STA 141 Final Project Report: Analysis of TikTok Song Virality

Alex Davis (alfdavis@ucdavis.edu)

Rue Ekpemiro (orekpemiro@ucdavis.edu)

Gunica Sharma (gsksharma@ucdavis.edu)

Himmat Toor (hxtoor@ucdavis.edu)

Siyuan Yang (siyyang@ucdavis.edu)

Professor: Akira Horiguchi

June 11, 2025

I Executive Summary

This project attempts to quantify the relationship between the features of a viral song and its popularity on TikTok. We sought to answer the questions: What audio features significantly affect a song's popularity on TikTok? As well as how does the time of a song's release and the artist affect its popularity on the app? To answer these questions we used logistic regression and three-way ANOVA modeling to fit our data and come to our conclusions, along with statistical graphs in our analysis.

Our analysis was based on the dataset of trending TikTok tracks that focused on song popularity with many different music descriptors such as tempo, valence and danceability. In order to thoroughly examine virality we employed a methodical approach, first we conducted exploratory data analysis where we visualized the data to understand the distribution and relationships of the different audio features. We also did a correlation matrix analysis and a Principle Component Analysis (PCA) to identify relevant variables and mitigate multicollinearity. We did our quantitative modeling on the basis of logistic regression to predict the trending status based on audio features. We also did a three way ANOVA to assess the impact of release timing on song popularity.

Our logistic regression model revealed insights into the audio characteristics that predispose songs that trend on TikTok, we found that higher loudness, increased acousticness, and low levels of energy, danceability and speechiness were more likely to achieve trending status. Our exploratory data analysis suggested that audiences favor lively, upbeat and dance friendly tracks. The PCA helped in selecting key predictors. Further, we found that the timing for when a song is released is crucial, with the month, season, and weekday being significant factors in how popular a song becomes on TikTok regarding our ANOVA model. The interactions between these variables were also highly significant. Songs released in certain seasons had higher virality, additionally the day they were released played a crucial role, these findings were supported both by our ANOVA and data plots. When looking at the artist data we also found patterns with consistent virality. Some artists displayed a balance of trending and nontrending tracks, others showed a higher proportion of trending songs relative to their total output. We also found that while some artists released lots of songs and few trends, others had fewer releases and overall more trendy songs. This suggests that for consistent virality, quality and artistry play a more critical role than the sheer number of tracks released.

Despite these findings we acknowledge certain limitations in our analysis and data. The data was specifically from southeast Asia and the Philippines meaning our conclusions may not represent the global TikTok user base. Our dataset also lacked accurate genre classifications, artist popularity metrics and information about sponsored tracks and promotions which likely also had a role in virality. Since our variables were predefined in the dataset the accuracy can also be questioned.

Overall, our project successfully quantified several factors in TikTok song virality providing a deeper understanding of the platform and modern music trends. We cannot claim a direct causal relationship due to data limitations but our results shed light into preferences for general audiences today. These findings are a valuable contribution to the ongoing effect to understanding virality in the digital music and social media landscape.

II Background

The purpose of the project is to explore how different kinds of music affect their virality on TikTok. Within these past few years this short form social media app has emerged as a dominant force in the music industry as it shapes music trends and influences listeners behaviors. Due to its widespread popularity, songs that gain traction often end up increasing in streams on music streaming platforms such as Spotify and Youtube. There seems to be a powerful connection between social media virality and the popularity of current music streaming.

Despite the growing role TikTok plays there is still a lack of quantitative understanding about what kinds of factors influence a song's virality. Some songs achieve unexpected popularity, while others are rediscovered from the past and garner renewed attention. This phenomenon can raise an important question for artists, producers and marketers: what makes a song trend on TikTok? Does it depend on the duration, tempo, danceability? Or is it simply dependent on the artist and release date? These are the different kinds of questions that this project aims to answer.

This project aims to bridge that gap by analyzing a dataset of TikTok trending tracks in order to uncover patterns and predictors of virality. The goal is to identify what kinds of song attributes correlate most with TikTok success. We want to conduct initial analyses to determine which specific factors are most correlated with each other then move onto creating a logistic regression model in order to predict what kinds of songs will become viral based on different factors.

III Data Description

This dataset was developed to explore the **popularity of trending TikTok tracks** and identify the factors that contribute to a track's success on the platform. It was created by **Team Dan**, a participant in **Eskwelab's DSCN Sprint 2 Final** (GitHub Repository).

The data was sourced from **Kaggle**, though the funding source remains unknown. It focuses specifically on tracks popular in **Southeast Asia and the Philippines**.

```
tiktok <- read.csv("tiktok.csv")
dim(tiktok)
```

```
## [1] 6746   23
```

```
# remove repeat
tiktok_clean <- tiktok %>% distinct(track_id, .keep_all = TRUE)
dim(tiktok_clean)
```

```
## [1] 3560   23
```

```
# trending indicator
tiktok_clean <- tiktok_clean %>%
  mutate(trending = ifelse(popularity > 75, 1, 0))
tiktok_clean <- tiktok_clean %>%
  select(-c(genre, playlist_id, playlist_name, duration_mins))
```

Before analyzing the dataset, we removed any observations with null values. We also dropped any non-numeric variables such as genre, duration_mins, playlist_id, and playlist_name, as they are not relevant to our research questions. Then we created a binary indicator, trending, labeling songs with popularity above 75 as trending (1) and others as non-trending (0). The dataset contained 6,746 instances of tracks. The dataset has 3,560 unique instances of tracks. Each observation in the dataset is a TikTok track. Below is a description of the columns.

