Min Cheol Kim
Joseph Wu

MS&E 226 Project: Prelim Data Report

Dataset Summary

In this project, we want to investigate audience behavior in eSports by analyzing data from Twitch, one of the biggest live streaming platform dedicated to videogames. We are using the dataset “Twitch Sessions”, a parsed dataset taken from Twitch API on a public data repository hosted by Telecom Bretagne, a French research center specializing in Information Technology and telecommunication. Since the data were originally taken from Twitch and minimally processed by a reliable institution, we are fairly confident about the fidelity and completeness of the data.

“Twitch Sessions” contains a series of time-stamped data, taken every second over a day, of every streaming session occurring at that time. Each entry has 28 attributes summarizing the streaming session, including channel information, viewers, video bitrate, category, etc.

We chose a week’s worth of data from January 6, 2014 (Monday), to January 12, 2014 (Sunday). That is 7 days of data, approximately 1 million rows per day. We plan to make observations about audience behavior over the cycle of a day as well as of a week.

Our reserved test set will be same data over the next week of days, from January 13th (Monday), 2014 to January 19th (Sunday).

Variables (Continuous vs. Binary)

One variable we plan on using as a continuous response variable is the number of viewers at each streaming session. This variable ranges from the low teens to upwards of hundred thousands who tune in for different channels and sessions. We want to use these exact numbers possibly for regression modelling and prediction, so it’s best to keep it as a continuous variable.

One variables we plan on using as a binary response variable is whether a streaming session is flagged as ‘mature’. We will be using this binary variable to explore how the audience behavior and composition might change due to the nature of the content, characterized by whether the content is intended for mature audience or not.

Other continuous variables from the dataset include upload time, video bitrate, total channel views. Another binary variable to use is whether the channel is featured on Twitch. Other variables, such as geographical location, content category, will be used as categorical variables.

How we plan to use data

We plan to execute some descriptive analysis as well as predictive analysis on the "Twitch Sessions" dataset. These analyses will help us uncover what makes the big stars in eSports, what type of content the audience is drawn to, and how different factors contribute to the success of certain video game streaming channels. Below are some preliminary directions we plan to explore with the dataset.

Descriptive Analysis:

- How does total audience fluctuate throughout a day? And throughout a week?

- How do sessions differ with respect to whether they are for mature audience? Whether they are featured on Twitch?

- What do the most successful channels do differently from channels that underperform?

- How do videos of different categories perform on Twitch?

Predictive Analysis:

- Can we predict # of viewers who tune in on a channel based on attributes besides their channel name (upload time, category, feature, etc.)? What type of regression modeling is best suited for this prediction?

- Are there ways to cluster the sessions in an intuitive way?

Preliminary Data Exploration Results

- The only N/A or none values are found in the "geo" column, representing the country of the streaming. We will not use the rows that contain the "none" values when we are using the geo column.

- There are several columns that are seemingly irrelevant, such as the ones generated by the Twitch system and are arbitrary. One example is the channel id. The channel name provides a categorical variable we can work with, but the id is simply an arbitrary and unique number assigned to each channel. The audio and the video codec also seem to be irrelevant columns.

- A cursory analysis from covariate pairs plots and our intuition led to the analysis of the following covariate associates: Mature vs Featured, Mature vs Viewers, Channel views vs Viewers and Featured vs. Viewers.

- We found that the sessions that were flagged as “mature” on average had a greater number of maximum viewers (the peak number of viewers on a given session). The mean for “mature” flagged sessions was 25 viewers while that of the non-mature sessions was about 14 viewers.

- There seemed to be an association between the mature and the featured flag. Out of the 185878 that were featured, around 28% or 51780 sessions were flagged as mature.

- As expected, the sessions that were featured had, on average, several times the amount of maximum viewer traffic compared to the non-featured sessions. This is also interesting since the “mature” and the “featured” flags both are associated with increased view count, but they themselves are inversely associated.

- As we continue to explore and analyze the data, several interaction variables could be added. A combination of “mature” and “featured” flags could be added; for example, we could have a covariate that indicates whether the session is both “mature” and “featured.” One other interaction term we can foresee now is with video bitrate and video uptime, since they both concern the quality of the streaming experience.

- One population model we can suggest for predicting the number of viewers is a linear combination of all of our factors and all of their interaction terms/variables. This is certainly dependent on the fact that we are first attempting a prediction problem. For example, we would not intuitively believe that having a high channel view would somehow make the content more interesting and therefore attract more viewers; however, someone who makes interesting sessions probably always have a high number of session viewers as well as a large number of channel viewers.

- The code used for the preliminary analysis above can be found on the following pages.

