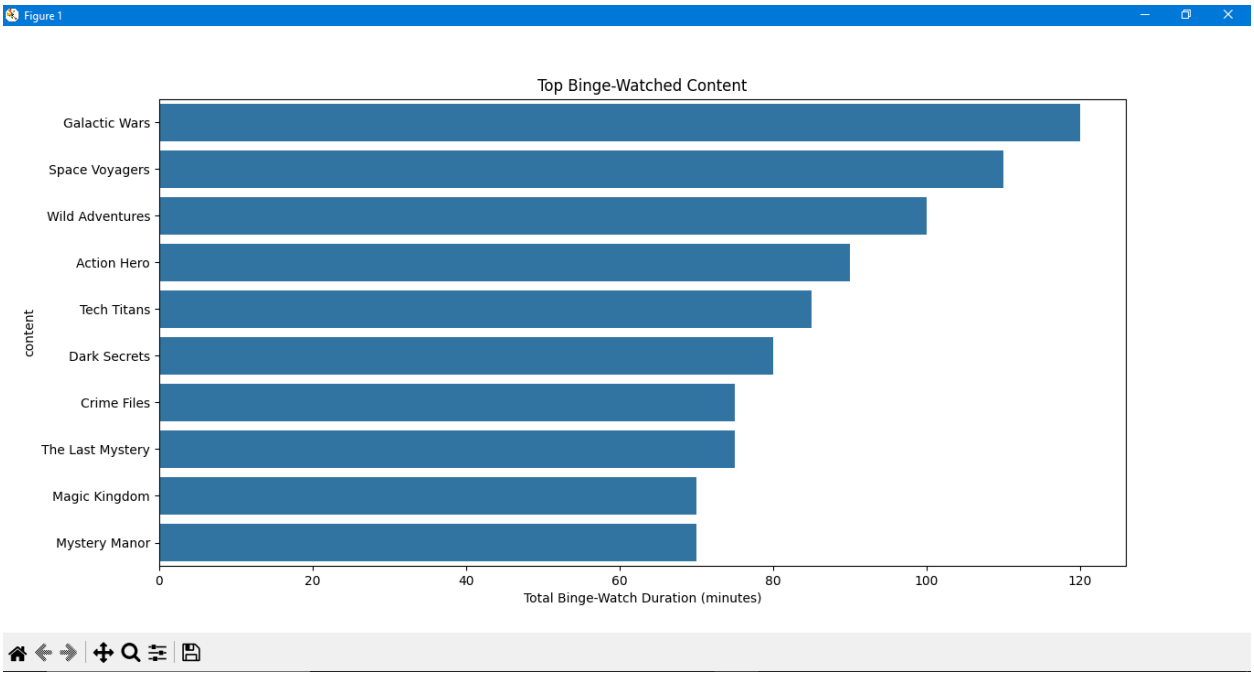
Figure 1
Top Binge-Watched Content
content
Galactic Wars
Space Voyagers
Wild Adventures
Action Hero
Tech Titans
Dark Secrets
Crime Files
The Last Mystery
Magic Kingdom
Mystery Manor
0
20
40
60
80
100
120
Total Binge-Watch Duration (minutes)