Table 1: Variable Descriptions

Variable	Description
artist_name	Name of the artist
artist_id	Spotify ID of the artist
album_id	Spotify ID of the album
acousticness	Measures how acoustic a track is (0.0 – 1.0)
danceability	Suitability for dancing (0.0 = least, 1.0 = most)
duration_mins	Duration of the track in minutes
energy	Energy level of a track (0.0 = calm, 1.0 = energetic)
instrumentalness	Likelihood the track is instrumental (closer to 1.0 = more instrumental)
key	Key of the track (0 = C, 1 = C , 2 = D, etc.)
liveness	Probability track was recorded live (> 0.5 = live)
loudness	Overall loudness in decibels (closer to 0 = louder)
mode	Modality (0 = minor, 1 = major); no clear pattern found
popularity	Popularity score (0 to 100)
speechiness	Degree of spoken words (0.0 = music, 1.0 = speech)
track_name	Name of the track
tempo	Estimated beats per minute (BPM)
valence	Musical positiveness (0.0 = sad, 1.0 = happy)

IV. Exploratory Data Analysis

IV.1 Summary Statistics

To gain a clearer understanding of the dataset's characteristics, we computed the summary statistics for all numeric audio features and relevant metadata.

```
# Summary Statistics
stats_df <- tiktok_clean %>%
  summarise(
    across(
      c(danceability, energy, loudness, speechiness,
        acousticness, instrumentalness, liveness, valence, tempo, duration),
      list(
        Mean   = ~round(mean(.   , na.rm=TRUE), 3),
        Median = ~round(median(. , na.rm=TRUE), 3),
        SD     = ~round(sd(.     , na.rm=TRUE), 3)
      )
```

```
    )
  ) %>%
  pivot_longer(
    cols      = everything(),
    names_to  = c("Variable", "Statistic"),
    names_sep = "_"
  ) %>%
  pivot_wider(
    names_from  = Statistic,
    values_from = value
  )

kable(
  stats_df,
  col.names = c("Variable", "Mean", "Median", "SD"),
  caption   = "Summary Statistics for Model Features",
  booktabs  = TRUE
)
```

Table 2: Summary Statistics for Model Features

Variable	Mean	Median	SD
danceability	0.728	0.743	0.139
energy	0.637	0.643	0.181
loudness	-6.967	-6.460	2.997
speechiness	0.136	0.080	0.128
acousticness	0.211	0.119	0.236
instrumentalness	0.040	0.000	0.152
liveness	0.181	0.124	0.137
valence	0.538	0.537	0.236
tempo	120.992	122.013	25.874
duration	194770.595	186088.000	60734.615

We observe that:

- **Danceability** scores are generally high, with a median of 0.743 and a mean of 0.7276, suggesting that most songs are well-suited for dance-oriented content.
- **Energy** also shows a high central tendency (median 0.643, mean 0.637), indicating that TikTok's popular tracks tend to be lively and upbeat.
- **Loudness** has a median of -6.46 dB, with most tracks ranging between -15 and -5 dB. This suggests a balance between clarity and energetic production in TikTok tracks.
- **Tempo** typically lies around 120 BPM (median 122.01 BPM), ideal for short, catchy videos.
- **Valence** (musical positiveness) has a mean of 0.537, implying that many TikTok songs are perceived as moderately positive and upbeat.

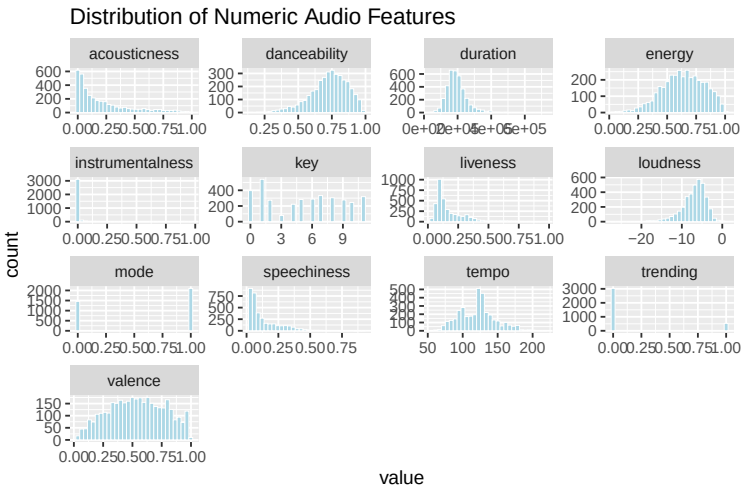
Other features, such as acousticness, instrumentalness, and liveness, tend to be skewed toward lower values. This reflects the predominance of studio-produced and vocal-heavy tracks in the dataset.

