Netflix Viewing Behavior Analysis Report

Begüm Özay

May 30, 2025

Abstract

A comprehensive analysis of personal Netflix viewing activity from January 2019 through May 2025 is presented. Temporal patterns at daily, weekly, and hourly resolutions are characterized; series and movie consumption are compared; binge-watching behaviors and seasonal effects are quantified; correlations among key metrics are explored; hypothesis tests are conducted; and predictive models for day-of-week and time-of-day classification are evaluated. All findings are supported by detailed visualizations and code excerpts.

1 Introduction

This document reports on the temporal and content preferences of a single Netflix user, herein referred to as the subject. The objectives of the study are to:

- Characterize viewing patterns over time (daily, weekly, hourly).
- Compare consumption between episodic series and stand-alone movies.
- Quantify binge-watching frequency, identify peak months, and determine favorite binge titles.
- Assess seasonal influences on session frequency and duration.
- Explore statistical correlations and perform formal hypothesis tests.
- Develop and evaluate classification models for viewing context.

2 Data and Preprocessing

The raw dataset (`ViewingActivity.csv`) contains semicolon-delimited records beginning with a secondary header. Key fields include:

- `Start Time`: playback start timestamp.
- `Duration`: session length in HH:MM:SS.
- `Title`: program title, often including season and episode.
- `Device Type`, `Country`, and related metadata.

Data cleaning and feature engineering were performed as follows:

```
import pandas as pd
from pathlib import Path

# Read raw CSV with semicolon delimiter and header row offset
netflix_data = pd.read_csv(
```

```
    Path('ViewingActivity.csv'),
    sep=';',
    header=1,
    encoding='utf-8'
)

# Parse datetime and compute additional fields
netflix_data['timestamp'] = pd.to_datetime(
    netflix_data['Start Time']
)
netflix_data['duration_min'] = (
    netflix_data['Duration']
    .str.split(':').apply(lambda x: int(x[0])*60 + int(x[1]) + int(x
       [2])/60)
)
netflix_data['date'] = netflix_data['timestamp'].dt.date
netflix_data['hour'] = netflix_data['timestamp'].dt.hour
netflix_data['day_of_week'] = netflix_data['timestamp'].dt.day_name()
netflix_data['DayNum'] = netflix_data['timestamp'].dt.dayofweek
```

Listing 1: Reading and initial parsing of raw data

Titles were decomposed into series, season, and episode components. Sessions without a season/episode pattern were classified as movies:

```
import re

pattern = r'^(.*?)(?:[: ]Season (\d+))?: (.*)$'
split_df = netflix_data['Title'].str.extract(pattern)
split_df.columns = ['Series', 'Season', 'Episode']
split_df['Series'] = split_df['Series'].fillna(netflix_data['Title'])
split_df['Season'] = split_df['Season'].fillna('Movie')
split_df['Episode'] = split_df['Episode'].fillna('Movie')
split_df['Season'] = split_df['Season'].astype('category')

df_clean = pd.concat(
    [netflix_data.drop(columns=['Title']), split_df],
    axis=1
)
df_clean['ContentType'] = df_clean['Season'].apply(
    lambda x: 'Movie' if x=='Movie' else 'Series'
)
```

Listing 2: Splitting Title into Series/Season/Episode

A `SeasonBin` variable was derived from month:

```
def month_to_season(m):
    if m in [12,1,2]: return 'Winter'
    if m in [3,4,5]:  return 'Spring'
    if m in [6,7,8]:  return 'Summer'
    return 'Fall'

df_clean['SeasonBin'] = (
    df_clean['timestamp']
    .dt.month
    .apply(month_to_season)
    .astype('category')
)
```

