Streaming Service Subscription Type Analysis

Step-by-Step Guide to Identifying Ad-Supported vs Ad-Free Subscriptions

---

Step 1: Environment Setup and Project Structure

Environment Configuration

Before starting the analysis, I established a proper development environment and project structure to ensure reproducible and organized work.

Git Repository Initialization

```
# Initialize git repository and environment using uv
uv init .
```

And linked with the remote repo

Folder Structure

Created a well-organized project structure:

```
streaming-analysis/
  data/
     data.csv
  src/
     exploring.py
     gap_analysis.py
  outputs/
  logs/
  README.md
```

This structure separates data, source code, logs, and outputs for better project management and collaboration.

Step 2: Data Exploration and Initial Assessment

Loading and Basic DataFrame Information

The first step involved loading the streaming data and getting a comprehensive overview of the dataset structure.

Key Exploration Activities:

1. **DataFrame Shape and Size**: Examined the dimensions of the dataset

2. **Column Analysis**: Reviewed all available columns and their data types
3. **Sample Data Inspection**: Looked at the first few rows to understand data format

Application Column Analysis

Focused specifically on the 'application' column to identify streaming services present in the data:

- Extracted unique values from the application column
- Examined variations in naming conventions
- Identified target streaming services (Netflix, Hulu, etc.)

Step 3: Data Cleaning with Regex Pattern Matching

Streaming Service Name Standardization

Applied regex pattern matching to ensure consistent identification of streaming services despite potential variations in naming:

Regex Implementation:

- **Case-Insensitive Matching**: Used regex with `re.IGNORECASE` flag
- **Exact Word Matching**: Implemented `^word$` patterns to avoid partial matches
- **Service Validation**: Verified that target services (Netflix, Hulu) were properly identified

Target Services Analyzed:

- **Netflix**: Various case combinations (netflix, Netflix, NETFLIX)
- **Hulu**: Different capitalizations (hulu, Hulu, HULU)

This step ensured data quality and prevented misclassification due to inconsistent naming.

Step 4: Subscription Type Identification Approach

Hypothesis: Gap-Based Classification

The core hypothesis was that **viewing gaps between sessions** can indicate subscription type:

- **Ad-Supported**: Frequent short gaps (60 seconds) indicating advertisement breaks

- **Ad-Free**: Longer, irregular gaps indicating natural user behavior (pausing, breaks, etc.)

Step 5: Data Preprocessing for Gap Analysis

5.1 TV Filtering

- **Multi-Session TVs Only**: Filtered to include only TVs with multiple viewing sessions
- **Rationale**: Cannot calculate gaps with single sessions

5.2 Session Identification

- **Unique Session IDs**: Created composite identifiers using `tv_id + content_id`
- **Purpose**: Distinguish individual viewing sessions for gap calculation

5.3 Time Data Standardization

- **DateTime Conversion**: Converted start_time and end_time to pandas datetime objects
- **Data Cleaning**: Stripped whitespace and handled format inconsistencies

---

Step 6: Gap Calculation Methodology

6.1 Chronological Sorting

For each TV-content combination:

- Sorted sessions by start_time in ascending order
- Ensured proper temporal sequence for gap calculations

6.2 Gap Computation

- **Gap Formula**: `current_session_start_time - previous_session_end_time`
- **Time Units**: Converted all gaps to seconds for consistent analysis
- **Edge Cases**: Handled first sessions (no previous session) with NaN values

6.3 Data Validation

- Removed negative gaps (data quality issues)
- Filtered out unrealistic gap values
- Maintained data integrity throughout the process

Step 7: Gap Frequency Analysis

7.1 Time Binning Strategy

- **Bin Size**: 15-second intervals for granular analysis
- **Range Creation**: Dynamic binning based on maximum observed gap
- **Labeling**: Clear range labels (e.g., “0-15”, “15-30”, “30-45”)

7.2 Frequency Distribution

- **Grouping**: By TV ID and gap range
- **Counting**: Frequency of gaps in each time range per TV
- **Aggregation**: Comprehensive view of viewing patterns

Step 8: Subscription Classification Logic

8.1 Ad-Like Gap Definition

- **Threshold**: Gaps 60 seconds classified as “ad-like”
- **Rationale**: Based on typical advertisement duration research for Netflix/Hulu

8.2 Classification Criteria

Ad-Supported Subscription:

- **Minimum Ad Gaps**: 3 ad-like gaps (configurable threshold)
- **Proportion Threshold**: 60% of gaps are ad-like (configurable)
- **Logic**: Frequent short gaps indicate advertisement breaks

Ad-Free Subscription:

- **Low Ad Proportion**: <30% ad-like gaps
- **Long Gap Dominance**: More long gaps than short gaps
- **Minimal Short Gaps**: <2 ad-like gaps total
- **Logic**: Predominantly natural viewing breaks

Mixed/Uncertain:

- **Ambiguous Patterns**: Doesn’t clearly fit ad-supported or ad-free criteria
- **Edge Cases**: Unusual viewing patterns requiring manual review

Insufficient Data:

- **No Gaps**: TVs with zero calculated gaps
- **Data Quality**: Insufficient data for reliable classification

---

Step 9: Metrics and Validation

Key Metrics Calculated:

1. **Total Gaps**: Overall gap count per TV
2. **Ad-Like Gaps**: Count of gaps 60 seconds
3. **Long Gaps**: Count of gaps >60 seconds
4. **Ad Gap Proportion**: Ratio of ad-like gaps to total gaps
5. **Most Common Ranges**: Top 3 most frequent gap ranges

Validation Approaches:

- **Threshold Sensitivity**: Tested different ad_threshold and ad_frequency_threshold values
- **Manual Spot Checks**: Verified classifications for sample TVs
- **Pattern Analysis**: Examined most common gap ranges for logical consistency

---

Step 10: Results and Insights

Output DataFrame Structure:

- `tv_id`: Television identifier
- `subscription_type`: Classified subscription type
- `total_gaps`: Total number of gaps observed
- `ad_like_gaps`: Number of short gaps (60 seconds)
- `long_gaps`: Number of longer gaps (>60 seconds)
- `ad_gap_proportion`: Proportion of gaps that are ad-like
- `most_common_ranges`: Most frequent gap ranges

Business Applications:

- **Market Research**: Understanding subscription type distribution
- **Content Strategy**: Tailoring content delivery based on subscription patterns
- **User Experience**: Identifying viewing behavior differences between subscription types
- **Revenue Analysis**: Correlating subscription types with viewing patterns