In addition to these core audio features, categorical variables like key, mode, and speechiness were also summarized, but since they do not exhibit clear central trends for audio content, they are used primarily as contextual descriptors in later analysis.

IV.2 Visual Exploration of Audio Features

To better understand the distribution of our numeric audio features, we created a set of histograms and boxplots.

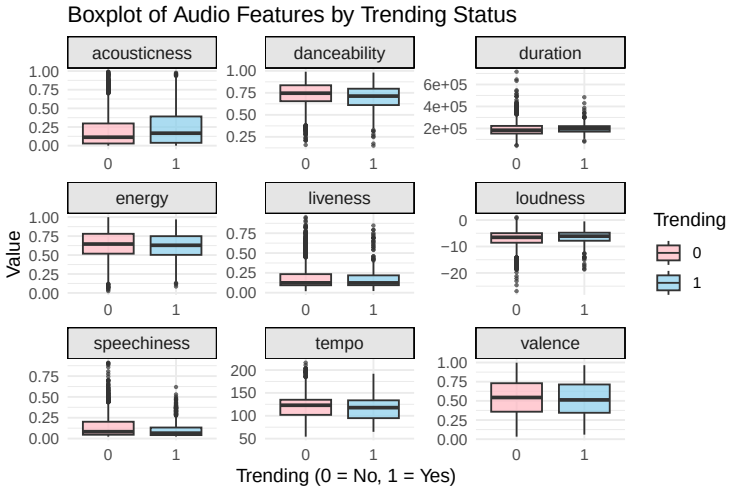
```
# Histograms for all features
num_vars <- tiktok_clean %>%
  select(where(is.numeric)) %>%
  select(-popularity)
num_vars %>%
  pivot_longer(cols = everything()) %>%
  ggplot(aes(value)) +
  geom_histogram(bins = 30, fill = "lightblue", color = "white", linewidth = 0.2) +
  facet_wrap(~name, scales = "free") +
  labs(title = "Distribution of Numeric Audio Features")
```



First, we examined overall distribution trends by generating histograms for all numeric audio features. Most features, such as danceability, energy, and valence, skew towards higher values, suggesting TikTok's general preference for lively, upbeat tracks. Features like acousticness and instrumentalness show a strong left skew, highlighting the dominance of electronic and vocal-heavy tracks. Meanwhile, features such as mode, key, and speechiness show little variation and appear less relevant for TikTok's short-video context.

```
# Boxplot for some features
audio_features <- tiktok_clean %>%
```

```
  select(trending, danceability, duration, energy, loudness, tempo, valence,
         acousticness, liveness, speechiness)
audio_long <- audio_features %>%
  pivot_longer(cols = -trending, names_to = "feature", values_to = "value")
ggplot(audio_long, aes(x = factor(trending), y = value, fill = factor(trending))) +
  geom_boxplot(alpha = 0.7, outlier.size = 0.5) +
  facet_wrap(~feature, scales = "free", ncol = 3) +
  labs(title = "Boxplot of Audio Features by Trending Status",
       x = "Trending (0 = No, 1 = Yes)", y = "Value") +
  scale_fill_manual(values = c("lightpink", "skyblue"), name = "Trending") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 0),
        strip.background = element_rect(fill = "gray90"),
        strip.text = element_text(size = 10))
```



Next, we compared trending versus non-trending songs using boxplots. We found that trending songs have slightly higher danceability and energy, indicating their importance for capturing user attention. Loudness also shows a tighter interquartile range in trending songs, suggesting consistent production volume. Valence—reflecting musical positiveness—is also somewhat higher in trending songs, aligning with TikTok's preference for catchy, positive tracks. However, features like key, mode, and speechiness show minimal differences, suggesting they may be less crucial for explaining a song's virality.

Based on this visual exploration, we decided to focus our subsequent analysis on the following key audio features: **danceability, energy, loudness, tempo, and valence**. These features not only show distributional patterns but also align well with the energetic and rhythmic qualities that are crucial for viral content on TikTok.

V. Methodology

For data cleaning and preprocessing purposes we will utilize a correlation matrix and PCA analysis to evaluate which variables contribute the most to the data set, and remove unnecessary variables

to reduce noise and redundant information from the data set.

To explore what makes a song viral on TikTok, we used logistic regression and three-way ANOVA. Logistic regression was chosen because it models binary outcomes—allowing us to classify songs as trending or not—based on audio features. A correlation matrix was used to check for multi-collinearity and guide feature selection. We used three-way ANOVA to analyze how categorical factors like release season, month, and day impact a song's popularity score. ANOVA is ideal here, as it evaluates the effects and interactions of multiple categorical variables on a numerical outcome.

