TikTok Platform:

Predicting Video Engagement & Virality

IMC 463 Final Project

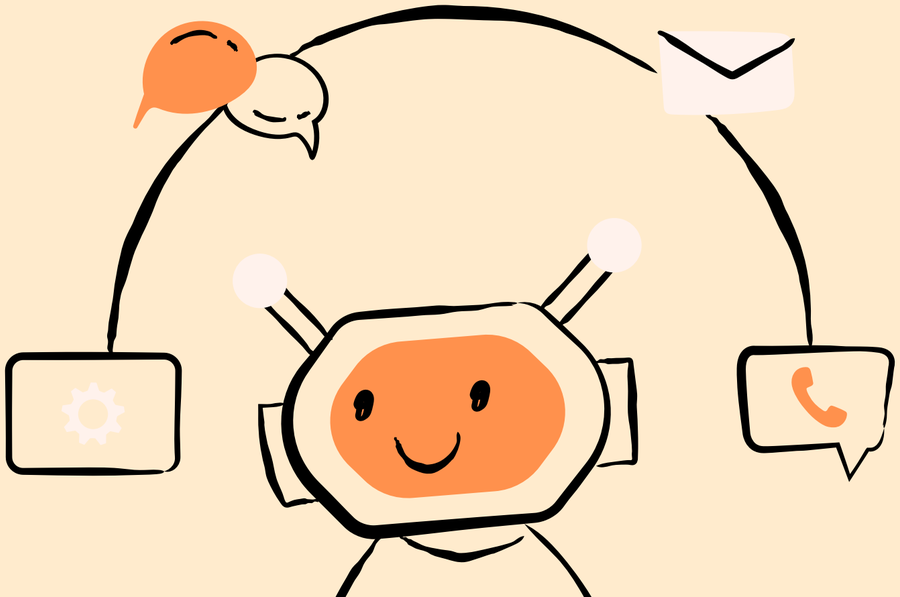
Group 9:

Grace Chen, Vanessa Chen, Amanda Lee, Sheryl Xu

Problem Definition

Research Question

How can TikTok video virality (measured by **view count** and **engagement**) be predicted using available user data and content feature variables, and to what extent can these variables be used to build a predictive model that forecasts a video's potential reach?



Dependent Variable Selection

After selecting the factors, we predicted **two** dependent variables, aim to detect whether these factors could affect the reach and engagement of TikTok videos.

Reach (View.Count)

Reach is mostly correlated with views, i.e., how many people saw the video. It's primarily driven by TikTok's algorithm.

However, it doesn't reflect whether a user is genuinely interested in this video or actively reacted to this video.

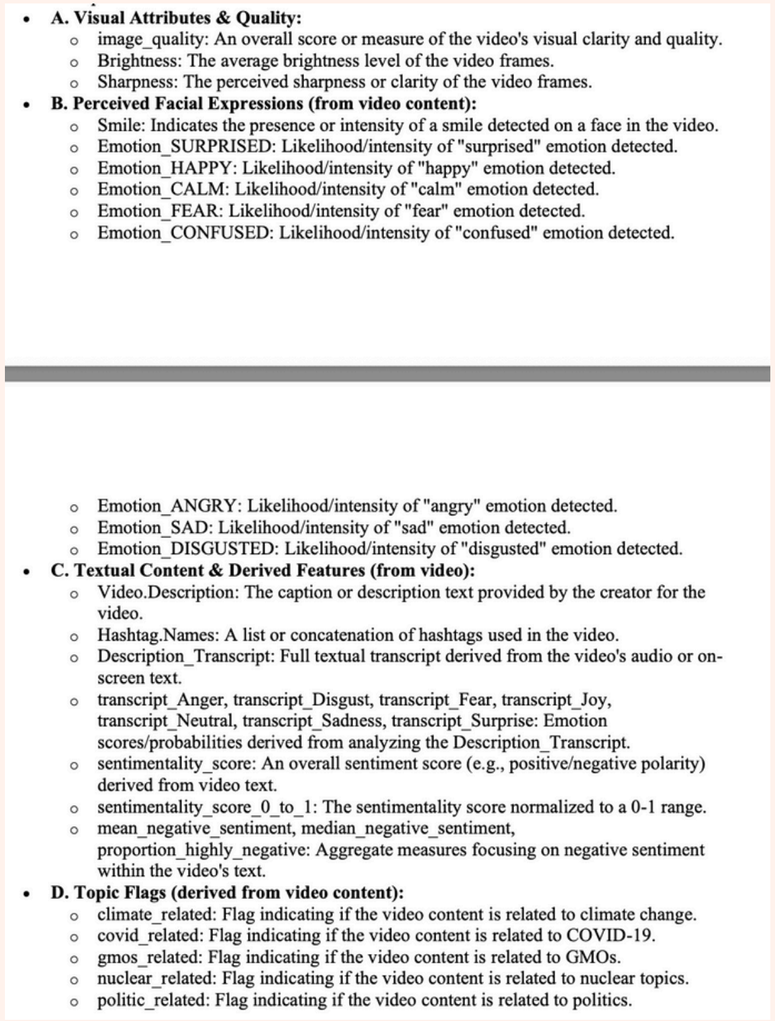
Engagement (Like.count + Share.count + Comment.count)

Engagement reflects active participation— how users responded to the content.

Sometimes high reach doesn't mean a high engagement, since users may scroll quickly but also count as a view. It often related to emotion, sentiment and other factors.

We believe reach and engagement are two different metrics that content creator should both take into consideration. Reach and Engagement have different business implications: Reach = brand awareness and Engagement = community/personal connection building.

Independent Variable Selection



- **A. Visual Attributes & Quality:**
  - image_quality: An overall score or measure of the video's visual clarity and quality.
  - Brightness: The average brightness level of the video frames.
  - Sharpness: The perceived sharpness or clarity of the video frames.
- **B. Perceived Facial Expressions (from video content):**
  - Smile: Indicates the presence or intensity of a smile detected on a face in the video.
  - Emotion_SURPRISED: Likelihood/intensity of "surprised" emotion detected.
  - Emotion_HAPPY: Likelihood/intensity of "happy" emotion detected.
  - Emotion_CALM: Likelihood/intensity of "calm" emotion detected.
  - Emotion_FEAR: Likelihood/intensity of "fear" emotion detected.
  - Emotion_CONFUSED: Likelihood/intensity of "confused" emotion detected.
  - Emotion_ANGRY: Likelihood/intensity of "angry" emotion detected.
  - Emotion_SAD: Likelihood/intensity of "sad" emotion detected.
  - Emotion_DISGUSTED: Likelihood/intensity of "disgusted" emotion detected.
- **C. Textual Content & Derived Features (from video):**
  - Video.Description: The caption or description text provided by the creator for the video.
  - Hashtag.Names: A list or concatenation of hashtags used in the video.
  - Description_Transcript: Full textual transcript derived from the video's audio or on-screen text.
  - transcript_Anger, transcript_Disgust, transcript_Fear, transcript_Joy, transcript_Neutral, transcript_Sadness, transcript_Surprise: Emotion scores/probabilities derived from analyzing the Description_Transcript.
  - sentimentality_score: An overall sentiment score (e.g., positive/negative polarity) derived from video text.
  - sentimentality_score_0_to_1: The sentimentality score normalized to a 0-1 range.
  - mean_negative_sentiment, median_negative_sentiment, proportion_highly_negative: Aggregate measures focusing on negative sentiment within the video's text.
- **D. Topic Flags (derived from video content):**
  - climate_related: Flag indicating if the video content is related to climate change.
  - covid_related: Flag indicating if the video content is related to COVID-19.
  - gmos_related: Flag indicating if the video content is related to GMOs.
  - nuclear_related: Flag indicating if the video content is related to nuclear topics.
  - politic_related: Flag indicating if the video content is related to politics.

Based on all the factors, we figured out three big categories that could affect the view and engagement of TikTok influencers' videos:

1. User profile characteristics:
   a. **Follower count, like sum count, and the following counts:** these are the metrics representing the social capital and popularity on platforms.
   b. **Verified Status:** serve as authenticity of content creator (categorical).
2. Video Quality
   a. **image_quality, brightness, sharpness:** Directly affect viewer experiences and retention
3. Emotional Content
   a. We believe all the emotion factors directly affect the engagement and views, and want to analyze whether different tone would affect the engagement differently.
4. Sentiment Analysis
   a. Similarly to the emotional content, we want to analyze whether sentiment detected in caption would affect the view and engagement.
5. Creator Demographics:
   a. **age, gender.deepface, gender.amazon, race:** These demographic factors may reveal patterns in content performance or audience engagement.