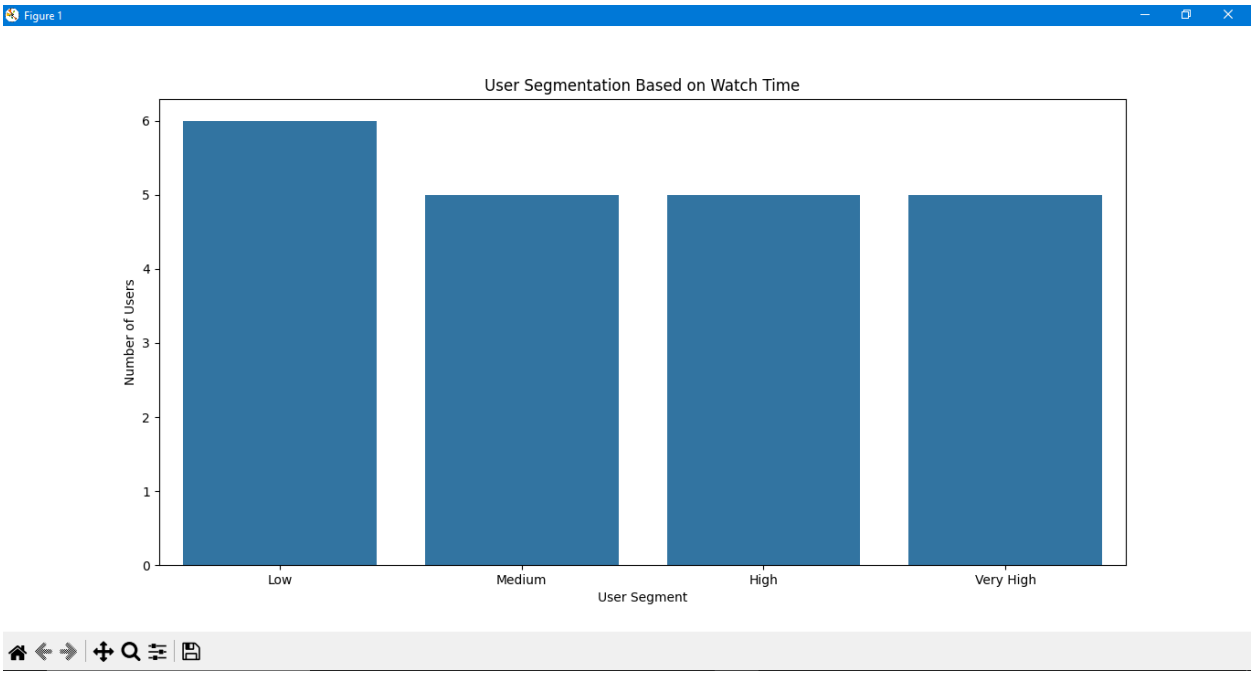
Figure 1
User Segmentation Based on Watch Time
Number of Users
0
1
2
3
4
5
6
Low
Medium
High
Very High
User Segment

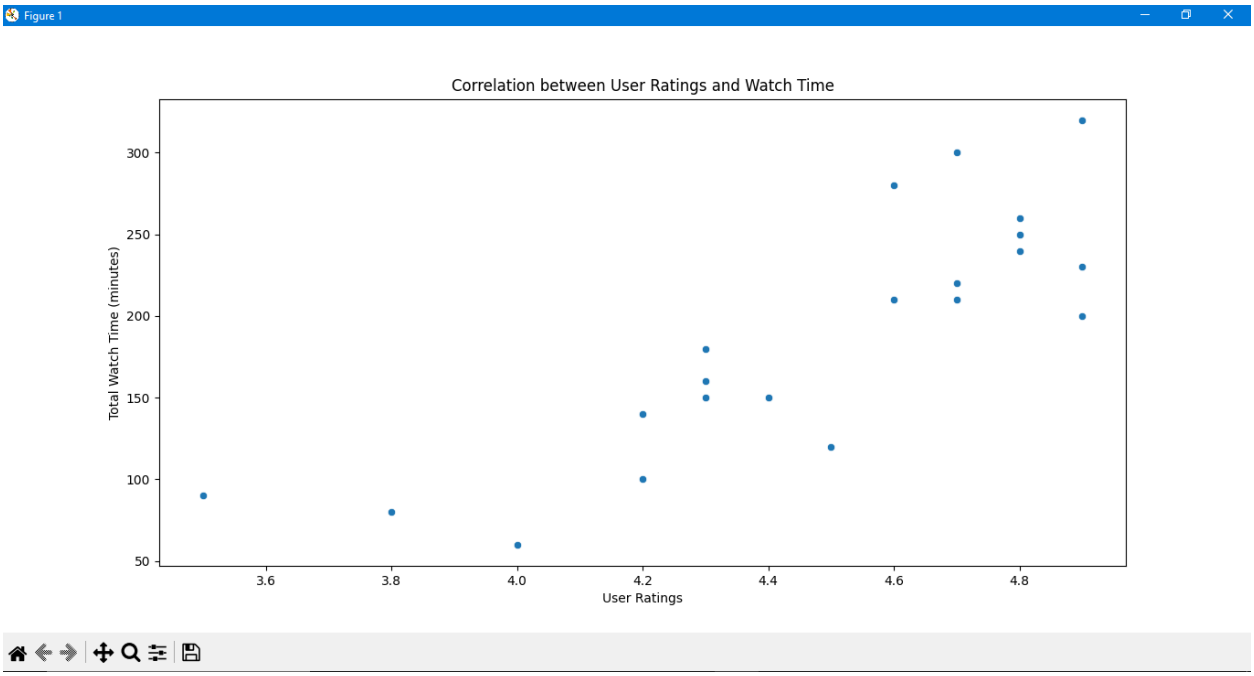
Figure 1
Correlation between User Ratings and Watch Time
Total Watch Time (minutes)
50
100
150
200
250
300
3.6
3.8
4.0
4.2
4.4
4.6
4.8
User Ratings