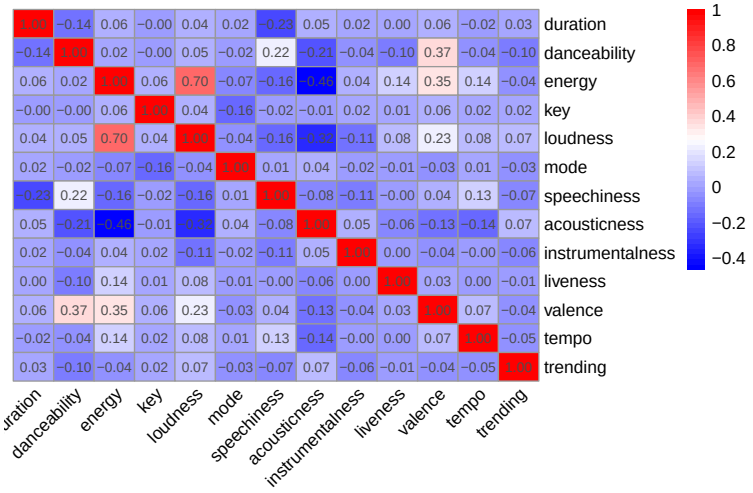
To measure the virality of a song we add a new column called trending in place of popularity. Trending is then just a logical variable where the value of 1 denotes popularity greater than 75 and the value of 0 denotes popularity less than equal to 75. The variable popularity is dropped from the dataset. A cut off 75 is deemed appropriate as it adequately distributes trendiness where only 14.9% of the songs in the dataset are considered trending.

VI. Analysis

VI.1 Correlation matrix

The variables - track_id, track_name, artist_id, artist_name, album_id, release_date, playlist_id, playlist_name and genre are all dropped from the dataset as they are non-numeric, which makes them non-essential for our prediction of virality. By separating out the numeric variable from the dataset we check for correlation of each variable on trendiness by making a correlation matrix.

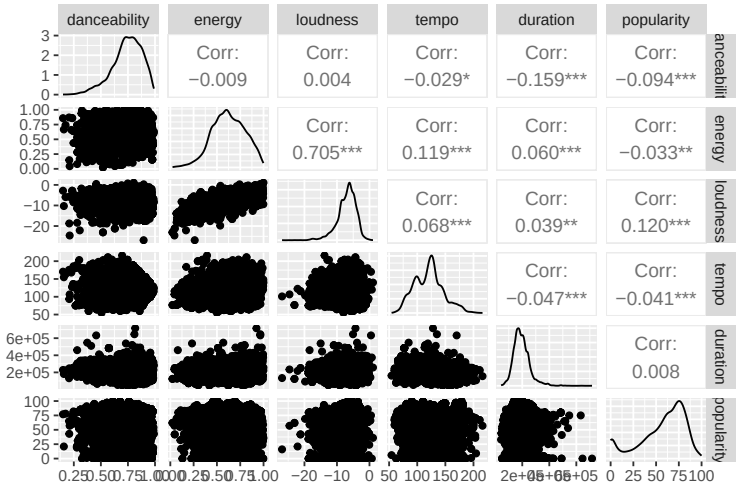
```
# Correlation matrix
num_vars <- num_vars %>% select(-trending, trending)
corr_matrix <- cor(num_vars, use = "complete.obs")
pheatmap(corr_matrix, color = colorRampPalette(c("blue", "white", "red"))(50),
         cluster_rows = FALSE, cluster_cols = FALSE,
         display_numbers = TRUE, fontsize = 10, angle_col = 45)
```



We drop the numeric variables – duration, key, mode, liveness, and duration_mins – as they have low correlation coefficients ($|r| < 0.4$) with trending status. Among the remaining features – danceability, energy, loudness, speechiness, acousticness, instrumentalness, valence, and tempo – several pairs exhibit notably high correlations with each other. Energy is strongly correlated with loudness (0.702), and moderately with valence (0.349) and danceability (0.370). Speechiness and danceability show a weaker correlation (0.225), while valence is also correlated with loudness (0.227). These patterns suggest potential multicollinearity, especially among energy, loudness, and valence. All eight features are retained for now, but caution is needed when interpreting individual model coefficients.

VI.2 Scatter plots

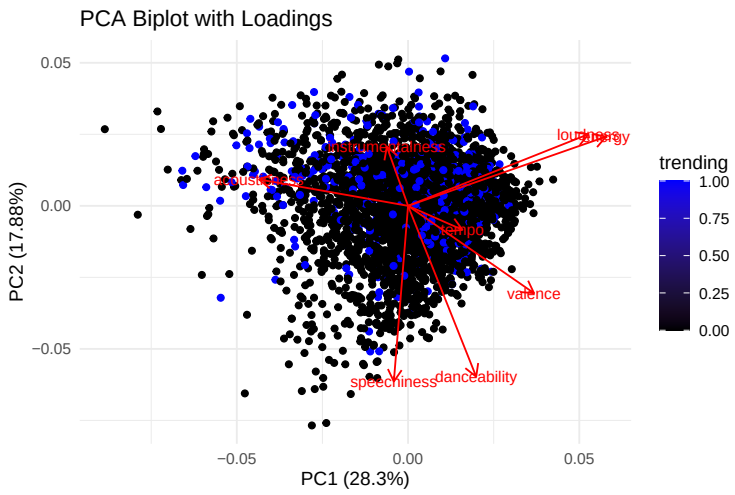
```
# Scatter plots
tiktok %>%
  select(danceability, energy, loudness, tempo, duration, popularity) %>%
  ggpairs()
```



The figure above shows scatter plots and density curves for the eight audio features and popularity. On the diagonal, danceability is skewed high, energy and valence are centered, loudness is left-skewed, tempo clusters around 120 BPM, and popularity peaks near 80. From the scatter plots, none of the features shows a strong linear relationship with popularity. Loudness and energy have a mild positive trend, tempo shows a slight negative trend, and other features have little or no visible trend—confirming their weak correlation with trending status. However, some features are clearly correlated with each other. Energy and loudness form a strong upward pattern, valence and danceability show a positive cloud, and speechiness and danceability also trend upward. These patterns suggest that while individual features do not explain popularity well, there is overlap between predictors. To address this, we apply Principal Component Analysis (PCA) to reduce dimensionality and remove redundancy.

