Deciphering Facebook Engagement: A Predictive Analysis of Consumer Interaction Metrics

Lab Section: C3
Team Members: Jack Campbell, Zhan Ding, Ji Hoon Lee, Jarrett Petto, Wei Shao, Mengyan Sun

Abstract

This report offers a predictive analysis of user interactions with Facebook posts, using a dataset encompassing various engagement metrics from a cosmetics brand`s page. The study aims to construct a model to forecast the `Lifetime Post Consumers` count by examining eight specific input features related to post content and timing. Employing linear regression techniques, we analyze posts to identify significant predictors of consumer engagement. The results confirm that certain variables, such as post type and hour, significantly affect user interactions. The model demonstrates proficiency in predicting user engagement, providing insights for optimizing social media marketing strategies.

Introduction

In the evolving landscape of social media marketing, understanding consumer behavior is pivotal for crafting effective strategies. This report delves into a comprehensive analysis of Facebook metrics, aiming to construct a predictive model based on historical data. The data comprises user interactions with posts from a well-known cosmetics brand`s Facebook page, covering various performance metrics.

Our research is rooted in the dataset labeled `facebook_updated.csv`, which separates observations into `Training` and `Validation` groups for model construction and evaluation. The primary goal is to predict `Lifetime Post Consumers`—a metric defining the number of unique users who have interacted with a post. To achieve this, we utilize eight variables: `TypePhoto`, `TypeStatus`, `TypeVideo`, `Paid`, `ln.Page.Total.Likes`, `ln.Page.Total.Likes:seasonspring`, `ln.Page.Total.Likes:seasonsummer`, and `ln.Page.Total.Likes:seasonwinter`. These predictors are selected based on insights from a prior analysis, including interaction terms between `Page Total Likes` and dummy variable `Season`.

This study aims to uncover the underlying patterns in user engagement and identify significant variables that influence consumer interaction on Facebook. By doing so, we aspire to provide actionable insights for marketers to optimize their content strategy and enhance user engagement.

Data

The dataset comprises posts published in 2014 on the Facebook page of a well-known cosmetics brand. It contains data points for 500 posts, represented across 20 columns—with one that divides the observations into `Training` and `Validation` subsets. The six categorical variables are `Type`, `Category`, `Post Month`, `Post Weekday`, `Post Hour`, and `Paid`: a binary feature indicating whether a post was paid to be promoted.. These variables are our primary predictors of interest and have been manually examined to ensure accuracy, i.e., eliminating erroneous entries such as '13' for `Post Month`. `Page Total Likes` is known before the post is published, and the remaining 12 are used to evaluate post impact. `Lifetime Post Total Reach` and `Lifetime Post Total Impressions` quantify visualizations, while `Lifetime Post Consumers`, the number of people who clicked anywhere on a post, measures user interaction and is also our response variable of interest.

Many of the variables included in the dataset are highly correlated and provide similar insights, giving rise to an issue of multicollinearity. `Lifetime Post Consumers` and `Lifetime Engaged Users` both measure the number of people who clicked anywhere in a post, for example, emphasizing the need for careful variable selection throughout the modeling process. It is discovered that some numerical variables cannot be normalized, such as the variable `Page Total Likes`, which may impede our normality assumption and modeling results.

Preprocessing

When preprocessing this dataset, the R function md.pattern() is utilized to display rows with missing values. We discover a missing value in the `Paid` variable and decide to remove the row containing this missing value. Since `Paid` is a dummy variable with values of '0' and '1', replacing the missing value with '0' would not be reasonable. Numerical values such as `Page Total Likes` and `Lifetime Post Consumers` are log transformed to reduce large values; the group sees improvement in the distribution for `Lifetime Post Consumers`, but no improvement in the distribution for `Page Total Likes`.

All relevant categorical variables, such as `Category`, are converted to factors. Following this, two new categorical variables, `Season` and `Weekday,` are introduced based on the `Post Month` and `Post Weekday` variables, respectively. The `Season` variable categorizes posts into seasons “winter”, “spring”, “summer”, “autumn”, while the `Weekday` variable distinguishes between “Weekday” and “Weekend” as binary values in that order. `Worktime` is created as well to denote whether a post was made during working hours, 9 am to 6 pm, assigning a value of 1 if true and 0 otherwise.

Furthermore, an investigation into the variation of user engagement across different seasons was conducted. The interaction term `season:ln.Page.Total.likes` was incorporated into the model to examine the effects of `Page Total Likes` on the outcome within each season.