Methodology & Key findings

Regression Model - Reach

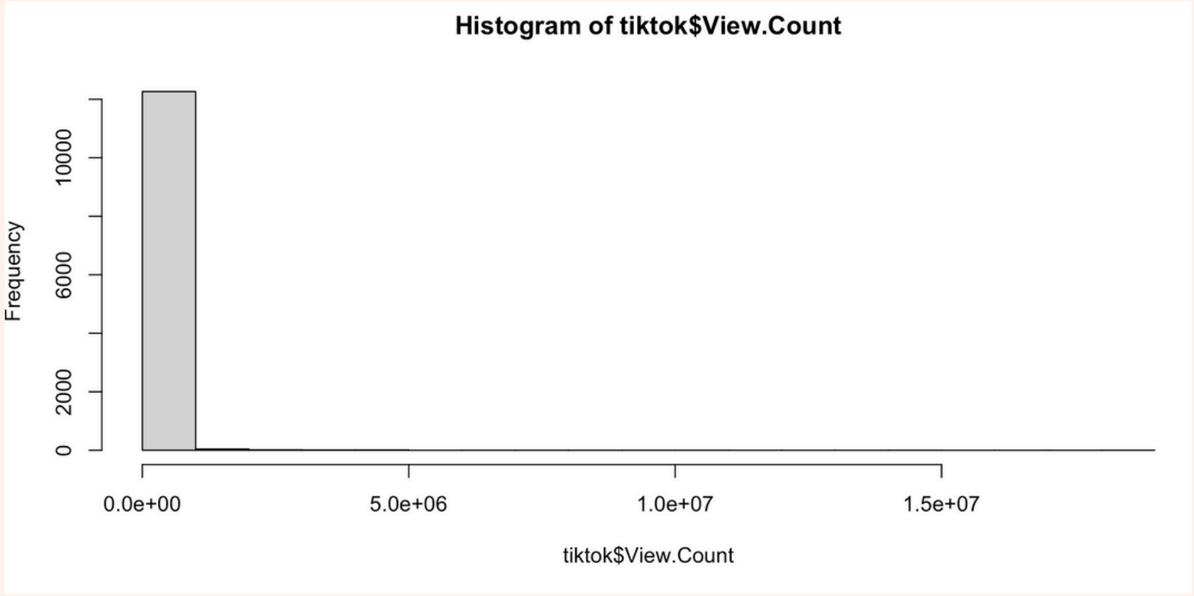
it.lm_view = lm(log(View.Count+1)~Follower.Count + Likes.sum.Count + Following.Count + as.factor(Verified.Status)
+ image_quality + Brightness + Sharpness + as.factor(Smile)+ Emotion_SURPRISED+Emotion_HAPPY
+Emotion_CALM+Emotion_FEAR+Emotion_CONFUSED+Emotion_ANGRY+Emotion_SAD+Emotion_DISGUSTED
+sentimentality_score_0_to_1 +age + as.factor(gender.deepface) + as.factor(gender.amazon) + as.factor(race), train_data)

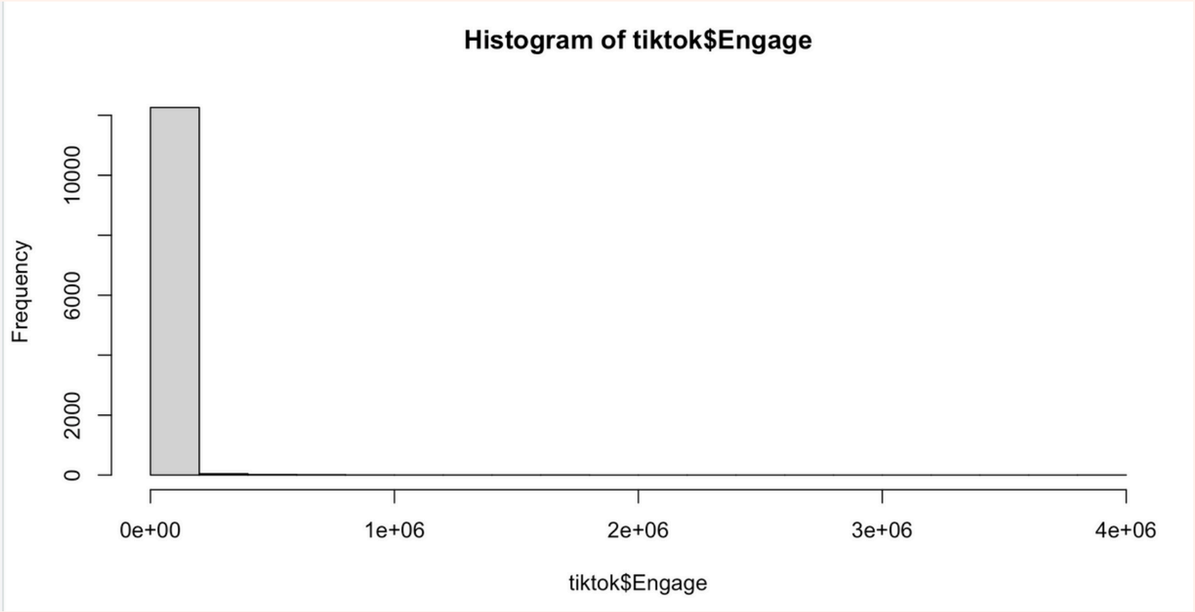
```
Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     6.292e+00  4.327e-01  14.542  < 2e-16 ***
Follower.Count                  2.444e-06  1.115e-07  21.925  < 2e-16 ***
Likes.sum.Count                -2.018e-08  1.173e-09 -17.200  < 2e-16 ***
Following.Count                -4.775e-05  2.570e-05  -1.858 0.063214 .
as.factor(Verified.Status)True  7.383e-01  1.662e-01   4.444 9.04e-06 ***
image_quality                   3.635e-03  1.649e-03   2.204 0.027582 *
Brightness                      4.005e-03  2.703e-03   1.482 0.138495
Sharpness                      -1.493e-03  1.287e-03  -1.159 0.246393
as.factor(Smile)True            8.000e-03  1.780e-01   0.045 0.964152
Emotion_SURPRISED              -3.270e-03  3.035e-03  -1.078 0.281265
Emotion_HAPPY                  -7.703e-03  4.058e-03  -1.898 0.057722 .
Emotion_CALM                   -1.190e-02  3.247e-03  -3.665 0.000250 ***
Emotion_FEAR                   -7.147e-03  4.927e-03  -1.451 0.146970
Emotion_CONFUSED               -3.260e-03  3.253e-03  -1.002 0.316361
Emotion_ANGRY                  -1.125e-02  4.447e-03  -2.529 0.011481 *
Emotion_SAD                    -1.407e-02  3.501e-03  -4.019 5.92e-05 ***
Emotion_DISGUSTED              -8.391e-03  5.885e-03  -1.426 0.153937
sentimentality_score_0_to_1    -2.071e-01  1.214e-01  -1.706 0.088102 .
age                            -1.873e-02  4.636e-03  -4.040 5.42e-05 ***
as.factor(gender.deepface)Woman -3.123e-01  1.312e-01  -2.380 0.017337 *
as.factor(gender.amazon)Male    4.792e-01  1.240e-01   3.864 0.000113 ***
as.factor(race)black           -4.181e-01  1.983e-01  -2.109 0.035012 *
as.factor(race)indian           4.603e-01  3.488e-01   1.320 0.187052
as.factor(race)latino hispanic  2.525e-01  2.503e-01   1.009 0.313036
as.factor(race)middle eastern   2.036e-01  1.682e-01   1.210 0.226203
as.factor(race)white            4.294e-01  1.460e-01   2.940 0.003294 **
```

From the model, we can see there are several factors affect the view count of the video.

- **Follower and like sum count:** directly indicate the popularity of a content creator
  - sum count acts negatively, probably because of multicollinearity
- The **negative emotion** strongly reduce the performance of the reach of the video.
- Demographic wise, **female** performed worse, and **black** creators shows a reduced performance.

Regression Model - log transformation





Since both view count and engagement are very likely to be **right skewed** (i.e., only a few of the videos have an extremely high reach; most of them only have a very low reach), we decided to use a log model to transform these two models.

Regression Model - Engage

tiktok$Engage <- tiktok$Share.Count + tiktok$Like.Count + tiktok$Comment.Count

fit.lm_engage = lm(log(Engage +1)~Follower.Count + Likes.sum.Count + Following.Count + as.factor(Verified.Status)

+ image_quality + Brightness + Sharpness + as.factor(Smile)+ Emotion_SURPRISED+Emotion_HAPPY

+Emotion_CALM+Emotion_FEAR+Emotion_CONFUSED+Emotion_ANGRY+Emotion_SAD+Emotion_DISGUSTED