VI.3 PCA Interpretation

```
# Selecting key numeric features for PCA
pca_features <- c("danceability", "energy", "loudness", "speechiness",
                  "acousticness", "instrumentalness", "valence", "tempo")
pca_data <- subset(tiktok_clean, select = pca_features)
pca_result <- prcomp(pca_data, scale. = TRUE)

# Visualization of PCA
autoplot(pca_result,
         data = tiktok_clean,
         colour = 'trending',
         loadings = TRUE,
         loadings.colour = 'red',
         loadings.label = TRUE,
         loadings.label.size = 3) +
  scale_color_gradient(low = "black", high = "blue") +
  theme_minimal() +
  labs(title = "PCA Biplot with Loadings")
```



The PCA test indicates that *tempo* points in the same direction as *valence* but contributes less to the principal components. Loudness and energy align with equal magnitude. Although energy and loudness are highly correlated, energy was retained because of its uniquely high loading. Other features diverge in direction but maintain strong contributions. Consequently, we drop *tempo* and retain *danceability*, *energy*, *loudness*, *speechiness*, *acousticness*, *instrumentalness*, and *valence* for logistic regression.

VI.4 Logistic Regression for Prediction

Based on the correlation analysis, we retained seven audio features and applied logistic regression to examine their association with trending status.

```r
df_model <- tiktok_clean %>%
  select(trending, danceability, energy, loudness,
         speechiness, acousticness, instrumentalness, valence)
df_model$trending <- as.factor(df_model$trending)

# Train-test split
set.seed(123)
trainIndex <- createDataPartition(df_model$trending, p=0.7, list=FALSE)
train <- df_model[trainIndex, ]
test <- df_model[-trainIndex, ]

# Fit model
model <- glm(trending ~ ., data = train, family = binomial)

# Extract and format coefficients
coefs <- tidy(model) %>%
  mutate(
    estimate    = round(estimate, 4),
    std.error   = round(std.error, 4),
    z.value     = round(statistic, 4),
    p.value     = sprintf("%.4f", p.value)
  ) %>%
  rename(
    Term        = term,
    Estimate    = estimate,
    `Std. Error`= std.error,
    `z value`   = z.value,
    `Pr(>|z|)`  = p.value
  ) %>%
  select(Term, Estimate, `Std. Error`, `z value`, `Pr(>|z|)`)

kable(coefs, booktabs = TRUE, caption = "Logistic Regression Coefficients")
```

Table 3: Logistic Regression Coefficients

Term	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	2.4078	0.5925	4.0635	0.0000
danceability	-1.4021	0.4589	-3.0553	0.0022
energy	-2.8341	0.5224	-5.4247	0.0000
loudness	0.1968	0.0302	6.5043	0.0000
speechiness	-1.4099	0.5399	-2.6114	0.0090
acousticness	0.5641	0.2724	2.0707	0.0384
instrumentalness	-1.3106	0.5821	-2.2515	0.0244
valence	0.0864	0.2916	0.2964	0.7669

The model identified several statistically significant predictors. Higher loudness and acousticness

increase the chance of trending, while higher energy, danceability, speechiness, and instrumentalness are associated with a lower chance. Valence was not statistically significant. The table below summarizes the model estimates.

```
# prediction
test <- test %>%
  mutate(
    predicted_prob  = predict(model, newdata = ., type = "response"),
    predicted_class = ifelse(predicted_prob > 0.5, 1, 0)
  )

test$predicted_class <- factor(test$predicted_class, levels = c(0, 1))
test$trending        <- factor(test$trending,        levels = c(0, 1))

# confusing matrix
cm <- confusionMatrix(test$predicted_class, test$trending, positive = "1")

# key performance indicators
metrics_tbl <- data.frame(
  Metric = c("Accuracy", "Sensitivity (Recall)", "Specificity"),
  Value  = c(round(cm$overall["Accuracy"], 4),
             round(cm$byClass["Sensitivity"], 4),
             round(cm$byClass["Specificity"], 4))
)

# table
cm_table_df <- as.data.frame.matrix(cm$table)
cm_table_df <- cbind(Actual = c("Actual 0", "Actual 1"), cm_table_df)
kable(cm_table_df,
      caption = "Confusion Matrix (Cutoff = 0.5)",
      col.names = c("Actual", "Pred 0", "Pred 1"),
      booktabs = TRUE)
```

Table 4: Confusion Matrix (Cutoff = 0.5)