Data Summary and Visualization

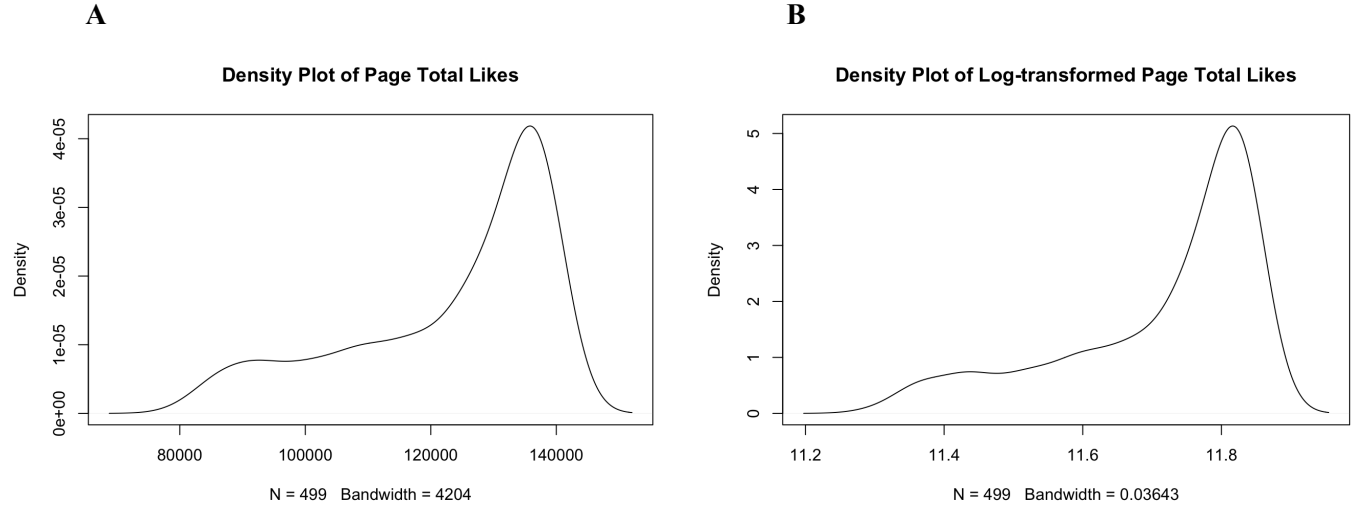


Figure 1. (A) Density Plot of `Page Total Likes`
(B) Density Plot of `ln.Page.Total.likes`

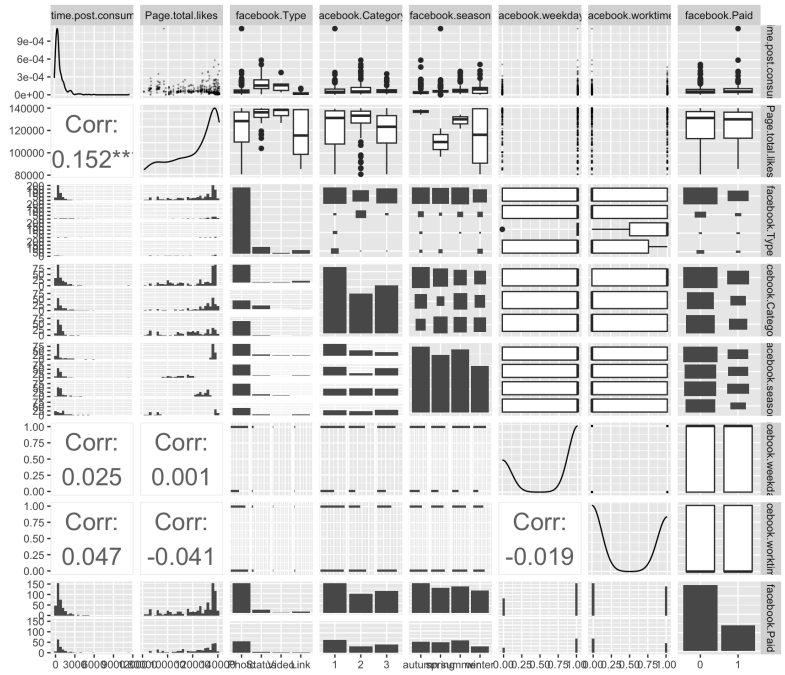


Figure 2. Scatterplot Matrix

Modeling and Analysis

The stepwise function, along with AIC and BIC scores, guided the model selection process. As shown in the table, the model selection process identifies the model with the lowest AIC. After eliminating statistically insignificant variables and addressing multicollinearity issues, the final model is derived. The AIC of this final model is comparable to that of the original model, with the added advantage of a lower BIC.