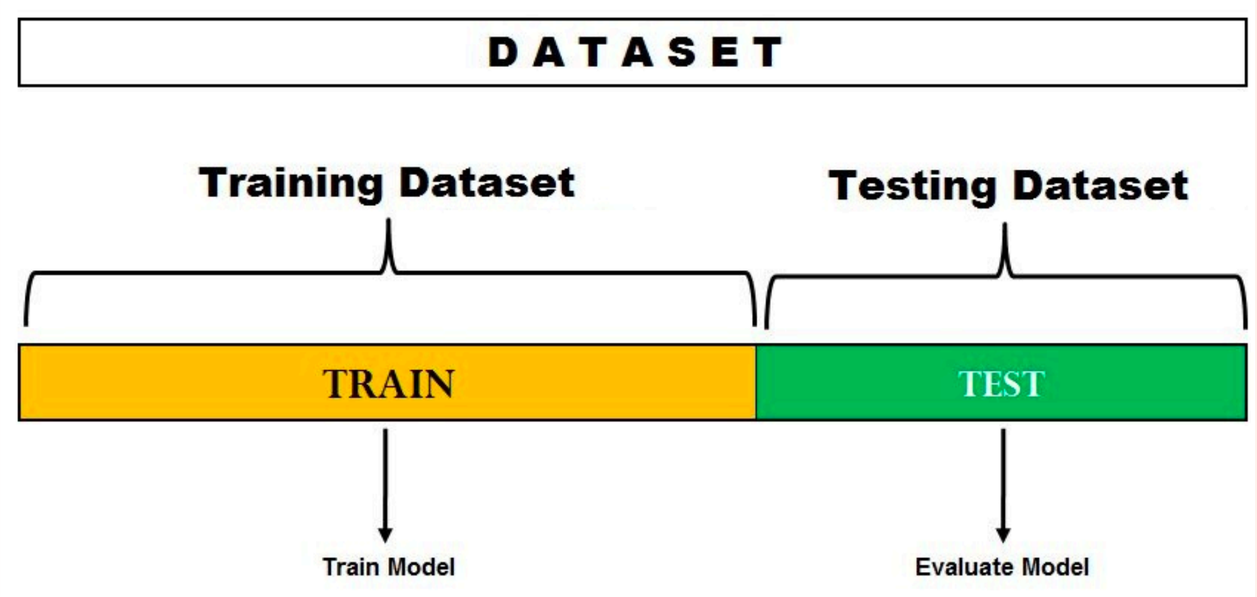
+sentimentality_score_0_to_1 +age + as.factor(gender.deepface) + as.factor(gender.amazon) + as.factor(race), train_data)

```
Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     5.573e+00  3.317e-01  16.804  < 2e-16 ***
Follower.Count                  1.926e-06  8.544e-08  22.539  < 2e-16 ***
Likes.sum.Count                -1.579e-08  8.995e-10 -17.550  < 2e-16 ***
Following.Count                 9.187e-07  1.970e-05   0.047 0.962805
as.factor(Verified.Status)True  5.382e-01  1.274e-01   4.226 2.42e-05 ***
image_quality                   5.937e-03  1.264e-03   4.696 2.72e-06 ***
Brightness                      4.138e-04  2.072e-03   0.200 0.841701
Sharpness                       1.476e-03  9.868e-04   1.495 0.134859
as.factor(Smile)True           -6.995e-02  1.364e-01  -0.513 0.608180
Emotion_SURPRISED               9.415e-04  2.326e-03   0.405 0.685659
Emotion_HAPPY                  -3.725e-03  3.110e-03  -1.198 0.231120
Emotion_CALM                   -4.965e-03  2.489e-03  -1.995 0.046116 *
Emotion_FEAR                   -3.898e-03  3.777e-03  -1.032 0.302009
Emotion_CONFUSED                1.260e-03  2.494e-03   0.505 0.613447
Emotion_ANGRY                  -2.882e-03  3.409e-03  -0.845 0.397929
Emotion_SAD                    -8.555e-03  2.684e-03  -3.188 0.001442 **
Emotion_DISGUSTED              -2.762e-03  4.511e-03  -0.612 0.540322
sentimentality_score_0_to_1     1.314e-01  9.304e-02   1.412 0.158086
age                            -1.893e-02  3.553e-03  -5.328 1.04e-07 ***
as.factor(gender.deepface)Woman -2.327e-01  1.006e-01  -2.314 0.020690 *
as.factor(gender.amazon)Male    2.147e-02  9.507e-02   0.226 0.821321
as.factor(race)black            8.170e-02  1.520e-01   0.538 0.590863
as.factor(race)indian           4.336e-01  2.674e-01   1.622 0.104900
as.factor(race)latino hispanic  2.256e-01  1.918e-01   1.176 0.239703
as.factor(race)middle eastern   1.899e-01  1.290e-01   1.473 0.140929
as.factor(race)white            3.938e-01  1.119e-01   3.518 0.000438 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The result of the engagement model is slightly different.

- Follower count and like sum count still have a significant impact on engagement. The age factor is also a significant one: younger creators have more engagement. The verified status is also positively related to the engagement.
- Demographic-wise, white creators receive more engagement of their videos.
- Compared to the view model, high engagement is tightly correlated to the image_quality.

Methodology



Out-of-Sample Estimation → 70/30 train/test split (holdout validation)

- Purpose: Evaluate model's generalization to unseen data
- Avoids overfitting and gives realistic performance estimates.

Methodology

- We have many predictors.
- We want a balance between *complexity* and *performance*.
- We are doing *exploratory modeling* and want guidance on important features.

Stepwise Selection

- We have many predictors.
- We want to shrink and select variables via *penalty*.
- We want better *generalization* and scalability.
- We want global optimization with better *interpretability*.

Lasso Regression

- Some predictors are correlated, and we want to *remove collinearity* via independent PCs.
- We want to have fewer predictors.
- We want to *reduce noise* by filtering out minor components with low variance.

PCA

Methodology

Methods	Stepwise Selection	Lasso Regression	PCA-Based Regression
Purpose	Selects the best subset of variables	Shrinks and selects variables via penalty	Reduces dimensionality using principal components
Variable Interpretability	✅ Easy to interpret (retains original variables)	✅ Mostly interpretable (keeps real variables)	❌ Not interpretable (PCs are combinations of variables)
Handles Multicollinearity	⚠️ Sometimes (removes redundant vars)	✅ Strong at handling multicollinearity	✅ Removes collinearity via orthogonal PCs
Model Limitations	❌ Only focus on local optimization	❌ Multicollinearity Bias (Retains one variable from correlated predictors)	❌ Uses abstract PCs (less human-readable)

Key Findings: Stepwise Selections

Goal: Predict log(View.Count + 1) & log(Engage + 1) using user metadata and content features.

Methods:

- Stepwise variable selection
- 10-fold cross-validation on 70% training data
- Final evaluation on 30% held-out test set

Best Variables Selected via CV Stepwise

```
> print(best_vars)
                  (Intercept)                Follower.Count              Likes.sum.Count as.factor(Verified.Status)True
                 6.117145e+00                 2.469552e-06                -2.046420e-08                  7.096583e-01
                  Emotion_CALM                  Emotion_SAD                          age   as.factor(gender.amazon)Male
                 -6.094611e-03                -9.459223e-03                -1.769787e-02                  7.239711e-01
       as.factor(race)black
                 -7.976684e-01
```

Positive predictors:

- Follower.Count, Verified.Status, gender.amazon = Male

Negative predictors:

- Likes.sum.Count, Emotion_CALM, Emotion_SAD, age, race = Black

Key Findings: Stepwise Selections for View

Basic Formula

```
Call:
lm(formula = log(View.Count + 1) ~ Follower.Count + Likes.sum.Count +
    Following.Count + as.factor(Verified.Status) + image_quality +
    Brightness + Sharpness + as.factor(Smile) + Emotion_SURPRISED +
    Emotion_HAPPY + Emotion_CALM + Emotion_FEAR + Emotion_CONFUSED +
    Emotion_ANGRY + Emotion_SAD + Emotion_DISGUSTED + sentimentality_score_0_to_1 +
    age + as.factor(gender.deepface) + as.factor(gender.amazon) +
    as.factor(race), data = train_data)

Residuals:
     Min       1Q   Median       3Q      Max
-15.5246  -1.7008  -0.0553   1.6341  10.8052

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     6.292e+00  4.327e-01  14.542  < 2e-16 ***
Follower.Count                  2.444e-06  1.115e-07  21.925  < 2e-16 ***
Likes.sum.Count                -2.018e-08  1.173e-09 -17.200  < 2e-16 ***
Following.Count                -4.775e-05  2.570e-05  -1.858 0.063214 .
as.factor(Verified.Status)True  7.383e-01  1.662e-01   4.444 9.04e-06 ***
image_quality                   3.635e-03  1.649e-03   2.204 0.027582 *
Brightness                      4.005e-03  2.703e-03   1.482 0.138495
Sharpness                      -1.493e-03  1.287e-03  -1.159 0.246393
as.factor(Smile)True            8.000e-03  1.780e-01   0.045 0.964152
Emotion_SURPRISED              -3.270e-03  3.035e-03  -1.078 0.281265
Emotion_HAPPY                  -7.703e-03  4.058e-03  -1.898 0.057722 .
Emotion_CALM                   -1.190e-02  3.247e-03  -3.665 0.000250 ***
Emotion_FEAR                   -7.147e-03  4.927e-03  -1.451 0.146970
Emotion_CONFUSED               -3.260e-03  3.253e-03  -1.002 0.316361
Emotion_ANGRY                  -1.125e-02  4.447e-03  -2.529 0.011481 *
Emotion_SAD                    -1.407e-02  3.501e-03  -4.019 5.92e-05 ***
Emotion_DISGUSTED              -8.391e-03  5.885e-03  -1.426 0.153937
sentimentality_score_0_to_1    -2.071e-01  1.214e-01  -1.706 0.088102 .
age                            -1.873e-02  4.636e-03  -4.040 5.42e-05 ***
as.factor(gender.deepface)Woman -3.123e-01  1.312e-01  -2.380 0.017337 *
as.factor(gender.amazon)Male    4.792e-01  1.240e-01   3.864 0.000113 ***
as.factor(race)black           -4.181e-01  1.983e-01  -2.109 0.035012 *
as.factor(race)indian           4.603e-01  3.488e-01   1.320 0.187052
as.factor(race)latino hispanic  2.525e-01  2.503e-01   1.009 0.313036
as.factor(race)middle eastern   2.036e-01  1.682e-01   1.210 0.226203
as.factor(race)white            4.294e-01  1.460e-01   2.940 0.003294 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.526 on 4920 degrees of freedom
Multiple R-squared:  0.185,     Adjusted R-squared:  0.1808
F-statistic: 44.66 on 25 and 4920 DF,  p-value: < 2.2e-16
```

After Stepwise Selection

log(View.Count + 1) ~ Follower.Count + Likes.sum.Count + as.factor(Verified.Status) +Emotion_CALM+Emotion_SAD+age + as.factor(gender.amazon) + as.factor(race))