	Actual	Pred 0	Pred 1
0	Actual 0	908	159
1	Actual 1	1	0

```
kable(metrics_tbl,
      caption = "Key Performance Indicators (Cutoff = 0.5)",
      booktabs = TRUE,
      row.names = FALSE)
```

Table 5: Key Performance Indicators (Cutoff = 0.5)

Metric	Value
Accuracy	0.8502
Sensitivity (Recall)	0.0000
Specificity	0.9989

Using a threshold of 0.5, the model achieved an overall accuracy of 85.02%. However, it failed to correctly identify any trending songs, resulting in a recall of 0. As shown in the confusion matrix, 908 out of 909 non-trending songs were classified correctly, while all 160 trending cases were misclassified.

```
# Try different cutoffs
cutoffs <- c(0.5, 0.4, 0.3, 0.25, 0.2, 0.15)
cutoff_results <- lapply(cutoffs, function(cut) {
  predicted_class <- ifelse(test$predicted_prob > cut, 1, 0)
  predicted_class <- factor(predicted_class, levels = c(0, 1))

  cm_cut <- confusionMatrix(predicted_class, test$trending, positive = "1")

  data.frame(
    Cutoff   = cut,
    Accuracy = round(cm_cut$overall["Accuracy"], 4),
    Recall   = round(cm_cut$byClass["Sensitivity"], 4)
  )
}) %>% bind_rows()

kable(cutoff_results,
      caption = "Model Performance at Different Cutoff Thresholds",
      booktabs = TRUE,
      row.names = FALSE)
```

Table 6: Model Performance at Different Cutoff Thresholds

Cutoff	Accuracy	Recall
0.50	0.8502	0.0000
0.40	0.8464	0.0189
0.30	0.8361	0.0943
0.25	0.8052	0.1384
0.20	0.7463	0.2516
0.15	0.6124	0.5786

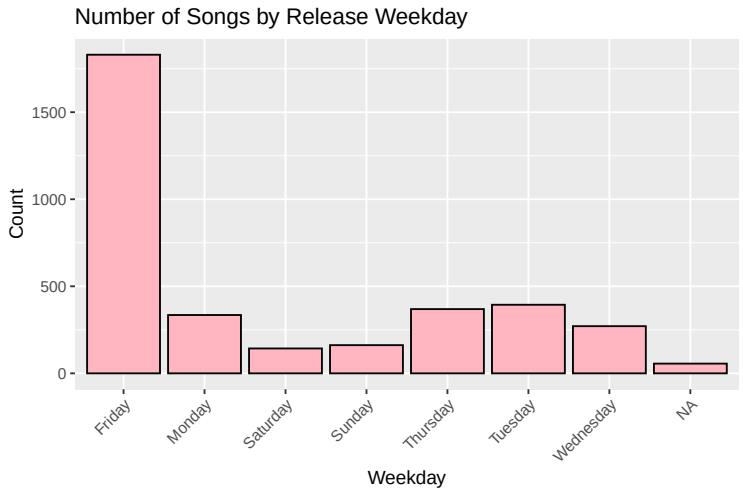
To improve the model's ability to detect trending songs, we tested a range of lower probability cutoffs. As the cutoff decreases, recall improves but accuracy drops. A cutoff of 0.2 yields a recall of 25.2% while maintaining reasonable accuracy (74.63%), making it a good balance for our use case. The table below summarizes the trade-offs.

The logistic model reveals important feature-trend relationships, but struggles with prediction due to class imbalance. We recommend using a cutoff of 0.2 for better recall, and suggest exploring resampling or alternative models in future work.

VI.4 Release Data

To determine if different release times play a role on whether something is trending (>0.75) we compared release weekdays, seasons, months and release years. The graph for the trending by release weekday compares the different days of the week songs were released and if their respective popularity, it can be seen that Sunday, Monday, Wednesday and Saturday had the lowest popularity rates whereas Tuesday, Thursday and Friday the highest.