Full Model

$$\begin{aligned}\ln(\text{Lifetime Post Consumers}) = {} & \beta_0 + \beta_1 * \ln(\text{Page Total Likes}) + \beta_2 * \text{TypePhoto} + \beta_3 * \text{TypeStatus} + \beta_4 * \text{TypeVideo} \\ & + \beta_5 * \text{Paid} + \beta_6 * \text{Category2} + \beta_7 * \text{Category3} \\ & + \beta_8 * \text{seasonspring} + \beta_9 * \text{seasonsummer} + \beta_{10} * \text{seasonwinter} \\ & + \beta_{11} * \text{worktime} + \beta_{12} * \ln(\text{Page Total Likes:TypePhoto}) \\ & + \beta_{13} * \ln(\text{Page Total Likes:TypeStatus}) + \beta_{14} * \ln(\text{Page Total Likes:TypeVideo}) \\ & + \beta_{15} * \ln(\text{Page Total Likes:seasonspring}) + \beta_{16} * \ln(\text{Page Total Likes:seasonsummer}) \\ & + \beta_{17} * \ln(\text{Page Total Likes:seasonwinter}) + \beta_{18} * \ln(\text{Page Total Likes:worktime}) + \varepsilon\end{aligned}$$

Reduced Model

$$\begin{aligned}y = \ln(\text{Lifetime Post Consumers}) = {} & \beta_0 + \beta_1 * \text{TypePhoto} + \beta_2 * \text{TypeStatus} + \beta_3 * \text{TypeVideo} \\ & + \beta_4 * \text{Paid} + \beta_5 * \ln(\text{Page Total Likes}) \\ & + \beta_6 * \ln(\text{Page Total Likes:seasonspring}) + \beta_7 * \ln(\text{Page Total Likes:seasonsummer}) \\ & + \beta_8 * \ln(\text{Page Total Likes:seasonwinter}) + \varepsilon\end{aligned}$$

Analysis

The p-values associated with the coefficients indicate whether the estimated effect of each predictor is statistically significant. For instance, all post types, “Photo”, “Status”, “Video”, are found to be statistically significant, as their p-values are much smaller than the conventional significance level of 0.05. The p-value for `ln.Page.Total.likes:seasonsummer` is significant, suggesting that the interaction between `ln.Page.Total.likes` and the summer season has a significant effect on `ln.Lifetime.Post.Consumers`. The small p-value, < 2.2e-16, for the entire model also suggests that the model is statistically significant. The residual standard error is

0.7458, providing a measure of the typical difference between observed and predicted values. Adjusted R^2, 0.3314, is adjusted for the number of predictors, indicating that approximately 33.14% of the variance in `ln.Lifetime.Post.Consumers` is explained by the model. The VIF outputs are around one, suggesting that there is no multicollinearity between the variables.

The Residual vs. Fitted Values Plot and $\sqrt{Standardized\ residuals}$ vs. Fitted values plots reveal that the residuals are not randomly scattered, but demonstrate a vertical pattern. This observation suggests a challenge to the assumption of constant variance, also known as homoscedasticity. The standard Q-Q plot demonstrates a generally fitted line against the prediction, spanning two standard deviations despite an exponential increase and decrease beyond two standard deviations. The leverage plot shows that most of the residuals fall within the -4 to 4 range, except for data point 442, indicating the model is nicely fitted. In fact, all four plots also identified points 422 and 233 as suspected outliers with high residuals and leverage that affected normality. However, upon inspecting the data, these points should not be considered outliers.

The added-variable plots correspond to the coefficient estimates and statistical significance in the output, signifying that `TypePhoto`, `TypeStatus`, `TypeVideo` all have positive linear contribution while `ln.Page.Total.likes` has a negative linear contribution. The variable `Paid` and the interaction term between `ln.Page.Total.likes` and `Season` do not significantly contribute.