```
Call:
lm(formula = stepwise_formula, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-15.650  -1.712  -0.051   1.649  10.730

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                    5.814e+00  2.036e-01  28.555  < 2e-16 ***
Follower.Count                 2.447e-06  1.113e-07  21.992  < 2e-16 ***
Likes.sum.Count               -2.027e-08  1.172e-09 -17.296  < 2e-16 ***
as.factor(Verified.Status)True 7.394e-01  1.662e-01   4.450 8.79e-06 ***
Emotion_CALM                  -6.150e-03  8.968e-04  -6.858 7.87e-12 ***
Emotion_SAD                   -9.245e-03  2.320e-03  -3.985 6.84e-05 ***
age                           -1.884e-02  4.570e-03  -4.124 3.78e-05 ***
as.factor(gender.amazon)Male   7.345e-01  7.940e-02   9.252  < 2e-16 ***
as.factor(race)black          -4.596e-01  1.965e-01  -2.339  0.01938 *
as.factor(race)indian          4.124e-01  3.486e-01   1.183  0.23683
as.factor(race)latino hispanic 2.069e-01  2.501e-01   0.827  0.40804
as.factor(race)middle eastern  1.808e-01  1.675e-01   1.080  0.28035
as.factor(race)white           4.136e-01  1.451e-01   2.851  0.00437 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.532 on 4933 degrees of freedom
Multiple R-squared:  0.1795,    Adjusted R-squared:  0.1775
F-statistic: 89.92 on 12 and 4933 DF,  p-value: < 2.2e-16
```

Summary of Omissions

- *Visual features removed:* image_quality, Brightness, Sharpness
- *Facial expression:* Smile
- *Many emotions dropped:* SURPRISED, HAPPY, FEAR, CONFUSED, ANGRY, DISGUSTED
- *Other dropped features*: sentimentality_score_0_to_1, Following.Count, gender.deepface

Key Findings: Stepwise Selections for View

Stepwise Selection Formula of View.Count

```
Call:
lm(formula = stepwise_formula, data = train_data)

Residuals:
    Min      1Q  Median      3Q     Max
-15.650  -1.712  -0.051   1.649  10.730

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     5.814e+00  2.036e-01  28.555  < 2e-16 ***
Follower.Count                  2.447e-06  1.113e-07  21.992  < 2e-16 ***
Likes.sum.Count                -2.027e-08  1.172e-09 -17.296  < 2e-16 ***
as.factor(Verified.Status)True  7.394e-01  1.662e-01   4.450 8.79e-06 ***
Emotion_CALM                   -6.150e-03  8.968e-04  -6.858 7.87e-12 ***
Emotion_SAD                    -9.245e-03  2.320e-03  -3.985 6.84e-05 ***
age                            -1.884e-02  4.570e-03  -4.124 3.78e-05 ***
as.factor(gender.amazon)Male    7.345e-01  7.940e-02   9.252  < 2e-16 ***
as.factor(race)black           -4.596e-01  1.965e-01  -2.339  0.01938 *
as.factor(race)indian           4.124e-01  3.486e-01   1.183  0.23683
as.factor(race)latino hispanic  2.069e-01  2.501e-01   0.827  0.40804
as.factor(race)middle eastern   1.808e-01  1.675e-01   1.080  0.28035
as.factor(race)white            4.136e-01  1.451e-01   2.851  0.00437 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.532 on 4933 degrees of freedom
Multiple R-squared:  0.1795,    Adjusted R-squared:  0.1775
F-statistic: 89.92 on 12 and 4933 DF,  p-value: < 2.2e-16
```

Variable	Estimate	Interpretation
Follower.Count	0.000002447	A 1-unit increase in followers is associated with a 0.00000245 increase in log(View.Count + 1). Though tiny in raw scale, it's statistically significant (p < 2e-16), and meaningful over large follower changes.
Likes.sum.Count	-0.00000002027	Surprisingly negative, suggests more likes are associated with slightly fewer views, but the scale is extremely small
Verified.Status (True)	0.7394	Verified users are expected to have ~107% more views (exp(0.7394) ≈ 2.09) than non-verified users, holding other variables constant.
Emotion_CALM	-0.00615	Presence of a calm emotion is associated with a slight decrease in log(Views) — potentially less attention-grabbing content.
Emotion_SAD	-0.009245	Stronger negative impact than CALM — sad content is less viral on average.
age	-0.01884	Each additional year of age corresponds to a ~1.88% drop in expected view count, controlling for other features.
gender.amazon = Male	0.7345	Male-labeled faces get ~108% more views than non-male, all else equal (exp(0.7345) ≈ 2.08).
race = Black	-0.4596	Predicted to get ~36.8% fewer views compared to the base race category (exp(-0.4596) ≈ 0.632).
race = White	0.4136	Predicted to get ~51.2% more views compared to the baseline (exp(0.4136) ≈ 1.512).

Result Interpretation

- About **17.75%** of the variation in the log-transformed view count is explained by the model.
- **Follower count, verification, male label, and white race label** are strong positive predictors of higher view counts.
- **Age, CALM/SAD emotions, and Black race label** are associated with lower view counts.

Key Findings: Stepwise Selections for Engagement

Stepwise Selection Formula of Engage

```
Call:
lm(formula = stepwise_formula_engage, data = train_data)

Residuals:
     Min       1Q   Median       3Q      Max
-12.7621  -1.2624  -0.3182   1.0463   9.0238

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                       5.726e+00  1.907e-01  30.024  < 2e-16 ***
Follower.Count                    1.924e-06  8.518e-08  22.591  < 2e-16 ***
Likes.sum.Count                  -1.576e-08  8.965e-10 -17.582  < 2e-16 ***
as.factor(Verified.Status)True    5.356e-01  1.271e-01   4.213 2.56e-05 ***
image_quality                     5.951e-03  1.256e-03   4.739 2.21e-06 ***
Sharpness                         1.513e-03  9.606e-04   1.575 0.115326
Emotion_HAPPY                    -5.497e-03  1.147e-03  -4.793 1.69e-06 ***
Emotion_CALM                     -6.068e-03  9.212e-04  -6.587 4.97e-11 ***
Emotion_FEAR                     -4.891e-03  3.244e-03  -1.508 0.131641
Emotion_ANGRY                    -3.929e-03  2.622e-03  -1.499 0.134054
Emotion_SAD                      -9.503e-03  1.877e-03  -5.064 4.26e-07 ***
Emotion_DISGUSTED                -3.826e-03  3.940e-03  -0.971 0.331604
sentimentality_score_0_to_1       1.303e-01  9.285e-02   1.403 0.160646
age                              -1.892e-02  3.535e-03  -5.352 9.08e-08 ***
as.factor(gender.deepface)Woman  -2.524e-01  6.555e-02  -3.851 0.000119 ***
as.factor(race)black              7.580e-02  1.505e-01   0.504 0.614505
as.factor(race)indian             4.334e-01  2.663e-01   1.628 0.103694
as.factor(race)latino hispanic    2.271e-01  1.914e-01   1.187 0.235384
as.factor(race)middle eastern     1.934e-01  1.281e-01   1.510 0.131091
as.factor(race)white              3.968e-01  1.112e-01   3.569 0.000362 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.935 on 4926 degrees of freedom
Multiple R-squared:  0.181,     Adjusted R-squared:  0.1779
F-statistic: 57.31 on 19 and 4926 DF,  p-value: < 2.2e-16
```

Variable	Estimate	Interpretation
Follower.Count	0.0000	More followers → more engagement.
Likes.sum.Count	−1.576e-08	Surprisingly negative; likely due to multicollinearity (likes already included in Engage).
Verified.Status = True	0.5356	Verified users get more engagement.
image_quality	0.0060	Better image quality increases engagement.
Emotion_HAPPY	−0.00550	Happy expressions reduce engagement.
Emotion_CALM	−0.00607	Calm content underperforms.
Emotion_SAD	−0.00950	Sadness significantly lowers engagement.
age	−0.01892	Older users get less engagement.
gender.deepface = Woman	−0.2524	Female-labeled faces receive less engagement.
race = White	0.3968	White race label associated with higher engagement.

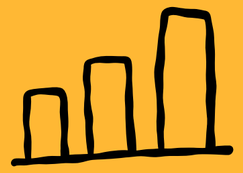
Result Interpretation

- The model explains about **17.8%** of the variance in engagement.
- **Follower.Count, Verified.Status, image_quality, race = White** are strong positive predictors of higher engagement
- **Likes.sum.Count, Emotion_HAPPY, Emotion_CALM, Emotion_SAD, age, gender.deepface = Woman, race = Black** are all negatively associated with engagement — possibly reflecting preference for exciting or controversial content.

Key Findings: Lasso Regression for View

Lasso Regression of View.Count