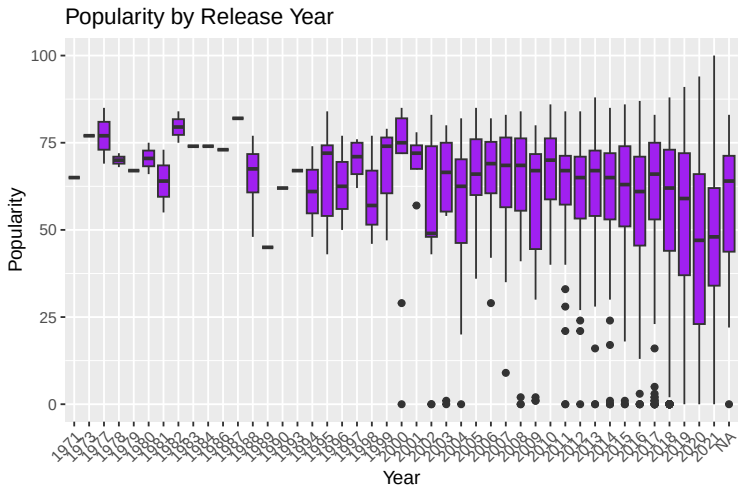
```
# Set year, month, weekday
tiktok_clean$release_date <- as.Date(tiktok_clean$release_date)
tiktok_clean <- tiktok_clean %>%
  mutate(release_year = format(release_date, "%Y"),
         release_month = format(release_date, "%m"),
         release_weekday = weekdays(release_date))
ggplot(tiktok_clean, aes(x = release_weekday)) +
  geom_bar(fill = "lightpink", color = "black") +
  labs(title = "Number of Songs by Release Weekday",
       x = "Weekday", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



We first analyzed the released songs by the weekday and found that Friday dominates as the most common release day by far which suggests that it's an industry standard. Other days had much lower releases likely because of Friday releases for weekend streams.

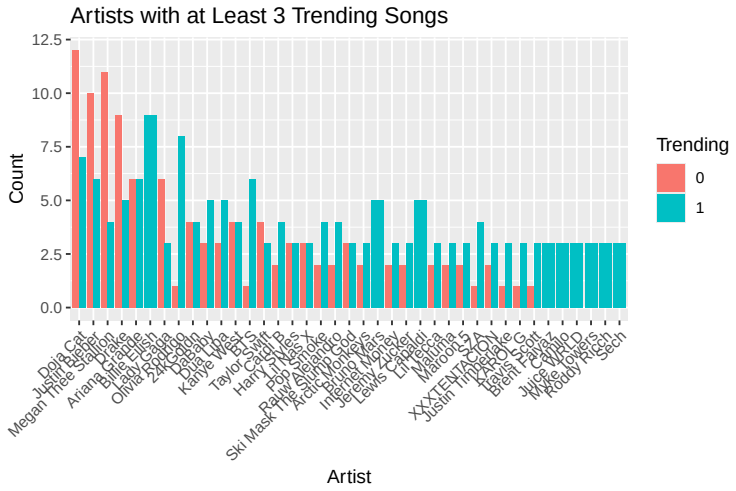
```
ggplot(tiktok_clean, aes(x = release_year, y = popularity)) +
  geom_boxplot(fill = "purple") +
  labs(title = "Popularity by Release Year",
```

```
       x = "Year", y = "Popularity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



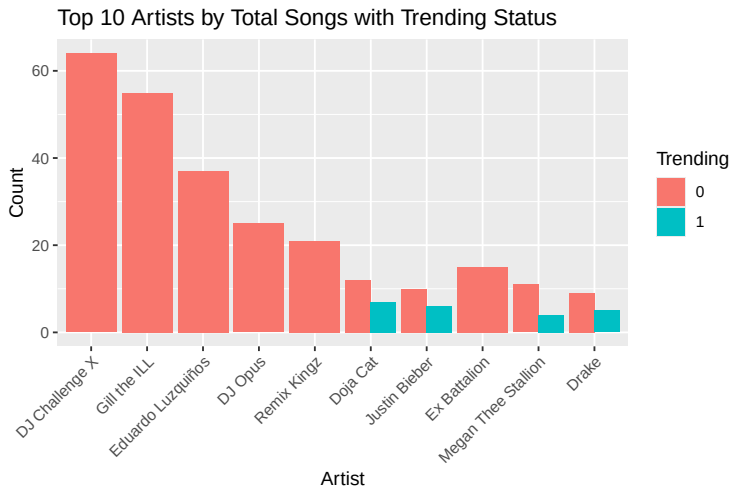
Looking at popularity by the release year we can see that the box plots show that earlier songs pre-2000 show less variability and higher average popularity likely due to a smaller sample size. From 2015 onwards there is widespread popularity indicating both hits and flops which reflects the competitive and diverse nature of modern releases.

```
# the number of trending topics for each artist and filter for >=3
artist_trend_ratio <- tiktok_clean %>%
  group_by(artist_name) %>%
  summarise(total_songs   = n(),
            trending_songs = sum(trending)) %>%
  filter(trending_songs >= 3) %>%
  arrange(desc(trending_songs))

# plot
ggplot(
  tiktok_clean %>% filter(artist_name %in% artist_trend_ratio$artist_name),
  aes(x = fct_infreq(artist_name), fill = factor(trending))
) +
  geom_bar(position = "dodge") +
  labs(title = "Artists with at Least 3 Trending Songs",
       x = "Artist", y = "Count", fill = "Trending") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



We next analyzed the artists with at least 3 trending songs, separating the counts by the trending status. Some artists like Doja Cat and Megan Thee Stallion show a relatively balanced mix of trending tracks whereas others such as BTS and Billie Elish have a higher proportion of trending tracks.

```
# top 10 Artists
top_artists <- tiktok_clean %>%
  count(artist_name, sort = TRUE) %>%
  slice_head(n = 10) %>%
  pull(artist_name)

# plot
ggplot(
  tiktok_clean %>% filter(artist_name %in% top_artists),
  aes(x = fct_infreq(artist_name), fill = factor(trending))
) +
  geom_bar(position = "dodge") +
  labs(title = "Top 10 Artists by Total Songs with Trending Status",
       x = "Artist", y = "Count", fill = "Trending") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



When looking at the total top 10 artists based on the number of songs they released. We can see that some artists like DJ Challenge and Gill the ILL have high volumes of songs but very few of them are trending. In contrast artists like Doja Cat and Megan Thee Stallion have fewer songs overall but a higher proportion of them are popular suggesting quality over quantity when it comes to virality.