Summary

```
##
## Call:
## lm(formula = ln.Lifetime.Post.Consumers ~ Type + Paid + ln.Page.Total.likes +
##     season:ln.Page.Total.likes, data = training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5652 -0.3232  0.0038  0.3809  2.0387
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     34.308220   4.364736   7.860 5.40e-14 ***
## TypePhoto                        1.062581   0.169474   6.270 1.12e-09 ***
## TypeStatus                       2.166097   0.206482  10.490  < 2e-16 ***
## TypeVideo                        2.098057   0.329744   6.363 6.58e-10 ***
## Paid                             0.137225   0.087669   1.565  0.11847
## ln.Page.Total.likes             -2.492655   0.369285  -6.750 6.61e-11 ***
## ln.Page.Total.likes:seasonspring -0.008111   0.011972  -0.677  0.49859
## ln.Page.Total.likes:seasonsummer  0.026671   0.009375   2.845  0.00472 **
## ln.Page.Total.likes:seasonwinter -0.008268   0.011908  -0.694  0.48799
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7458 on 332 degrees of freedom
## Multiple R-squared:  0.3471, Adjusted R-squared:  0.3314
## F-statistic: 22.06 on 8 and 332 DF,  p-value: < 2.2e-16
```

Table 1. Summary of Linear Regression Model

```
##                                GVIF Df GVIF^(1/(2*Df))
## Type                       1.115166  3        1.018333
## Paid                       1.014518  1        1.007233
## ln.Page.Total.likes        1.735933  1        1.317548
## ln.Page.Total.likes:season 1.768911  3        1.099726
```