```
> print(coefficients_lasso_view)
27 x 1 sparse Matrix of class "dgCMatrix"
                                          s1
(Intercept)                     5.504371e+00
(Intercept)                     .
Follower.Count                  1.552492e-06
Likes.sum.Count                -1.084401e-08
Following.Count                -3.998470e-05
as.factor(Verified.Status)True  1.073524e+00
image_quality                   2.651928e-03
Brightness                      4.192604e-03
Sharpness                      -6.456749e-04
as.factor(Smile)True            .
Emotion_SURPRISED               3.240076e-03
Emotion_HAPPY                   .
Emotion_CALM                   -3.226051e-03
Emotion_FEAR                    .
Emotion_CONFUSED                5.495774e-04
Emotion_ANGRY                   .
Emotion_SAD                    -4.126397e-03
Emotion_DISGUSTED               .
sentimentality_score_0_to_1    -8.513281e-02
age                            -1.320874e-02
as.factor(gender.deepface)Woman -1.904295e-01
as.factor(gender.amazon)Male    4.839711e-01
as.factor(race)black           -3.841310e-01
as.factor(race)indian           8.762434e-02
as.factor(race)latino hispanic  .
as.factor(race)middle eastern   .
as.factor(race)white            2.020228e-01
```



Methods:

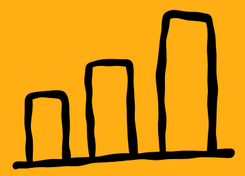
- Lasso regression via glmnet with $\alpha = 1$
- 10-fold cross-validation to find optimal λ
- Use λ.1se (simpler model within 1 SE of best MSE)

- **Positive drivers:**
  - **verified status, follower count, male/white features, high image quality**
- **Negative drivers:**
  - **sad/angry/calm emotions, being Black or woman-labeled, higher age**
- **Variables Shrunk to 0**
  - **Emotion_HAPPY, Emotion_FEAR, Emotion_DISGUSTED, Smile, latino hispanic, middle eastern**
  - These variables did not add additional explanatory power beyond what's already captured by other predictors

Key Findings: Lasso Regression for Engagement

Lasso Regression of Engage

```
27 x 1 sparse Matrix of class "dgCMatrix"
                                          s1
(Intercept)                     5.608047e+00
(Intercept)                     .
Follower.Count                  1.409043e-22
Likes.sum.Count                 .
Following.Count                 .
as.factor(Verified.Status)True  .
image_quality                   .
Brightness                      .
Sharpness                       .
as.factor(Smile)True            .
Emotion_SURPRISED               .
Emotion_HAPPY                   .
Emotion_CALM                    .
Emotion_FEAR                    .
Emotion_CONFUSED                .
Emotion_ANGRY                   .
Emotion_SAD                     .
Emotion_DISGUSTED               .
sentimentality_score_0_to_1     .
age                             .
as.factor(gender.deepface)Woman .
as.factor(gender.amazon)Male    .
as.factor(race)black            .
as.factor(race)indian           .
as.factor(race)latino hispanic  .
as.factor(race)middle eastern   .
as.factor(race)white            .
```



Methods:

- Lasso regression via glmnet with $\alpha = 1$
- 10-fold cross-validation to find optimal λ
- Use λ.1se (simpler model within 1 SE of best MSE)

Findings

- Only the **Follower.Count** (1.4e-22) is retained
- All other predictors (demographics, emotions, image quality) were shrunk to zero.

Interpretation:

- Lasso determined that *none* of the predictors added meaningful, non-redundant information for predicting engagement.
- **Likely causes:**
  - Multicollinearity: Engage includes Likes, which are also used as predictors.
  - Noisy or weak signals from demographic or emotional features when predicting Engage.

Key Findings: PCA

Goal: Reduce predictors, capture as much variance, and remove multicollinearity from model

Methods:

- PCA on 70% training data
- Final evaluation on 30% held-out test set

Variables Included

Follower.Count
Likes.sum.Count
Following.Count
Verified.Status
image_quality
Brightness
Sharpness

Smile
sentimentality_score_0_to_1
age
gender.deepface
gender.amazon
race

Emotion_SURPRISED
Emotion_HAPPY
Emotion_CALM
Emotion_FEAR
Emotion_CONFUSED
Emotion_ANGRY
Emotion_SAD
Emotion_DISGUSTED

These variables are selected for performing PCA because they have a higher likelihood of influencing video virality.

Key Findings: PCA

PCA Reults:

The first PC only explained 10.6% of the variance.

	PC1	PC2	PC3	PC4	PC5
Proportion of Variance	0.1063	0.08917	0.07499	0.06265	0.05707
Cumulative Proportion	0.1063	0.1955	0.27049	0.33314	0.39021

	PC6	PC7	PC8	PC9	PC10
Proportion of Variance	0.05609	0.05505	0.04426	0.04166	0.04014
Cumulative Proportion	0.4463	0.50134	0.5456	0.58725	0.6274

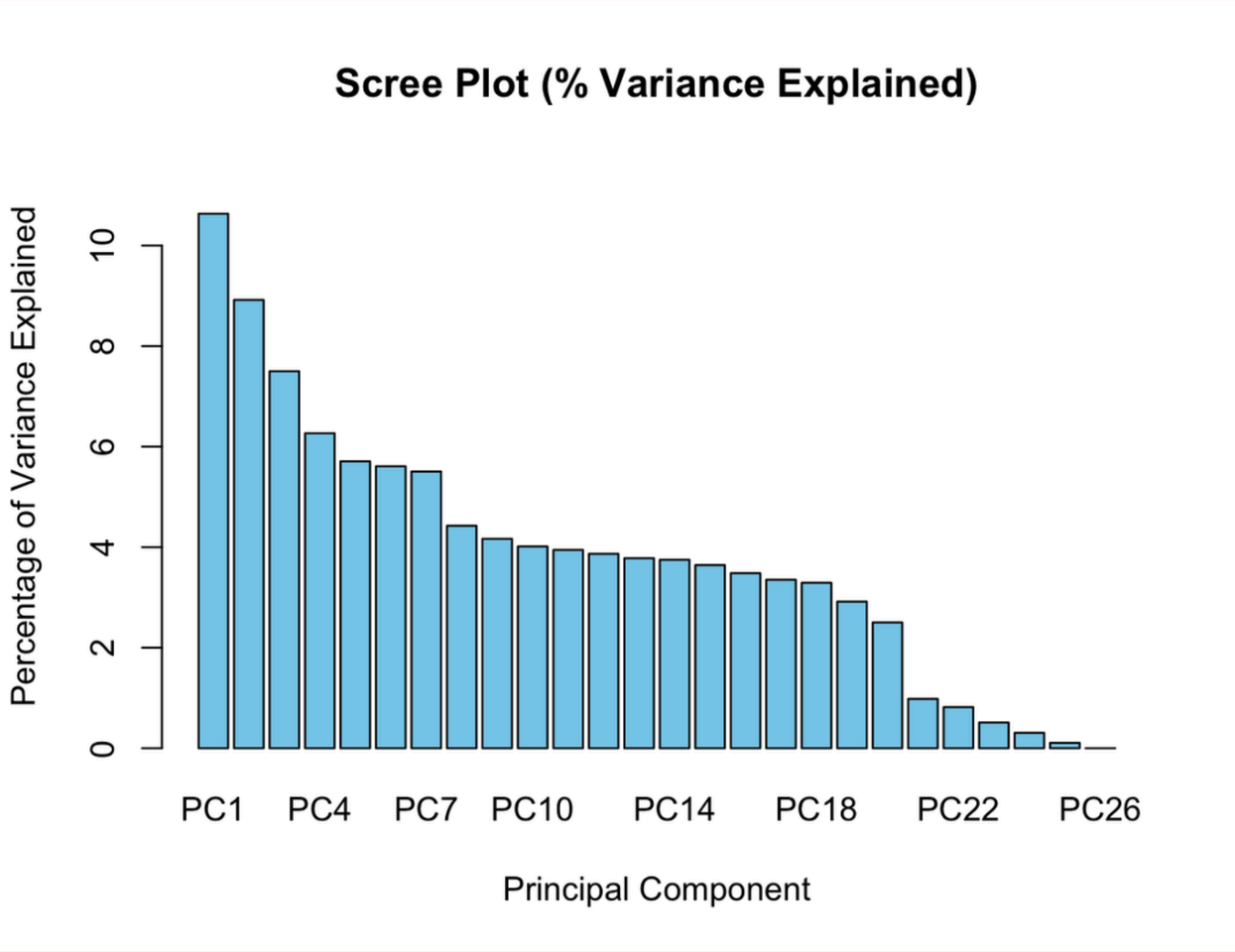
	PC11	PC12	PC13	PC14	PC15
Proportion of Variance	0.03946	0.03868	0.03781	0.03749	0.03643
Cumulative Proportion	0.66686	0.70553	0.74334	0.78083	0.81726

The PCA results show that **80% of variance is explained by the first 15 PCs**.

Therefore, we keep the first 15 PCs and will later run regression to find their predicting power on virality.

Key Findings: PCA

Scree Plot visualization of percent variance explained by PCs:



Scree plot shows that there is **no single dominant pattern**.

- This means the included **variables are weakly correlated** or highly independent.
- This result reflects the true situation because virality is expected to be influenced by many factors except of one underlying drive.

Key Findings: PCA

Linear regression result on **View.Count** using the 15 PCs stored from PCA results

```
Call:
lm(formula = Y ~ ., data = df_pca_train)

Residuals:
    Min      1Q  Median      3Q     Max
-20.511  -1.823  -0.056   1.724  10.203