Listing 3: Assigning Meteorological Seasons

3 Temporal Viewing Patterns

3.1 Monthly and Weekly Trends

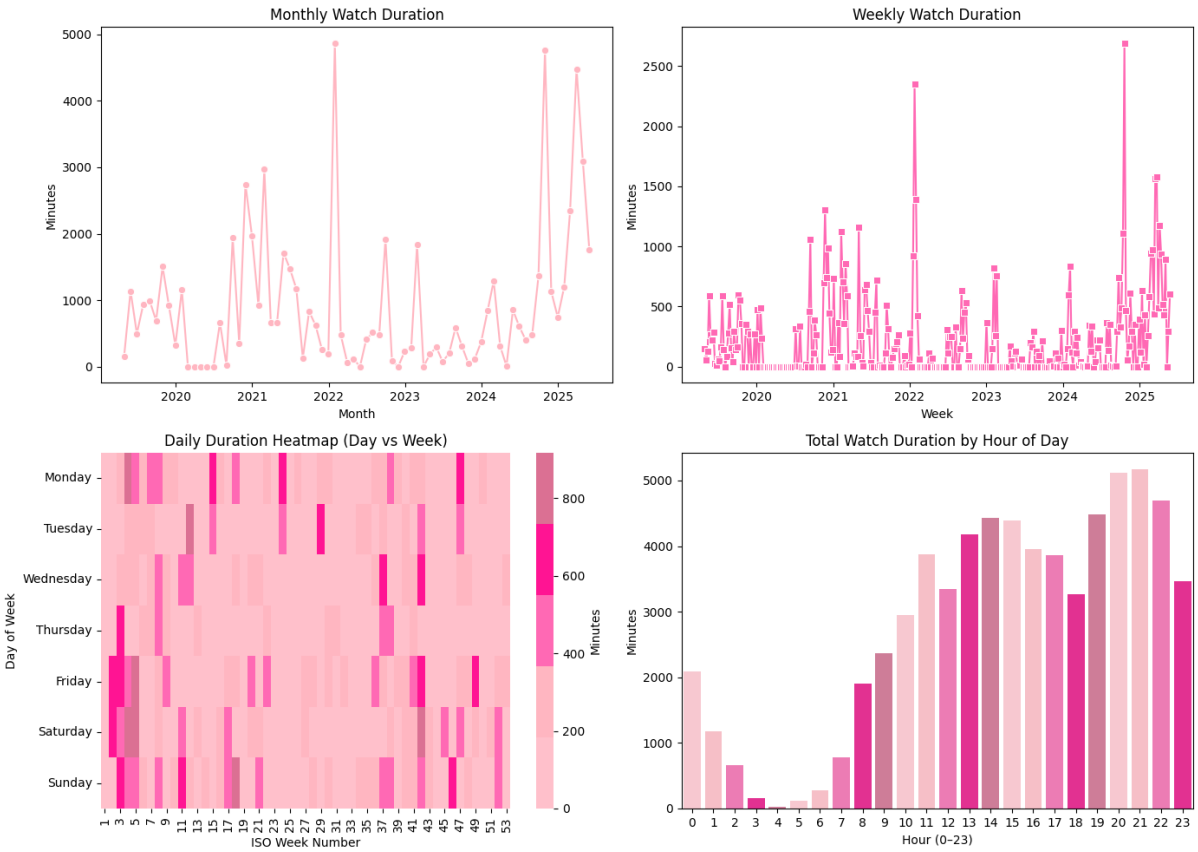

Figure 1: (a) Monthly total watch duration; (b) Weekly total watch duration; (c) Heatmap of daily duration by ISO week & weekday; (d) Hourly distribution of total duration.

Monthly durations peaked in January 2022 (4,900 min), November 2024 (4,800 min), and March 2025 (4,500 min), while dips occurred in mid-2020 and mid-2023. Weekly aggregates exceeded 2,700 min in high-intensity intervals and fell below 500 min during troughs. The heatmap showed heavier weekend usage (Fridays/Saturdays) and intermittent midweek spikes. Hourly distribution revealed prime viewing hours at 20:00–22:00, confirming evening-dominant consumption.

3.2 Series vs. Movie Share

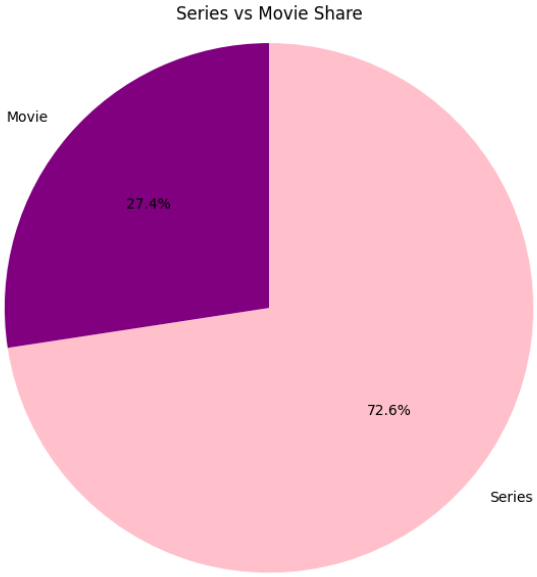

Figure 2: Proportion of sessions classified as Series versus Movies.