Table 2. Variation Inflation Factor

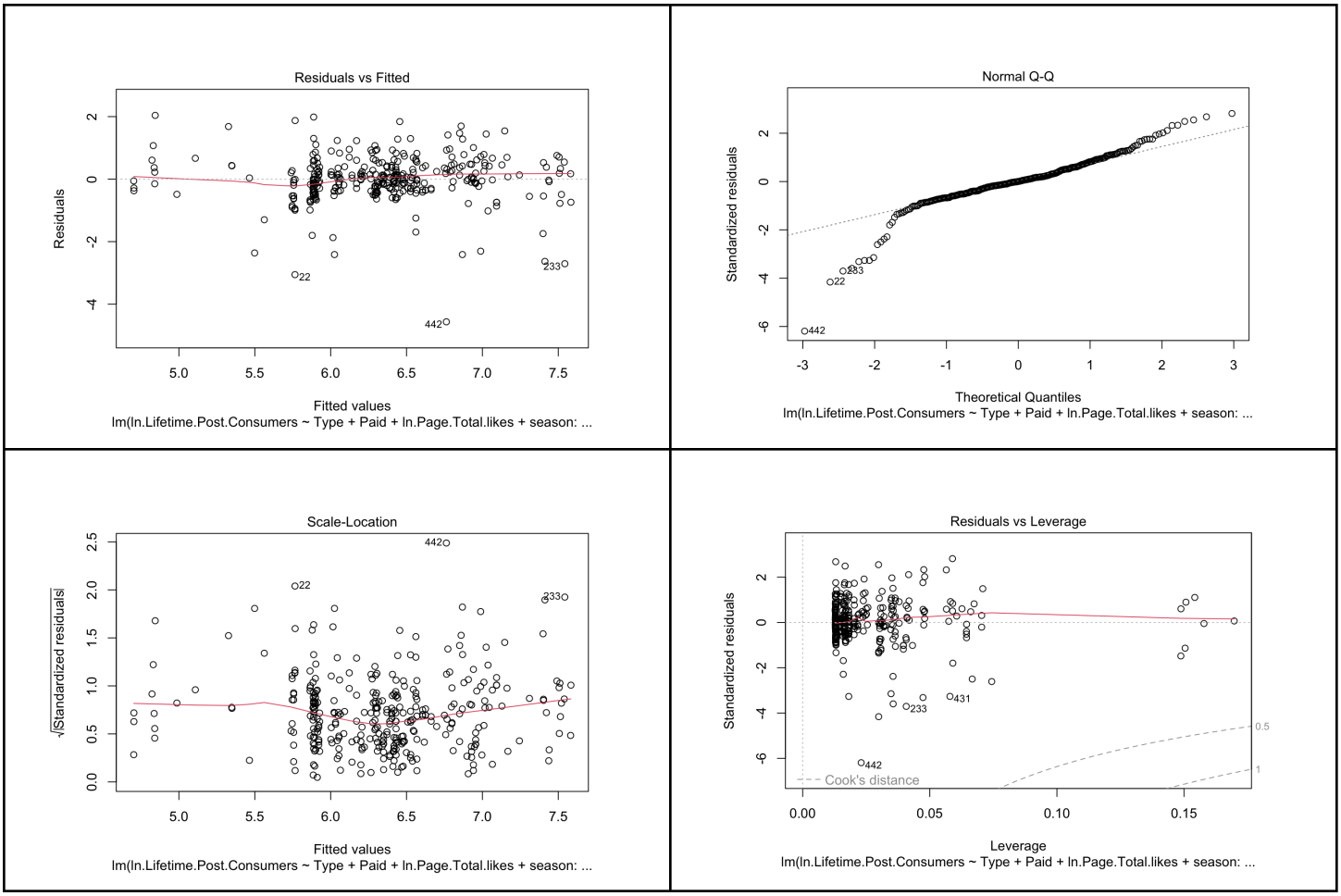


Figure 3. Diagnostic Plots

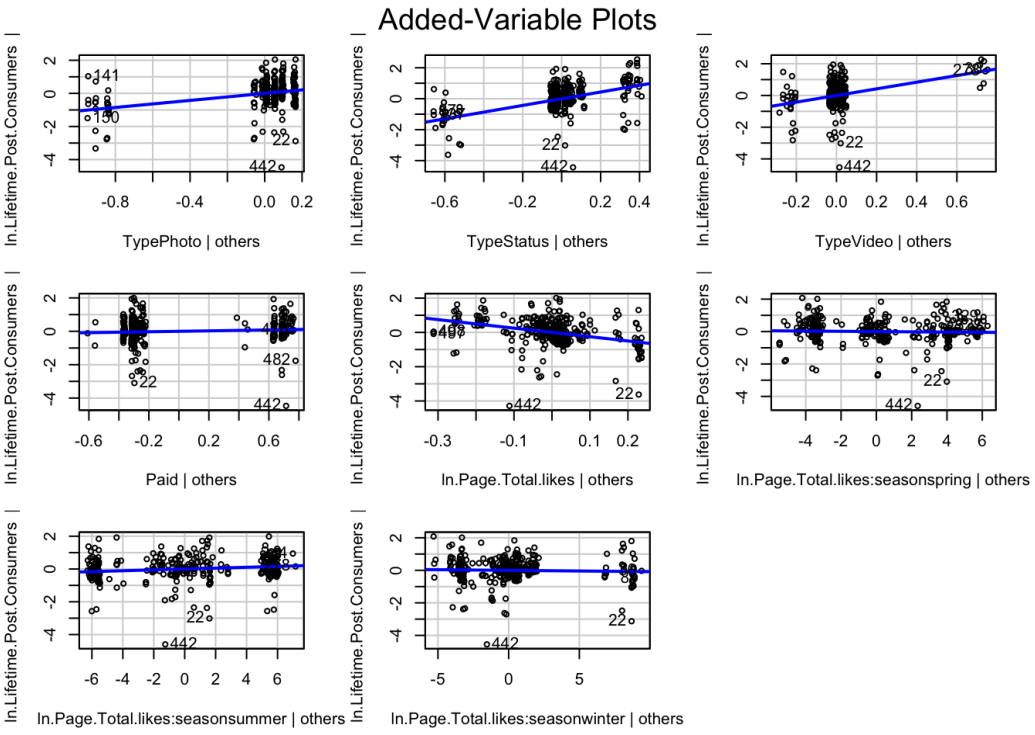


Figure 4. Added-Variable Plots

Lasso Regression and Ridge Regression

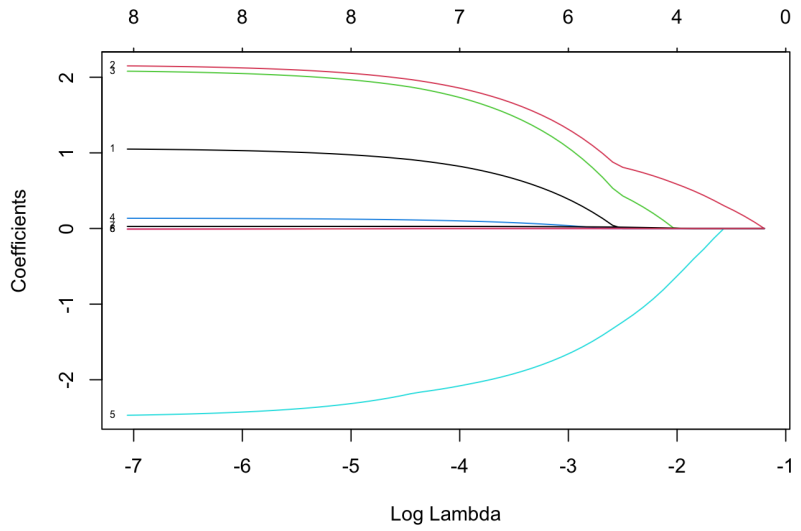
The lasso regression model identified the best regularization parameter (lambda) is 0.00802, which is used to minimize overfitting while retaining predictive power.