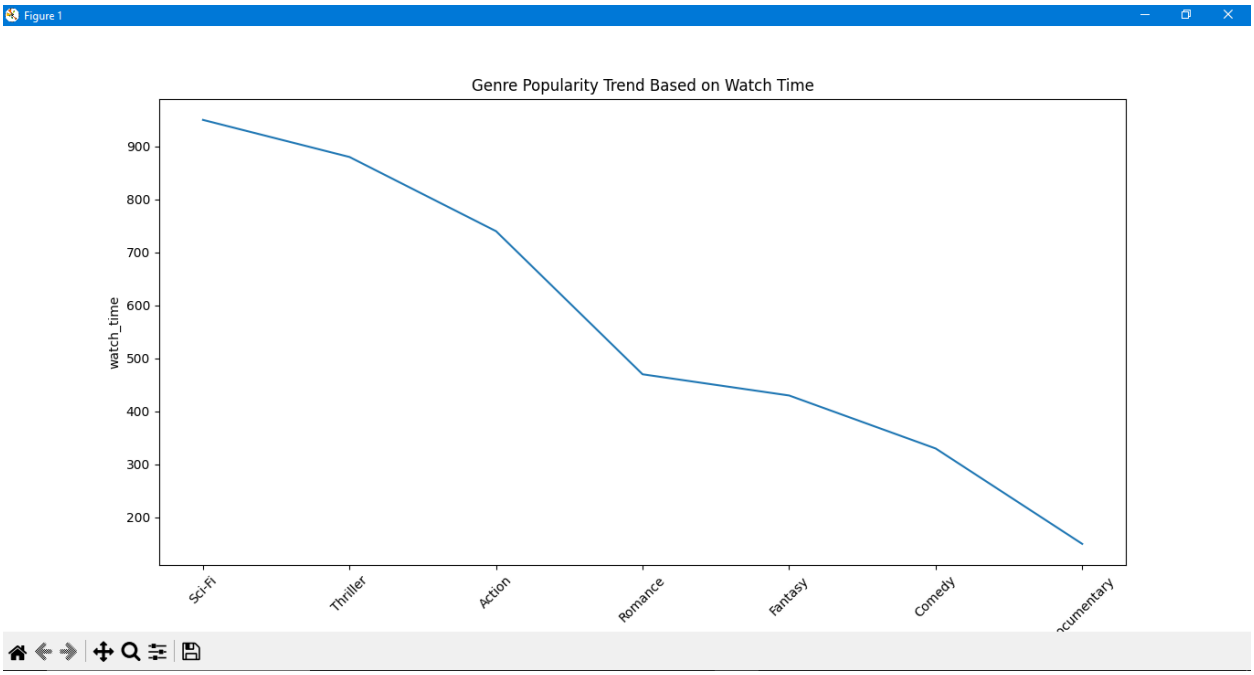
Figure 1
Genre Popularity Trend Based on Watch Time
watch_time
900
800
700
600
500
400
300
200
Sci-Fi
Thriller
Action
Romance
Fantasy
Comedy