Series accounted for 72.6% of sessions, with movies comprising 27.4%. This reflects a strong episodic preference.

4 Content Breakdown

4.1 Top Series Titles and Episode Durations

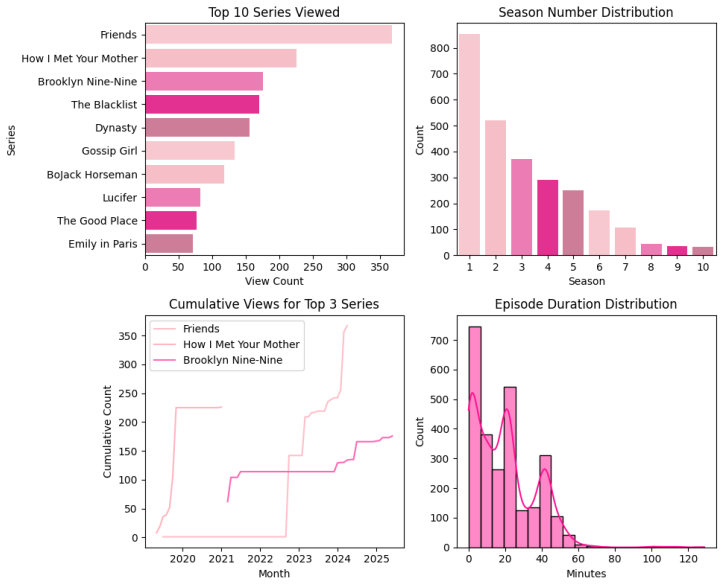

Figure 3: (a) Top 10 series by view count; (b) Distribution of seasons; (c) Episode duration histogram; (d) Cumulative views for top 3 series.

- **Friends** was the most-viewed series (359 episodes), followed by *How I Met Your Mother* (195) and *Brooklyn Nine-Nine* (170).

- Season distribution showed a long tail, with Season 1 (around 850), Season 2 (around 520), down to around 40 episodes at Season 10.
- Episode durations exhibited peaks at 22–25 min (sitcoms) and 45–50 min (dramas).
- Cumulative plots indicated renewed interest in Friends during late 2024.