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.979714   0.037513 159.404  < 2e-16 ***
PC1         -0.360493   0.022565 -15.976  < 2e-16 ***
PC2          0.365522   0.024639  14.835  < 2e-16 ***
PC3         -0.051564   0.026868  -1.919  0.05502 .
PC4         -0.264278   0.029395  -8.991  < 2e-16 ***
PC5          0.041751   0.030800   1.356  0.17530
PC6         -0.083666   0.031068  -2.693  0.00711 **
PC7         -0.040114   0.031360  -1.279  0.20090
PC8         -0.074470   0.034975  -2.129  0.03328 *
PC9          0.141479   0.036049   3.925 8.81e-05 ***
PC10        -0.119885   0.036724  -3.265  0.00110 **
PC11         0.008404   0.037039   0.227  0.82051
PC12         0.017869   0.037413   0.478  0.63295
PC13        -0.014843   0.037839  -0.392  0.69488
PC14         0.066388   0.038002   1.747  0.08070 .
PC15        -0.036940   0.038546  -0.958  0.33795
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.638 on 4930 degrees of freedom
Multiple R-squared:  0.1094,    Adjusted R-squared:  0.1067
F-statistic: 40.37 on 15 and 4930 DF,  p-value: < 2.2e-16
```

Regression result shows that:

- PC1, PC4, PC6, PC8, PC10 have significantly negative impact on View.Count
- PC2 & PC9 have significantly positive impact on View.Count

Out of these PCs with significant impact, **PC1 has the largest negative impact, and PC2 has the largest positive impact**.

Key Findings: PCA

Linear regression result on **Engage** using the 15 PCs stored from PCA results

```
Call:
lm(formula = Y_engage ~ ., data = df_pca_train_engage)

Residuals:
     Min       1Q   Median       3Q      Max
-16.4756  -1.3549  -0.3777   1.1305   8.8418

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.60805    0.02879 194.789  < 2e-16 ***
PC1         -0.29606    0.01732 -17.096  < 2e-16 ***
PC2          0.23933    0.01891  12.656  < 2e-16 ***
PC3         -0.11320    0.02062  -5.490 4.23e-08 ***
PC4         -0.12510    0.02256  -5.545 3.09e-08 ***
PC5          0.10612    0.02364   4.489 7.31e-06 ***
PC6         -0.03966    0.02384  -1.663 0.096289 .
PC7         -0.00373    0.02407  -0.155 0.876836
PC8         -0.10162    0.02684  -3.786 0.000155 ***
PC9         -0.00875    0.02767  -0.316 0.751819
PC10        -0.05246    0.02818  -1.861 0.062748 .
PC11         0.01521    0.02843   0.535 0.592553
PC12        -0.04318    0.02871  -1.504 0.132668
PC13        -0.04728    0.02904  -1.628 0.103600
PC14         0.04768    0.02916   1.635 0.102172
PC15        -0.05571    0.02958  -1.883 0.059740 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.025 on 4930 degrees of freedom
Multiple R-squared:  0.1029,    Adjusted R-squared:  0.1002
F-statistic: 37.71 on 15 and 4930 DF,  p-value: < 2.2e-16
```

Regression result shows that:

- PC1, PC3, PC4, PC8 have significantly negative impact on Engage
- PC2 & PC5 have significantly positive impact on Engage

Out of these PCs with significant impact, **PC1 has the largest negative impact, and PC2 has the largest positive impact**.

Key Findings: PCA

By examining the **top 5 positive loadings in PC1 and PC2,** we generalized possible **underlying pattern in characteristics for viral videos.**

PC1: 10.6% of variance
(Negative drivers)

Variable	Loading
Verified.StatusFalse	0.388803371
Emotion_CALM	0.293366732
gender.amazonMale	0.23705927
age	0.112600577
raceblack	0.092756723

Small account inspirational videos made by Black creators

PC2: 8.9% of data variance
(Positive drivers)

Variable	Loading
Follower.Count	0.37352991
gender.amazonMale	0.36110377
Likes.sum.Count	0.33802407
Verified.StatusTrue	0.31564549
age	0.112600577

Large account videos made by established male creators

Model Comparisons for Predicting View.Count

Comparison of Models' Performance on 30% Test Set

Model	MSE	R^2	Interpretation
Stepwise (with CV)	7.137873	0.0922	**Lowest performance; weak predictive power**
PCA Regression	7.009429	0.1085	**Slightly better than stepwise, moderate complexity**
Lasso ✅	6.856228	0.1280	**Best performance; balances accuracy and sparsity**

✅ **Use the Lasso Regression Model to predict View.Count**

- *Lowest Test MSE*: Lasso has the smallest prediction error.
- *Highest R^2*: It explains the most variance on unseen data (~12.8%).
- *Regularizatio*vant predictors.
- *Interpretability*: Keeps model sparse and focused on the strongest signals.

Model Comparisons for Predicting Engagement

Comparison of Models' Performance on 30% Test Set

Model	Test MSE	Test R^2	Interpretation
✅ **Stepwise (CV)**	4.0337	0.0915	**Good R^2 and lowest MSE among all models**
PCA Regression	4.2763	0.1073	**Best R^2, but slightly higher MSE**
Lasso Regression	4.4841	0.0639	**Worst performance: both lowest R^2 and highest MSE,** likely because it shrinks nearly all coefficients to zero, oversimplifying the model.

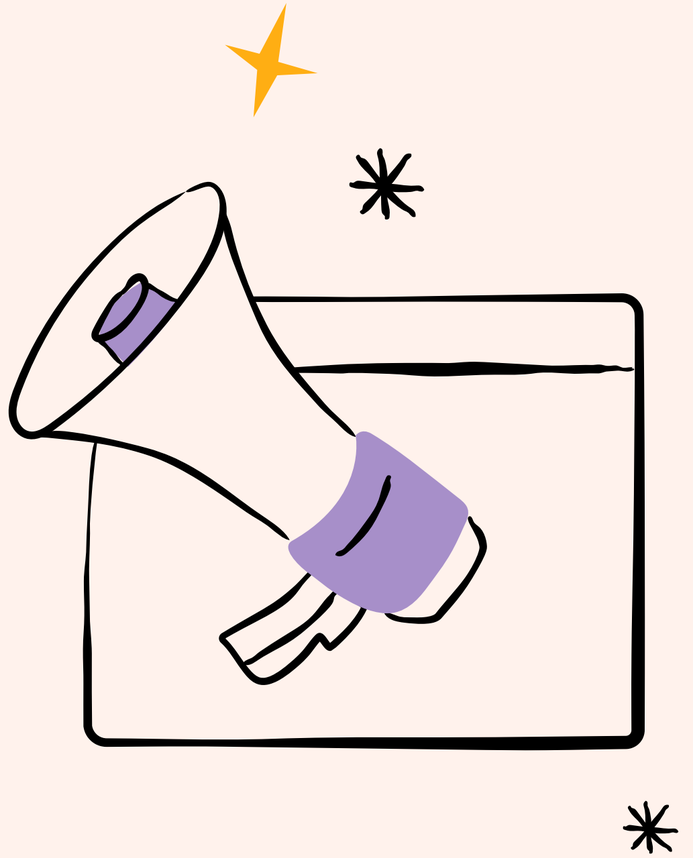
✅ **Use the Stepwise selection with Cross-Validation to predict Engage**

- *Stepwise CV performs best in terms of lowest MSE and second-highest R^2.*
- It balances predictive accuracy (lowest error) and model interpretability.
- *PCA Regression has the highest R^2, but slightly worse MSE than Stepwise.*

Summary of Best Models for View and Engagement Prediction

View - Lasso	Engagement - Stepwise
Positive drivers: • verified status, follower count, male/white features, high image quality **Negative drivers:** • sad/angry/calm emotions, being black or woman, higher age	**Positive drivers:** • verified status, follower count, high image quality, white feature **Negative drivers:** • Likes.sum.Count, happy/calm/sad emotions, age, being black or woman
↑ **Common positive drivers: verified status, follower count, high image quality, being white** ↓ **Common negative drivers: sad/calm emotions, being black or woman**	

Business Implications & Recommendations



Business Implications

Verified Status & Follower Count Drive View & Engage
Creators with verified flags and larger follower bases consistently receive higher views and engagement.

Image Quality Is a Significant Performance Factor
Videos with better visual clarity and quality outperform lower-quality ones on both reach and engagement.

Static Emotions Hurt Performance
Calm, sad, angry, and even happy tones drive fewer views and lower engagement.

Demographic Bias Evident in Performance
“Male” and “White” individuals have higher views and engagement; “Black” and “Woman” labels have lower.

Recommendations for Tiktok

Prioritize Verified, High-Follower Creators

Focus on influencers with verification flags and ≥20K followers; help promising micro-creators get verified.

Provide a Visual Toolkit for Better Visual Quality

Offer more adjustable editing functions, filters and stickers, plus small stipends for polished content.

Inspire Engaging Openings & Continue Testing

Guide creators to start with eye-catching, energetic hooks. Continue testing which emotional tones drive better performance, and get more accurate emotions data.

Support Inclusive Amplification

Reserve budget to promote Black and female creators with paid support and creative guidance.

Thank You for Listening !!!

R-code