Code / Output

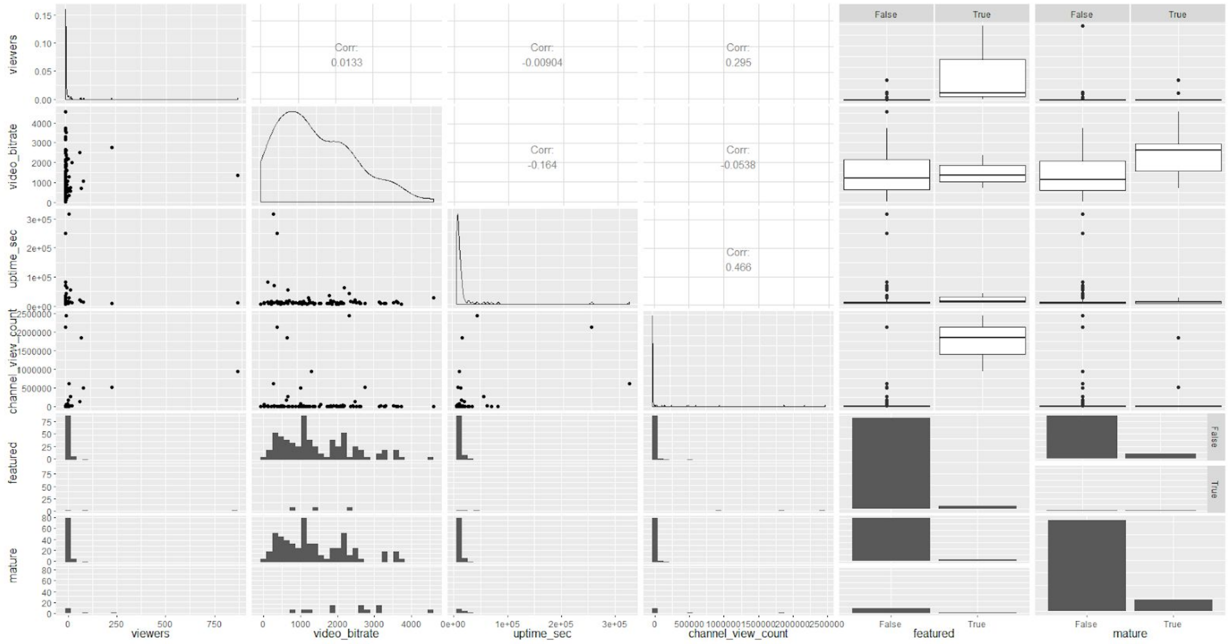
```
library(tidyverse)
library(GGally)

# Read and clean data -----------------------------------------------------------

load("C:/Users/minch_000/Desktop/226 project/.RData")

stream_data <- select(week_data, -accessed_at_utc, -channel_login, -video_height,
                      -video_width, -embed_count, -site_count, -audio_codec,
                      -video_codec, -session_count)
levels(stream_data$mature) <- c("False", "True")
stream_data$viewers <- as.numeric(as.character(stream_data$viewers))

# Finding associated covariates
assoc_data <- select(stream_data, viewers, video_bitrate, uptime_sec, channel_view_count,
featured, mature)
sample_assoc_data <- assoc_data[sample(nrow(assoc_data), 100),]
ggpairs(sample_assoc_data)
```



```
# Group by session for future uses
by_session <- group_by(stream_data, session_id)

# Find out how having a "Mature" flag correlates with max # of viewers
# of a given session.
by_mature <- group_by(stream_data, session_id) %>%
         summarise(
           max_viewer = max(viewers),
           mature = mature[1]
         ) %>%
         filter(
           !is.na(max_viewer),
           !is.na(mature)
         ) %>%
         group_by(mature) %>%
         summarise(
           max_viewer_avg = mean(max_viewer)
         )
by_mature

# A tibble: 2 × 2
  mature max_viewer_avg
  <fctr>          <dbl>
1  False       13.87001
2   True       25.28092

# Association between the mature and the featured flag
by_feature_mature <-  select(by_session,
              featured,
              mature
            ) %>%
            filter(featured == 'True') %>% # only consider featured sessions
            group_by(mature) %>%
            summarise(
              count = n()
            )
by_feature_mature

# A tibble: 2 × 2
  mature  count
  <fctr>  <int>
1  False 134098
2   True  51780

# Find out how having a "Featured" flag correlates with max # of viewers
# of a given session.
by_featured <- group_by(stream_data, session_id, featured) %>%
  summarise(
    max_viewer = max(viewers)
  ) %>%
  filter(
    !is.na(max_viewer),
    !is.na(featured),
    featured == 'False' | featured == 'True'
  ) %>%
  group_by(featured) %>%
  summarise(
    max_viewer_avg = mean(max_viewer)
  )
by_featured

# A tibble: 2 × 2
  featured max_viewer_avg
    <fctr>          <dbl>
1    False       8.734332
2     True     538.790890

# There were a total of 134098 + 51780 = 185878 sessions that were featured.
# Out of these, 51780, or 28%
```