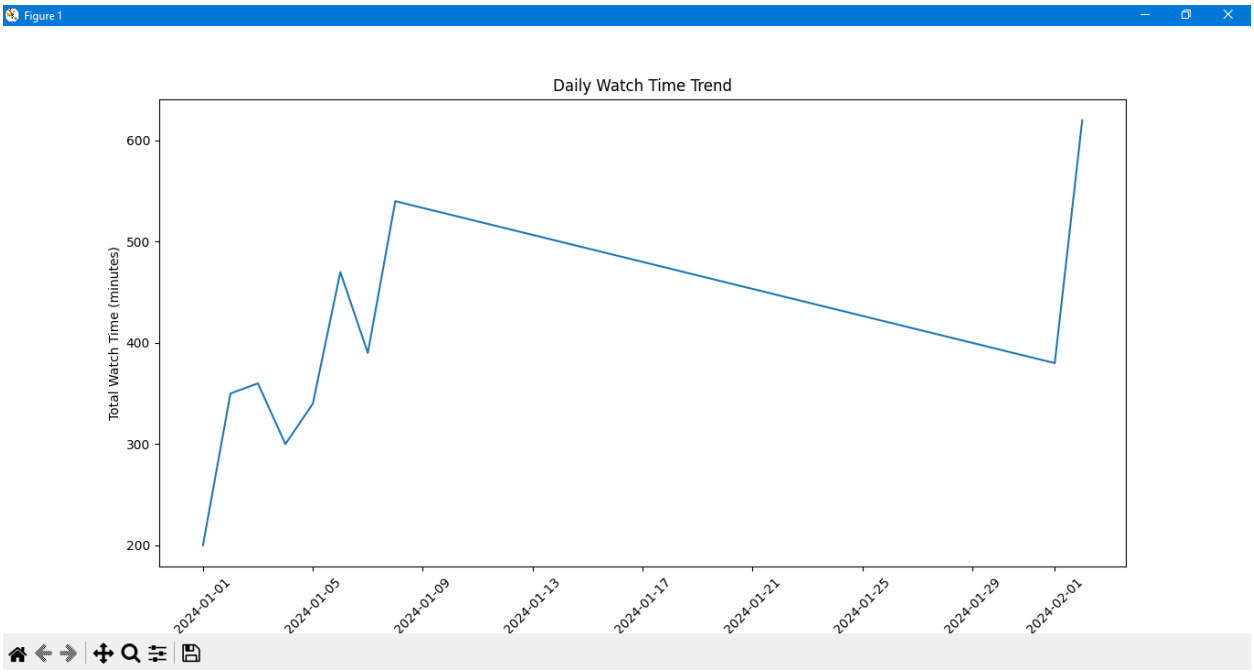
Figure 1
Daily Watch Time Trend
Total Watch Time (minutes)
600
500
400
300
200
2024-01-01
2024-01-05
2024-01-09
2024-01-13
2024-01-17
2024-01-21
2024-01-25
2024-01-29
2024-02-01

Conclusion:

This project successfully analyzed user engagement patterns in subscription-based streaming services through data handling, statistical summarization, and machine learning techniques. By addressing missing data in viewing history and user ratings, the analysis ensured data completeness, enabling more accurate insights. Data aggregation methods revealed watch time distributions across genres, while statistical summarization provided a clearer picture of session durations and user behavior.

Advanced pattern detection uncovered binge-watching habits, and user segmentation identified key engagement clusters. Visualizations of genre popularity, top binge-watched content, and user segmentation distributions transformed complex datasets into easily interpretable insights. Furthermore, content recommendations — powered by cosine similarity — delivered personalized suggestions, enhancing the user experience.

Predictive modeling using linear regression forecasted future watch time trends, while churn risk analysis identified users likely to disengage, supporting proactive retention strategies. Time-based trend analysis and genre preference evolution provided a dynamic view of changing user interests and emerging content trends.

In conclusion, this project demonstrates how data-driven techniques can unlock valuable insights into user behavior, empowering streaming platforms to improve content strategies, boost user engagement, and mitigate churn. The approach outlined in this analysis serves as a foundation for further exploration into user behavior modeling, predictive analytics, and recommendation system enhancements within the fast-evolving digital entertainment landscape.