5 Binge-Watching Patterns

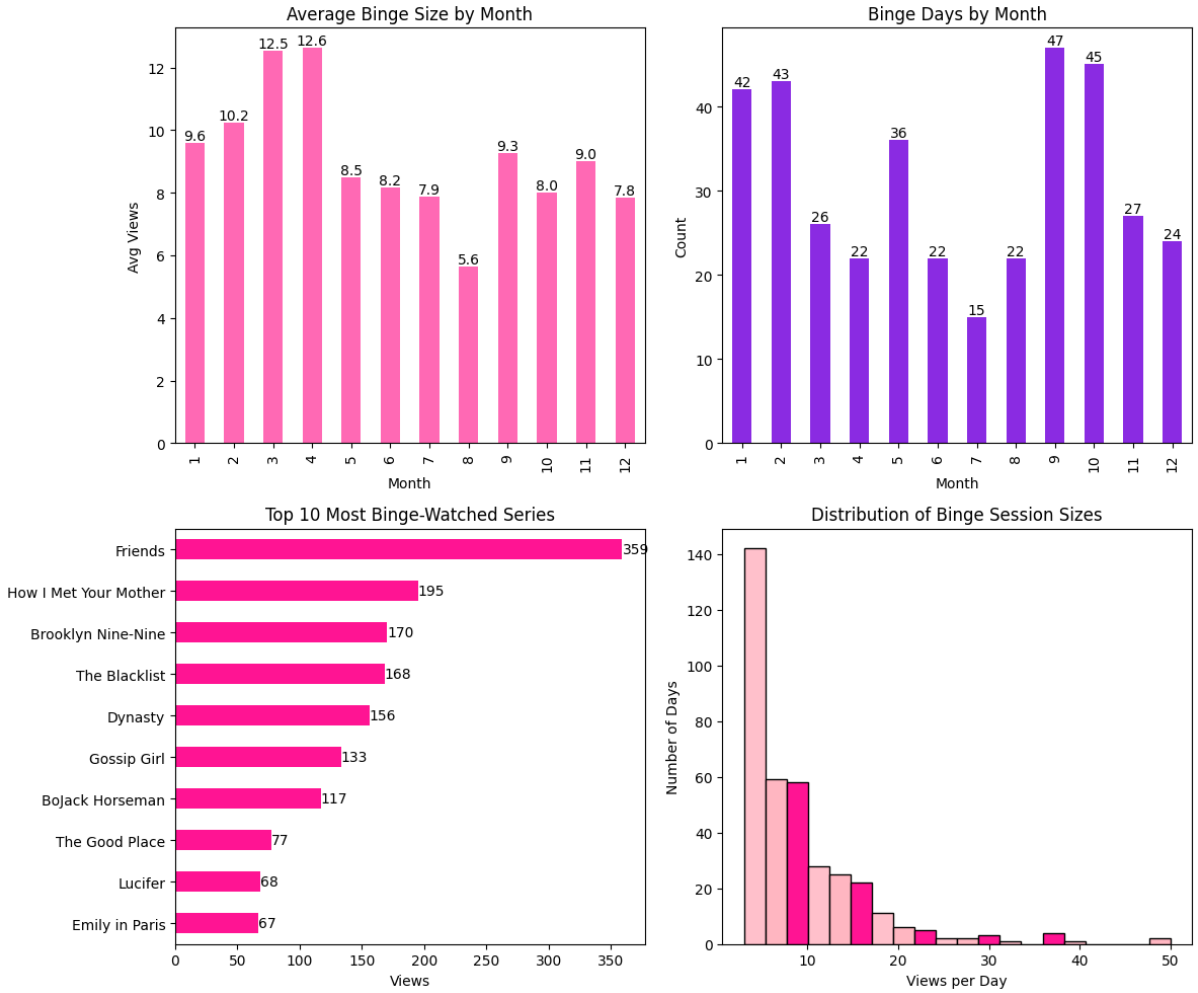

Figure 4: Binge-watching analysis: (a) Average binge size by month; (b) Number of binge days by month; (c) Top binge series; (d) Distribution of binge sizes.

Days with at least three sessions were classified as binge days. Average binge sizes peaked in March (12.6 eps) and April (12.5 eps), and troughs occurred in August (5.6 eps). September (47) and October (45) had the most binge days. Friends dominated binge counts (359), followed by HIMYM (195). The distribution of binge sizes revealed a right-skewed pattern, with occasional marathon days exceeding 20 episodes.

6 Seasonal Viewing Patterns

Fall (1100) and Winter (1097) led session counts; Spring (986) and Summer (498) trailed. Winter also had the greatest total duration (22,018 min), followed by Fall (19,589 min). Series