```
tiktok$Engage <- tiktok$Share.Count + tiktok$Like.Count + tiktok$Comment.Count
clean_tiktok <- na.omit(tiktok)
set.seed(123)
train_idx <- sample(1:nrow(clean_tiktok), 0.7 * nrow(clean_tiktok))
train_data <- clean_tiktok[train_idx, ]
test_data  <- clean_tiktok[-train_idx, ]

fit.lm_view = lm(log(View.Count+1)~Follower.Count + Likes.sum.Count + Following.Count + as.factor(Verified.Status)
            + image_quality + Brightness + Sharpness + as.factor(Smile)+ Emotion_SURPRISED+Emotion_HAPPY
            +Emotion_CALM+Emotion_FEAR+Emotion_CONFUSED+Emotion_ANGRY+Emotion_SAD+Emotion_DISGUSTED
            +sentimentality_score_0_to_1 +age + as.factor(gender.deepface) + as.factor(gender.amazon) + as.factor(race),
train_data)
summary(fit.lm_view)



fit.lm_view2 = lm(View.Count ~Follower.Count + Likes.sum.Count + Following.Count + as.factor(Verified.Status)
             + image_quality + Brightness + Sharpness + as.factor(Smile)+ Emotion_SURPRISED+Emotion_HAPPY

+Emotion_CALM+Emotion_FEAR+Emotion_CONFUSED+Emotion_ANGRY+Emotion_SAD+Emotion_DISGUSTED
             +sentimentality_score_0_to_1 +age + as.factor(gender.deepface) + as.factor(gender.amazon) + as.factor(race),
train_data)
summary(fit.lm_view2)


AIC(fit.lm_view, fit.lm_view2)
BIC(fit.lm_view, fit.lm_view2)

fit.lm_engage = lm(log(Engage +1)~Follower.Count + Likes.sum.Count + Following.Count + as.factor(Verified.Status)
             + image_quality + Brightness + Sharpness + as.factor(Smile)+ Emotion_SURPRISED+Emotion_HAPPY

+Emotion_CALM+Emotion_FEAR+Emotion_CONFUSED+Emotion_ANGRY+Emotion_SAD+Emotion_DISGUSTED
             +sentimentality_score_0_to_1 +age + as.factor(gender.deepface) + as.factor(gender.amazon) + as.factor(race),
train_data)
summary(fit.lm_engage)
```

R-code

```
## Split the data into 70/30 train-test dataset

clean_tiktok <- na.omit(tiktok)
set.seed(123)
train_idx <- sample(1:nrow(clean_tiktok), 0.7 * nrow(clean_tiktok))
train_data <- clean_tiktok[train_idx, ]
test_data  <- clean_tiktok[-train_idx, ]


fit.lm_view = lm(log(View.Count+1)~Follower.Count + Likes.sum.Count + Following.Count + as.factor(Verified.Status)
            + image_quality + Brightness + Sharpness + as.factor(Smile)+ Emotion_SURPRISED+Emotion_HAPPY

+Emotion_CALM+Emotion_FEAR+Emotion_CONFUSED+Emotion_ANGRY+Emotion_SAD+Emotion_DISGUSTED
            +sentimentality_score_0_to_1 +age + as.factor(gender.deepface) + as.factor(gender.amazon) + as.factor(race),
train_data)
summary(fit.lm_view)

## Stepwise Regression

# Load caret
library(caret)

# Prepare full model formula
full_formula <- as.formula(log(View.Count + 1) ~
                        Follower.Count + Likes.sum.Count + Following.Count + as.factor(Verified.Status) +
                        image_quality + Brightness + Sharpness + as.factor(Smile) +
                        Emotion_SURPRISED + Emotion_HAPPY + Emotion_CALM + Emotion_FEAR +
                        Emotion_CONFUSED + Emotion_ANGRY + Emotion_SAD + Emotion_DISGUSTED +
                        sentimentality_score_0_to_1 + age +
                        as.factor(gender.deepface) + as.factor(gender.amazon) + as.factor(race))

# Set up training control with 10-fold cross-validation
ctrl <- trainControl(method = "cv", number = 10)

# Fit stepwise model using AIC as selection method via caret + MASS
set.seed(123)
stepwise_model_cv <- train(
 form = full_formula,
 data = train_data,
 method = "leapSeq", # Forward/stepwise selection
 tuneGrid = data.frame(nvmax = 1:25), # Try all subset sizes
 trControl = ctrl
)


# Final model selected
best_vars <- coef(stepwise_model_cv$finalModel, stepwise_model_cv$bestTune$nvmax)
print(best_vars)

# Use those variables to refit model on full training data
selected_vars <- names(best_vars)[-1]  # remove intercept
stepwise_formula <- as.formula(log(View.Count + 1)~Follower.Count + Likes.sum.Count + as.factor(Verified.Status)
                        +Emotion_CALM+Emotion_SAD+age  + as.factor(gender.amazon) + as.factor(race))

final_stepwise_model <- lm(stepwise_formula, data = train_data)
summary(final_stepwise_model)

# Predict on test set
Y_test <- log(test_data$View.Count + 1)
pred_test <- predict(final_stepwise_model, newdata = test_data)

# Evaluate
mse_test <- mean((Y_test - pred_test)^2)
rsq_test <- 1 - sum((Y_test - pred_test)^2) / sum((Y_test - mean(Y_test))^2)

cat("Stepwise CV model - Test MSE:", mse_test, "\n") #7.137873
cat("Stepwise CV model - Test R²:", rsq_test, "\n") #0.09220836
```

R-code

```
## lasso
library(glmnet)

# Prepare X matrix
formula_lasso_view <- as.formula(log(View.Count + 1) ~
                                   Follower.Count + Likes.sum.Count + Following.Count + as.factor(Verified.Status)
                                 + image_quality + Brightness + Sharpness + as.factor(Smile)+
Emotion_SURPRISED+Emotion_HAPPY

+Emotion_CALM+Emotion_FEAR+Emotion_CONFUSED+Emotion_ANGRY+Emotion_SAD+Emotion_DISG
USTED
                                 +sentimentality_score_0_to_1 +age + as.factor(gender.deepface) +
as.factor(gender.amazon) + as.factor(race)
)


X_view_train <- model.matrix(formula_lasso_view, train_data)
X_view_test <- model.matrix(formula_lasso_view, test_data)

y_view_train <- log(train_data$View.Count + 1)
y_view_test <- log(test_data$View.Count + 1)

lasso_model_view_train <- glmnet(X_view_train, y_view_train, alpha = 1)  # alpha = 1 for Lasso
lasso_model_view_test <- glmnet(X_view_test, y_view_test, alpha = 1)  # alpha = 1 for Lasso

set.seed(123)
cv_lasso_view <- cv.glmnet(X_view_train, y_view_train, alpha = 1)


# # sparser model as lambda.1se uses a larger penalty (λ) than lambda.min:
simpler_lambda_lasso_view <- cv_lasso_view$lambda.1se  # Within 1 SE of lowest MSE

final_lasso_view <- glmnet(X_view, y_view, alpha = 1, lambda = simpler_lambda_lasso_view)
# coefficients
coefficients_lasso_view <- coef(final_lasso_view, s = simpler_lambda_lasso_view)
print(coefficients_lasso_view)


## on test set
lasso_pred <- predict(cv_lasso_view, s = "lambda.1se", newx = X_view_test)

mse_lasso <- mean((y_view_test - lasso_pred)^2)
rsq_lasso <- 1 - sum((y_view_test - lasso_pred)^2) / sum((y_view_test - mean(y_view_test))^2)

mse_lasso ## 6.856228
rsq_lasso ## 0.1280279
```

R-code

```
## PCA

X_vars <- c(
    "Follower.Count",    "Likes.sum.Count", "Following.Count", "Verified.Status",
    "image_quality",     "Brightness",      "Sharpness",      "Smile",
    "Emotion_SURPRISED", "Emotion_HAPPY",   "Emotion_CALM",   "Emotion_FEAR",
    "Emotion_CONFUSED",  "Emotion_ANGRY",   "Emotion_SAD",    "Emotion_DISGUSTED",
    "sentimentality_score_0_to_1", "age",
    "gender.deepface",   "gender.amazon",   "race"
)

# Build model matrix for training predictors

pca_train_matrix <- model.matrix(~ . - 1, data = train_data[, X_vars, drop = FALSE])

# Fit PCA on training set only
pca_result <- prcomp(pca_train_matrix, center = TRUE, scale. = TRUE)




pca_data_train <- train_data[, X_vars]
pca_data_test  <- test_data[, X_vars]

X_pca_train <- model.matrix(~ . - 1, data = pca_data_train)
X_pca_test  <- model.matrix(~ . - 1, data = pca_data_test)

pca_model <- prcomp(X_pca_train, center = TRUE, scale. = TRUE)

eigenvalues <- pca_model$sdev^2
percent_variance_explained <- (eigenvalues / sum(eigenvalues)) * 100

# Scree-plot of % variance explained
barplot(percent_variance_explained,
        names.arg = paste0("PC", seq_along(percent_variance_explained)),
        main = "Scree Plot (% Variance Explained)",
        xlab = "Principal Component",
        ylab = "Percentage of Variance Explained",
        ylim = c(0, max(percent_variance_explained) + 1),
        col = "skyblue")

print(summary(pca_model))
```

```
# keep the first 15 PCs (    80%+ of variance)
num_pcs_to_keep <- 15



# Project data onto PCs
train_pcs <- predict(pca_model, newdata = X_pca_train)[, 1:num_pcs_to_keep]
test_pcs  <- predict(pca_model, newdata = X_pca_test)[, 1:num_pcs_to_keep]

df_pca_train <- data.frame(Y = Y_train, train_pcs)
df_pca_test  <- data.frame(test_pcs)