VI.5 ANOVA

```
tiktok_clean <- tiktok_clean %>%
  mutate(
    release_month  = factor(format(as.Date(release_date), "%m")),
    release_season = case_when(
      release_month %in% c("12","01","02") ~ "Winter",
      release_month %in% c("03","04","05") ~ "Spring",
      release_month %in% c("06","07","08") ~ "Summer",
      release_month %in% c("09","10","11") ~ "Fall",
      TRUE ~ NA_character_
    ),
    release_season = factor(release_season, levels=c("Winter","Spring","Summer","Fall"))
  )
tiktok_clean <- tiktok_clean %>%
  mutate(release_month = factor(release_month))
aov_model2 <- aov(popularity ~ release_weekday * release_season * release_month,
                  data = tiktok_clean)
anova_df <- broom::tidy(aov_model2)
anova_df %>%
  select(term, df = df, `Sum Sq` = sumsq, `Mean Sq` = meansq,
         `F value` = statistic, `Pr(>F)` = p.value) %>%
  mutate(
    `Sum Sq`   = round(`Sum Sq`,   4),
    `Mean Sq`  = round(`Mean Sq`,  4),
```

```
    `F value`  = round(`F value`,  4),
    `Pr(>F)`   = ifelse(is.na(`Pr(>F)`), NA, sprintf("%.4f", `Pr(>F)`))
  ) %>%
  arrange(!term %in% "Residuals", term) %>%
  kable(format = "markdown", col.names = c(
    "", "Df", "Sum Sq", "Mean Sq", "F value", "Pr(>F)"
  ))
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Residuals	3420	1993888.971	583.0085	NA	NA
release_month	8	12744.457	1593.0571	2.7325	0.0053
release_season	3	9105.524	3035.1746	5.2061	0.0014
release_weekday	6	51564.095	8594.0158	14.7408	0.0000
release_weekday:release_month	48	55032.002	1146.5000	1.9665	0.0001
release_weekday:release_season	18	33173.703	1842.9835	3.1612	0.0000

This ANOVA table analyzes how different aspects of a song such as weekday, timing and month affect its popularity on TikTok. The results show that the release weekday had the strongest effect on popularity as it has the lowest p-value followed by season and month. This indicates that when a song is released significantly impacts its chances of trending. Additionally, there are significant interaction effects between weekday and season and weekday and month, so release timing varies depending on a combination of these different variables. However the residual variance is high so most of the variation is likely dependent on factors such as audio features.

VII Conclusions

From our analysis, we discovered that the popularity of a TikTok track largely depends on its release date, as well as a combination of audio features including danceability, energy, acousticness, and loudness.

Based on our logistic regression model, we found that songs that are louder, more acoustic, and less energetic, danceable, or speech-heavy are more likely to be trending. We also found that factors such as tempo, duration, liveness, and valence were not statistically significant, meaning that they likely do not contribute much to the trending status. However, as noted from the prediction of the logistic regression model for a standard cutoff of 0.5, the model rarely predicts a true positive due to an imbalance caused by a much higher proportion of non-trending songs in the dataset. Decreasing this cutoff value to 0.2 makes it suitable for a disproportionate dataset, and gives a better result of recall at a trade-off of a slight decrease in accuracy.

Our visual analysis and three-way ANOVA results show that release timing significantly affects a song's popularity. Season, month, and day of release were all significant predictors, with interactions simply indicating that a song's release day depends on the month and season. Most trending songs were released in the first half of the year and on Fridays—likely benefiting from longer exposure and increased weekend streaming.

Due to the limitations of our data, we cannot claim a direct causal relationship between a song's audio features and release timing on its popularity. However, these findings help us better understand patterns in the music industry, the general TikTok audience, and the app's tendency to favor certain trending sounds. Our results contribute to quantifying the factors that influence virality and help fill a gap in understanding what makes a song successful on TikTok.

Limitations of Data: The data we used is specific to Southeast Asia and the Philippines, which is only a small portion of the global users of this app. Additionally, the dataset itself lacks some important factors that should be considered. For example, the dataset does not include the correct genre of each respective song, nor does it include how popular the artist behind the song is. Additionally, some variables within the dataset do not have accurate values. For example, the tempo column, which details the BPM of the song, is not accurate for many of the songs. This is also true with the modality and key of the songs. Another important consideration is that the dataset does not take into account the factor of artists paying for a recommendation push from the app or outside "influencers" to help promote their song. The dataset is limited to numerical and categorical variables based on data from TikTok, while a song's popularity on TikTok is not purely meritocratic. External factors can greatly affect a song's popularity on TikTok.

Contributions: Executive summary (Gunica), Background, problem, and goals of project (Gunica), Describe the dataset and data source (Alex), Exploratory data analysis (Siyuan), Methodology Model (Rue, Himmat), Analysis (Himmat, Siyuan), Conclusions (Gunica, Rue, Alex)