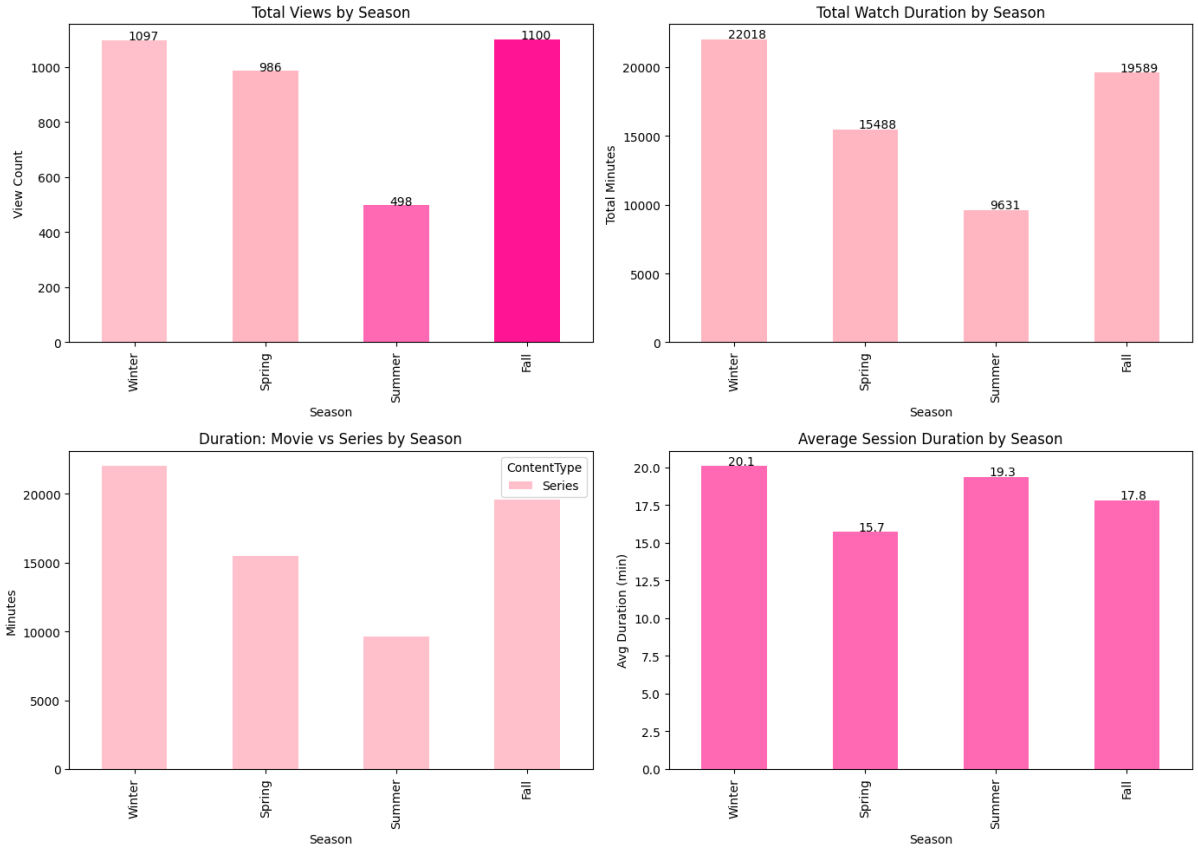

Figure 5: Seasonal aggregates: (a) Session counts; (b) Total minutes; (c) Movie vs. Series minutes; (d) Average session duration.

consumption far exceeded movies in each season. Average session duration was highest in Winter (20.1 min) and lowest in Spring (15.7 min), suggesting seasonal engagement differences.