# coefficient
# Fit a linear model on the 15 PCs using the training data
pca_reg <- lm(Y ~ ., data = df_pca_train)
summary(pca_reg)

## on test set
pca_pred <- predict(pca_reg, newdata = df_pca_test)

mse_pca <- mean((y_view_test - pca_pred)^2)
rsq_pca <- 1 - sum((y_view_test - pca_pred)^2) / sum((y_view_test - mean(y_view_test))^2)

mse_pca  # 7.009429
rsq_pca  #0.1085438
```

R-code

```
## engage

## Weeks 1–4: regression (using Engage)
tiktok$Engage <- tiktok$Share.Count + tiktok$Like.Count + tiktok$Comment.Count

clean_tiktok <- na.omit(tiktok)
set.seed(123)
train_idx <- sample(1:nrow(clean_tiktok), 0.7 * nrow(clean_tiktok))
train_data <- clean_tiktok[train_idx, ]
test_data  <- clean_tiktok[-train_idx, ]
## Stepwise Regression

# Load caret
library(caret)

# Prepare full model formula
full_formula_engage <- as.formula(log(Engage + 1) ~
                        Follower.Count + Likes.sum.Count + Following.Count + as.factor(Verified.Status) +
                        image_quality + Brightness + Sharpness + as.factor(Smile) +
                        Emotion_SURPRISED + Emotion_HAPPY + Emotion_CALM + Emotion_FEAR +
                        Emotion_CONFUSED + Emotion_ANGRY + Emotion_SAD + Emotion_DISGUSTED +
                        sentimentality_score_0_to_1 + age +
                        as.factor(gender.deepface) + as.factor(gender.amazon) + as.factor(race))


# Set up training control with 10-fold cross-validation
ctrl <- trainControl(method = "cv", number = 10)

# Stepwise Engage
set.seed(123)
stepwise_model_cv_engage <- train(
  form = full_formula_engage,
  data = train_data,
  method = "leapSeq",                    # Forward/stepwise selection
  tuneGrid = data.frame(nvmax = 1:25),       # Try all subset sizes
  trControl = ctrl
)
```

```
# Final model selected
best_vars_engage <- coef(stepwise_model_cv_engage$finalModel,
                         stepwise_model_cv_engage$bestTune$nvmax)
print(best_vars_engage)

# Use those variables to refit model on full training data
selected_vars_engage <- names(best_vars_engage)[-1]  # remove intercept
stepwise_formula_engage <- as.formula(log(Engage + 1) ~
                             Follower.Count + Likes.sum.Count + as.factor(Verified.Status) +
                             image_quality + Sharpness + Emotion_HAPPY + Emotion_CALM + Emotion_FEAR
                           + Emotion_ANGRY + Emotion_SAD + Emotion_DISGUSTED +
                             sentimentality_score_0_to_1 + age +
                             as.factor(gender.deepface) + as.factor(race))

final_stepwise_model_engage <- lm(stepwise_formula_engage, data = train_data)
summary(final_stepwise_model_engage)

# Predict on test set
Y_test_engage <- log(test_data$Engage + 1)
pred_test_engage <- predict(final_stepwise_model_engage, newdata = test_data)

# Evaluate
mse_test_engage <- mean((Y_test_engage - pred_test_engage)^2)
rsq_test_engage <- 1 - sum((Y_test_engage - pred_test_engage)^2) /
  sum((Y_test_engage - mean(Y_test_engage))^2)

cat("Stepwise CV model - Test MSE (Engage):", mse_test_engage, "\n") #4.03368
cat("Stepwise CV model - Test R² (Engage):", rsq_test_engage, "\n") #0.09149395
```

R-code

```
# Lasso engage

## Lasso (using Engage instead of View.Count)
library(glmnet)

# Ensure Engage is defined
# tiktok$Engage <- tiktok$Share.Count + tiktok$Like.Count + tiktok$Comment.Count

# Prepare formula with Engage
formula_lasso_engage <- as.formula(
  log(Engage + 1) ~ Follower.Count + Likes.sum.Count + Following.Count + as.factor(Verified.Status)
  + image_quality + Brightness + Sharpness + as.factor(Smile)+ Emotion_SURPRISED+Emotion_HAPPY
  +Emotion_CALM+Emotion_FEAR+Emotion_CONFUSED+Emotion_ANGRY+Emotion_SAD+Emotion_DISGUSTED
  +sentimentality_score_0_to_1 +age + as.factor(gender.deepface) + as.factor(gender.amazon) + as.factor(race)
)

# Build design matrices for train/test
X_train_engage <- model.matrix(formula_lasso_engage, train_data)
X_test_engage  <- model.matrix(formula_lasso_engage, test_data)

# Create response vectors (log-transformed Engage)
y_train_engage <- log(train_data$Engage + 1)
y_test_engage  <- log(test_data$Engage + 1)

# Fit Lasso models
lasso_model_engage_train <- glmnet(X_train_engage,y_train_engage,alpha = 1)

lasso_model_engage_test <- glmnet(X_test_engage,y_test_engage,alpha = 1)

set.seed(123)
cv_lasso_engage <- cv.glmnet(
  X_train_engage,
  y_train_engage,
  alpha = 1
)

# Choose the “1 SE” lambda for a sparser model
simpler_lambda_lasso_engage <- cv_lasso_engage$lambda.1se

final_lasso_engage <- glmnet(
  X_train_engage,
  y_train_engage,
  alpha  = 1,
  lambda = simpler_lambda_lasso_engage
)

# Extract coefficients at lambda.1se
coefficients_lasso_engage <- coef(final_lasso_engage, s = simpler_lambda_lasso_engage)
print(coefficients_lasso_engage)

## On the test set
lasso_pred_engage <- predict(
  cv_lasso_engage,
  s    = "lambda.1se",
  newx = X_test_engage
)

mse_lasso_engage <- mean((y_test_engage - lasso_pred_engage)^2)
rsq_lasso_engage <- 1 - sum((y_test_engage - lasso_pred_engage)^2) /
  sum((y_test_engage - mean(y_test_engage))^2)

# Print metrics
mse_lasso_engage #4.484075
rsq_lasso_engage #0.06387005
```

R-code

```
## PCA Engage

## 1) Define the predictor columns for PCA
X_vars <- c(
   "Follower.Count",     "Likes.sum.Count",   "Following.Count",   "Verified.Status",
   "image_quality",      "Brightness",        "Sharpness",         "Smile",
   "Emotion_SURPRISED",  "Emotion_HAPPY",     "Emotion_CALM",      "Emotion_FEAR",
   "Emotion_CONFUSED",   "Emotion_ANGRY",     "Emotion_SAD",       "Emotion_DISGUSTED",
   "sentimentality_score_0_to_1",  "age",
   "gender.deepface",    "gender.amazon",     "race"
)

## 2) Build model matrices for train/test (for PCA predictors)
pca_train_matrix_engage <- model.matrix(
   ~ . - 1,
   data = train_data[, X_vars, drop = FALSE]
)
pca_test_matrix_engage <- model.matrix(
   ~ . - 1,
   data = test_data[, X_vars, drop = FALSE]
)

## 3) Fit PCA on the training-only predictors
pca_model_engage <- prcomp(
   pca_train_matrix_engage,
   center = TRUE,
   scale. = TRUE
)

## 4) Examine eigenvalues and percent variance explained
eigenvalues_engage <- pca_model_engage$sdev^2
percent_variance_explained_engage <- (eigenvalues_engage / sum(eigenvalues_engage)) * 100

# (Optional) Scree-plot
barplot(
 percent_variance_explained_engage,
 names.arg = paste0("PC", seq_along(percent_variance_explained_engage)),
 main = "Scree Plot (% Variance Explained) – Engage",
 xlab = "Principal Component",
 ylab = "Percentage of Variance Explained",
 ylim = c(0, max(percent_variance_explained_engage) + 1),
 col = "skyblue"
)

print(summary(pca_model_engage))

## 5) Choose how many PCs to keep (e.g., first 15 for ≈80%+ variance)
num_pcs_to_keep_engage <- 15

## 6) Project both train & test data onto the first 15 PCs
train_pcs_engage <- predict(
   pca_model_engage,
   newdata = pca_train_matrix_engage
)[, 1:num_pcs_to_keep_engage]

test_pcs_engage <- predict(
   pca_model_engage,
   newdata = pca_test_matrix_engage
)[, 1:num_pcs_to_keep_engage]
## 7) Construct new data frames for regression
df_pca_train_engage <- data.frame(
   Y_engage    = Y_train_engage,
   train_pcs_engage
)

df_pca_test_engage <- data.frame(
   test_pcs_engage
)
## Coeffiecient
## 8) Fit a linear model on the retained PCs (using training data)
pca_reg_engage <- lm(
   Y_engage ~ .,
   data = df_pca_train_engage
)
summary(pca_reg_engage)

## 9) Predict on the test set & compute metrics
pca_pred_engage <- predict(
   pca_reg_engage,
   newdata = df_pca_test_engage
)

mse_pca_engage <- mean((Y_test_engage - pca_pred_engage)^2)

rsq_pca_engage <- 1 - sum((Y_test_engage - pca_pred_engage)^2) /
   sum((Y_test_engage - mean(Y_test_engage))^2)

# Print out MSE and R²
mse_pca_engage #4.27625
rsq_pca_engage #0.1072572
```