From the graph, it is clear that variable `Type`is the best predictor for `Lifetime Post Consumers`; variables `Page Total Likes` and its interaction term with `season` have negative values, posing a more complex picture for the two variables.`TypeStatus` has the strongest potential to explain the variation in `Lifetime Post Consumers`, which is coherent with our analysis of our model.

Best Lambda (λ) : 0.008025502

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                                          s1
## (Intercept)                       31.13903134
## TypePhoto                          0.92425935
## TypeStatus                         1.98708523
## TypeVideo                          1.89042737
## Paid                               0.11590718
## ln.Page.Total.likes               -2.21327168
## ln.Page.Total.likes:seasonspring -0.00106242
## ln.Page.Total.likes:seasonsummer  0.02820889
## ln.Page.Total.likes:seasonwinter -0.00220299
```

Table 3. Lasso Regression



Red Line (2) = `TypeStatus`
Green Line (3) = `TypeVideo`
Black Line (1) = `TypePhoto`
Blue Line (5) =`ln.Page.Total.likes`

Figure 5. Lasso Regression Plot

Prediction

For the training data plot, the points are closely lined up with the red dashed line, indicating that the predicted values closely match the observed values. Moreover, the points do not deviate significantly from the line, indicating that overfitting is unlikely For the validation data plot, the points are more spread out than the training data point randomly, suggesting a less predictive power compared to the training data. Also, since the validation data plot performs worse than the training data, implying that the training data and not capturing underlying patterns The residuals plot, the residuals spread out randomly across the predicted line, indicates that heteroscedasticity is unlikely.

This is the result of using our model to predict the validation dataset. We have MAE = 0.4942, suggesting that the difference between predicted and observed values is around 0.4942. The $R^2 = 0.31$, representing 31.0% of the variance in the observed values explained by the predicted values from the model. From the Observed vs. Predicted Value Plot, the model predicts well, as most points are scattered around the line without systematic deviations. However, there is some variance in the predictions, as indicated by the spread of points, especially for higher observed values. The Residual vs. Observation Index Plot shows that residuals appear to be randomly distributed around the zero line, with no obvious patterns or trends. This suggests that the model does not have systematic errors in its predictions across the range of observations.

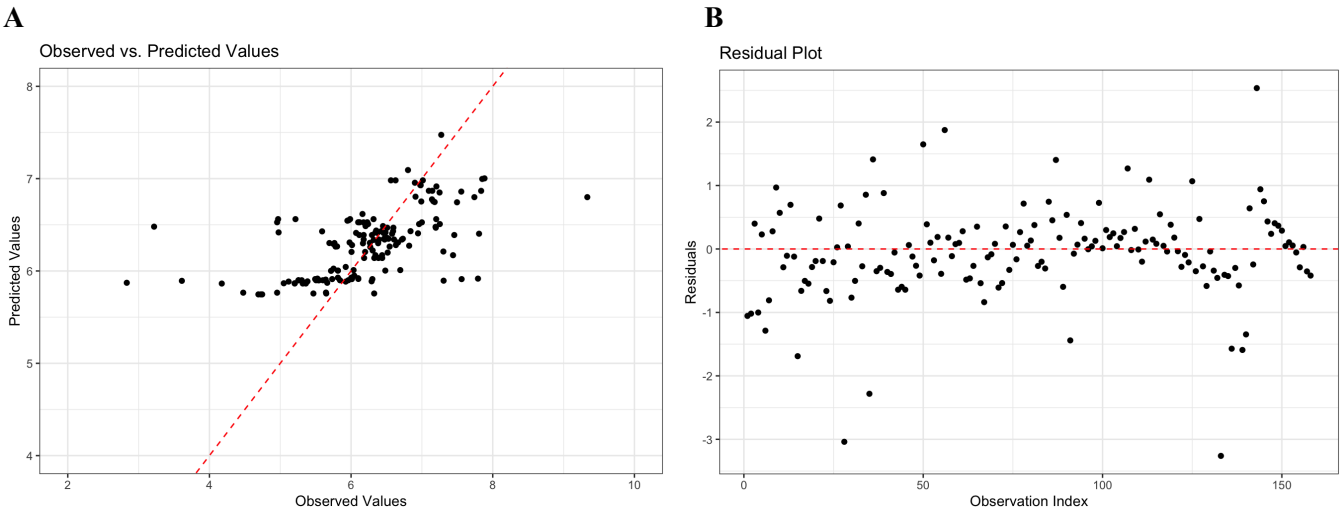


Figure 6. (A) Observed vs. Predicted Values
(B) Residual Plot

Discussion

There are multiple results from this study leading us to conclude that we've come up with a useful model that can predict the `ln.Lifetime.Post.Consumers`. Our AV plots show that many of our variables either have a positive or negative relationship with the `ln.Lifetime.Post.Consumers` and even though some of the variables don't show a positive or negative relationship, they can still contribute to our model and help avoid overfitting. We distinguished the variable `Type` as the most influential in building our model and predicting `ln.Lifetime.Post.Consumers`. Other diagnostics such as our VIF values, p-values of the variables, and the p-value of the model all lead us to conclude that we have a useful model to predict `ln.Lifetime.Post.Consumers`.

Although we've created a model that meets most of our standards as far as the constant variance assumption and having significant predictors, there are still a few limitations of the model. Our R^2 value is fairly low, telling us that our model only represents 33.14% of the data. Additionally, our Normal Q-Q plot has tails that after two standard deviations vary greatly from the expected values and do not follow the linear path we expect it to. This prompts a closer examination of the normality assumption. Despite this, the belief persists in the model's utility. This is attributed to the statistical significance of almost all predictors, the overall statistical significance of the model, and the proper appearance of plots and prediction of the validation data for the most part.

Incorporating knowledge about the timezone and location of the brand's user base could substantially enhance our model's predictive accuracy, particularly concerning the `Worktime` and `Weekday` variables. Understanding when users are most active online due to regional work hours and cultural habits could allow better prediction and capitalization on peak engagement time, thus refining the strategy for post timing and potentially increasing the reach and interaction with the content.

Acknowledgements

Our sincerest appreciation goes to Professor. Julio Castrillon for his invaluable mentorship, astute guidance, and the profound impact on our academic journey through the course of our project.

Our team appreciates the support and insightful feedback from our lab instructor, Shariq Mohammed, which played a crucial role in the successful completion of our project.

Appendix