7 Correlation and Trend Analysis

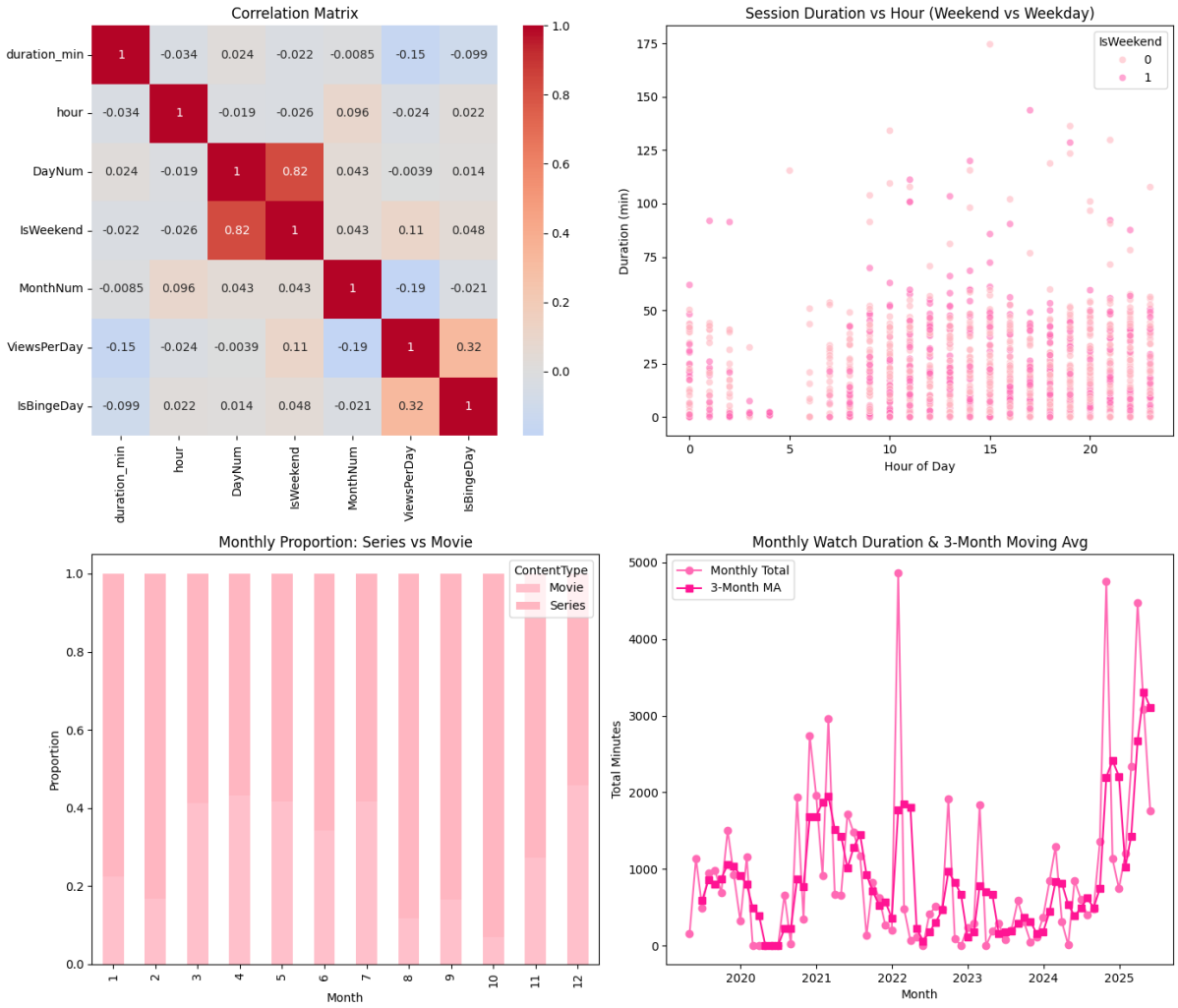

Figure 6: (a) Correlation matrix among key metrics; (b) Session duration vs. start hour, colored by weekend flag.

The correlation between daily views and binge-day indicator was moderate ($r = 0.32$). Session duration was largely uncorrelated with time-of-day or day-of-week, indicating that length of viewing is independent of temporal context. Scatter visualization confirmed no distinct weekend versus weekday clusters in session length.

8 Hypothesis Testing

Formal hypotheses were tested at $\alpha = 0.05$:

8.1 Weekend vs. Weekday Daily Views

- H_0: The mean daily view count on weekends equals that on weekdays.
- H_1: The means differ.

```
from scipy import stats

# Aggregate daily views
adv = df_clean.copy()
adv['DateOnly'] = adv['timestamp'].dt.date
adv['IsWeekend'] = adv['DayNum'].isin([5,6]).astype(int)
adv['ViewsPerDay'] = adv.groupby('DateOnly')['Series'].transform('count
    ')
daily = adv.groupby('DateOnly').agg({'ViewsPerDay':'first','IsWeekend
    ':'first'})

wknd = daily.loc[daily['IsWeekend']==1,'ViewsPerDay']
wday = daily.loc[daily['IsWeekend']==0,'ViewsPerDay']
t1, p1 = stats.ttest_ind(wknd, wday, equal_var=False)
print(f"t = {t1:.2f}, p = {p1:.4f}")
```

Listing 4: T-test for Weekend vs. Weekday Views

Result: $t = 1.68,\ p = 0.0943 \Rightarrow$ Fail to reject H_0.

8.2 Winter vs. Summer Binge Rates

- H_0: Binge-day rate in Winter equals that in Summer.
- H_1: The rates differ.

```
binge = daily.copy()
binge['IsBinge'] = (binge['ViewsPerDay']>=3).astype(int)
binge['SeasonBin'] = adv['SeasonBin']
tbl = pd.crosstab(binge['SeasonBin'], binge['IsBinge']).loc[['Winter','
    Summer']]
chi2, p2, _, _ = stats.chi2_contingency(tbl)
print(f"chi2 = {chi2:.2f}, p = {p2:.4f}")
```

Listing 5: Chi-square test for Winter vs. Summer Binge Rates

Result: $\chi^2 = 8.41,\ p = 0.0037 \Rightarrow$ Reject H_0.