```
library(tidyverse)
```

```
library(GGally)
```

```
library(car)
```

```
library(ggplot2)
library(MASS)
```

```
library(mice)
```

```
library(MLmetrics)
```

```
library(caret)
```

```
library(glmnet)
```

```
Facebook <- read.csv("facebook_updated.csv", header=TRUE, as.is=TRUE, sep=',')

na_row <- which(is.na(Facebook$Paid)) # remove missing values

Facebook <- Facebook[-na_row, ] # new data set

# Create a Categorical Variable

Facebook$Category <- as.character(Facebook$Category)

Facebook$season <- NA # create a new variable called `season`
Facebook$season[Facebook$Post.Month <= 2] <- "winter"
Facebook$season[Facebook$Post.Month > 11] <- "winter"
Facebook$season[Facebook$Post.Month >= 3 & Facebook$Post.Month < 6] <- "spring"
Facebook$season[Facebook$Post.Month > 5 & Facebook$Post.Month < 9] <- "summer"
Facebook$season[Facebook$Post.Month > 8 & Facebook$Post.Month < 12] <- "autumn"

Facebook$weekday <- NA # create a new variable called `weekday`
Facebook$weekday[Facebook$Post.Weekday < 6] <- 1 # 1 for weekdays
Facebook$weekday[Facebook$Post.Weekday > 5] <- 0 # 0 for weekends

Facebook$worktime <- 0 # 0 for not worktime
Facebook$worktime[Facebook$Post.Hour > 9 & Facebook$Post.Hour < 18] <- 1 # 1 for worktime

# Transformation

Facebook$ln.Page.Total.likes <- log(Facebook$Page.total.likes) # transformation on the variable `Page Total Likes`
with the use of logarithm
Facebook$ln.Lifetime.Post.Consumers <- log(Facebook$Lifetime.Post.Consumers) # transformation on the response vari
able `Lifetime Post Consumers`
```

Code for Figure 1A

```
# Density Plot of Page Total Likes
plot(density(Facebook$Page.total.likes), main = "Density Plot of Page Total Likes")
```

Code for Figure 1B

```
# Density Plot of Log-transformed Page Total Likes
plot(density(Facebook$ln.Page.Total.likes), main = "Density Plot of Log-transformed Page Total Likes")
```

Code for Figure 2

```
# Scatterplot Matrix
co_data <- data.frame(Facebook$`Lifetime.Post.Consumers`, Facebook$`Page.total.likes`, Facebook$`Type`, Facebook$`
Category`, Facebook$`season`, Facebook$`weekday`, Facebook$`worktime`, Facebook$`Paid`)

co_data1 <- data.frame(Facebook$Lifetime.Post.Consumers, Facebook$Page.total.likes, Facebook$Type, Facebook$seaso
n, Facebook$Paid)


ggpairs(co_data,
        upper=list(continuous=wrap("points", alpha=0.3, size=0.1)),
        lower=list(continuous=wrap('cor', size=7)))
```