8.3 Weekend vs. Weekday Total Duration

- H_0: Mean daily total duration is equal on weekends and weekdays.
- H_1: The means differ.

```
daily_dur = adv.groupby('DateOnly')['duration_min'].sum().reset_index(
    name='TotalDuration')
daily_dur['IsWeekend'] = adv.groupby('DateOnly')['IsWeekend'].first().
    values
dur_wknd = daily_dur.loc[daily_dur['IsWeekend']==1,'TotalDuration']
dur_wday = daily_dur.loc[daily_dur['IsWeekend']==0,'TotalDuration']
t3, p3 = stats.ttest_ind(dur_wknd, dur_wday, equal_var=False)
print(f"t = {t3:.2f}, p = {p3:.4f}")
```

Listing 6: T-test for Weekend vs. Weekday Duration

Result: $t = 1.17,\ p = 0.2447 \Rightarrow$ Fail to reject H_0.

9 Predictive Modeling

9.1 Day-of-Week Classification

A Random Forest classifier was trained using features `hour`, `DayNum`, `IsWeekend`, and `MonthNum`. Performance was moderate (40% accuracy), with confusion matrix shown in Figure 7.

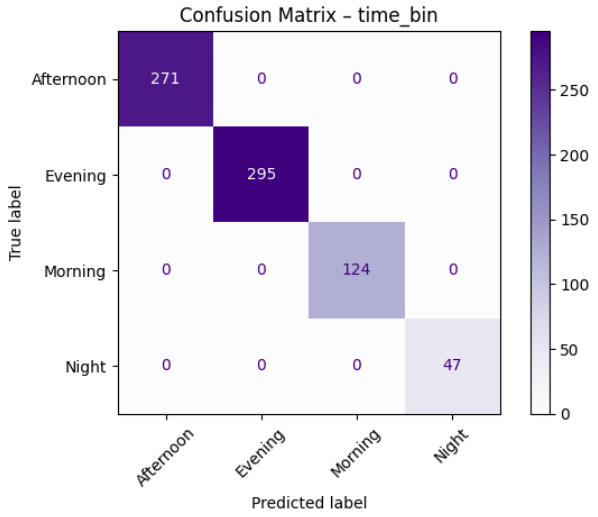

Figure 7: Confusion matrix for day-of-week classifier.

9.2 Time-Bin Classification

Classification of sessions into {Morning, Afternoon, Evening, Night} yielded perfect accuracy, as expected given the deterministic mapping from hour to bin (Figure 8).

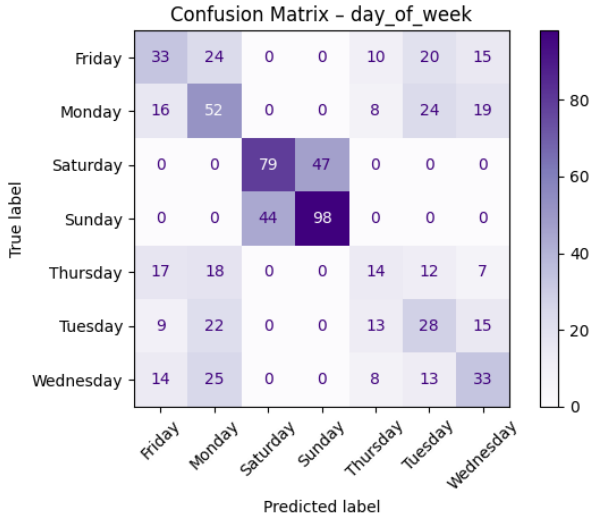

Figure 8: Confusion matrix for time-bin classifier.

10 Conclusion

The analysis of the viewing history revealed distinct seasonal and binge-watching trends, with Winter and Fall exhibiting the highest engagement and Summer the lowest. Despite a clear preference for episodic content—series accounted for over 70 % of sessions and dominated total watch minutes—daily session counts and total daily durations did not differ significantly between weekends and weekdays ($t = 1.68$, $p = 0.0943$; $t = 1.17$, $p = 0.2447$). However, binge-watching rates were found to be significantly higher in Winter than in Summer ($^2 = 8.41$, $p = 0.0037$).

Predictive modeling demonstrated that time-bin classification achieved near-perfect accuracy, validating the engineered temporal features, while day-of-week classification based on hour, day number, and weekend status yielded only moderate performance (40 %), indicating limited temporal distinctiveness of viewing days. These findings suggest that recommendation strategies should emphasize seasonal content scheduling and support binge-watching behaviors, rather than relying solely on predictable daily viewing patterns.