```
ggpairs(co_data1,
        upper=list(continuous=wrap("points", alpha=0.3, size=0.1)),
        lower=list(continuous=wrap('cor', size=4)))
```

Code for Table 1

```
# Create Training and Validation Data Set

training_data <- subset(Facebook, obs_type == "Training")
validation_data <- subset(Facebook, obs_type == "Validation")

# Model Selection

m.mlr <- lm(ln.Lifetime.Post.Consumers ~ ln.Page.Total.likes*Type + ln.Page.Total.likes*Category + ln.Page.Total.l
ikes*Paid + ln.Page.Total.likes*season + ln.Page.Total.likes*weekday + ln.Page.Total.likes*worktime, data = traini
ng_data) # full model

stepwise <- stepAIC(m.mlr, direction = "both")
```

```
stepwise_model <- lm(ln.Lifetime.Post.Consumers ~ ln.Page.Total.likes + Type + Category + Paid + season + worktime
+ ln.Page.Total.likes:Type + ln.Page.Total.likes:season + ln.Page.Total.likes:worktime, data = training_data) # mo
del based on the stepwise process

summary(stepwise_model)
```

```
m.mlr <- lm(ln.Lifetime.Post.Consumers ~ Type + Paid + ln.Page.Total.likes + season:ln.Page.Total.likes, data = tr
aining_data) # final model

summary(m.mlr)
```

Code for Table 2

```
# Variation Inflation Factor
vif(m.mlr)
```

Code for Figure 3

```
# Diagnostic Plots
plot(m.mlr)
```

Code for Figure 4

```
# Added-Variable Plots
avPlots(m.mlr)
```

Code for Figure 5

```
# Lasso Regression

lasso_regression <- glmnet(x = model.matrix(m.mlr)[,-1], y = training_data$ln.Lifetime.Post.Consumers, alpha = 1)

plot(lasso_regression, xvar = "lambda", label = TRUE)
```

```
cv_model <- cv.glmnet(x = model.matrix(m.mlr)[,-1], y = training_data$ln.Lifetime.Post.Consumers, alpha = 1)

best_lambda <- cv_model$lambda.min # best lambda value
cat("Best Lambda - LASSO:", best_lambda)
```

Code for Table 3

```
lasso_coef <- coef(lasso_regression, s = best_lambda)
print(round(lasso_coef,8))
```

```
# Ridge Regression

ridge_regression <- glmnet(x = model.matrix(m.mlr)[,-1], y = training_data$ln.Lifetime.Post.Consumers, alpha = 0)

plot(ridge_regression, xvar = "lambda", label = TRUE)
```

```
cv_model <- cv.glmnet(x = model.matrix(m.mlr)[,-1], y = training_data$ln.Lifetime.Post.Consumers, alpha = 0)

best_lambda <- cv_model$lambda.min #best lambda value
cat("Best Lambda - Ridge:", best_lambda)
```

```
ridge_coef <- coef(ridge_regression, s = best_lambda)
print(round(ridge_coef,8))
```

```
# Prediction

validation_data$Predicted_ln.Lifetime.Post.Consumers <- predict(m.mlr, newdata = validation_data) # predicting the
response variable `ln.Lifetime.Post.Consumers`

observed_values <- validation_data$ln.Lifetime.Post.Consumers # observed values
predicted_values <- validation_data$Predicted_ln.Lifetime.Post.Consumers # predicted values

rmse <- RMSE(predicted_values, observed_values)
mae <- MAE(predicted_values, observed_values)
r_squared <- R2_Score(predicted_values, observed_values)
```

Code for Figure 6A

```
# Observed vs. Predicted Values Plot
ggplot(validation_data, aes(x = ln.Lifetime.Post.Consumers, y = Predicted_ln.Lifetime.Post.Consumers)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
  labs(x = "Observed Values", y = "Predicted Values", title = "Observed vs. Predicted Values") +
  xlim(c(2,10)) +
  ylim(c(4,8)) +
  theme_bw()
```

Code for Figure 6B

```
# Residual Plot
ggplot(validation_data, aes(x = 1:nrow(validation_data), y = ln.Lifetime.Post.Consumers-Predicted_ln.Lifetime.Pos
t.Consumers)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 0, color = "red", linetype = "dashed") +
  labs(x = "Observation Index", y = "Residuals", title = "Residual Plot") +
  theme_bw()
```