Intelligent Posting: Applications of Linear Models to Social Media Marketing

Kai Bernardini, Jueru Jin, Maria Ren, Sijie Xiang, Alex Brebenel

December 12, 2017

Abstract

This research project seeks to find meaningful relationships between the type of post a particular social media page makes, and the total number of interactions the post will receive. Through different statistical methods, this paper presents our analysis of models for predicting user interactions of posts via seven main effects. The project starts by preprocessing our original dataset "Facebook Performance Metrics", examining and highlighting possible outliers and bad leverage points. Our initial model with ordinary least squares proves to be a misfit for the dataset as only few variables are significant. Going through different diagnostic and variable selection methods including transformation, forward/backward AIC BIC, variable selection processes, etc. we come up with two more approaches toward modeling: Poisson General Linear Model approach and a Ridge Regression Model approach.

Contents

1 Introduction

Since its launch in 2004, Facebook has become one of the most powerful social media sites across the internet. Whether it's sharing photos, advertising, posting status updates, or connecting with friends, it has become a great platform for people to connect with each other in workplaces, schools, and other organizations. This project uses Facebook Social Media Metrics data published by Moro et Al. in 2016, and seeks to analyze different trends among variables trying to find the trend that maximizes total interaction. We will be focusing on analyzing and predicting effect on Total Interaction variable, which is the sum of the number of likes, comments, and shares a posts on the cosmetic page receives. By constructing models with respect to the seven main effects - Category, Page Total Likes, Type, Month, Hour, Weekday, and Paid, we want to interpret and build models to predict our response variable, Total Interaction. This project utilizes various model building techniques to analyze the relationship between Column Space (X) and Y variable. And the primary goal of the project centers around creating the optimal model with the best prediction accuracy.

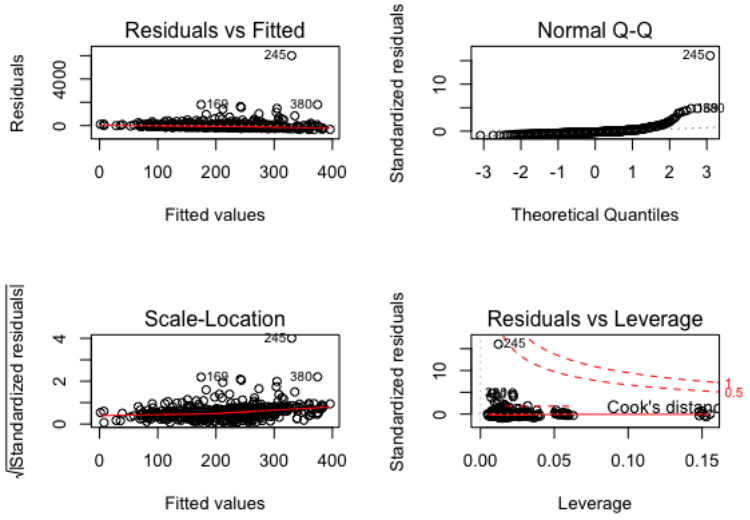
2 Background

The dataset include 500 data points with 19 variables (for this project, we are only focusing on seven main-effects variables and the response variable, including category, page total likes, type, month, hour, weekday, and paid, and response variable Total Interaction). According to the dataset information from the journal Predicting Social Media Performance Metrics and Evaluation of the Impact on Brand Building: A Data Mining Approach by Sergio Moro, Paulo Rita and Bernardo Vala, this data was collected between January 1sts and December 31st of 2014 across a worldwide renowned cosmetic brand Facebook page. The variables we are using for our analysis include both categorical and numerical values:

1. Category (categorical variable) - different contents of the posts, including action(promotion), product(advertisement), and inspiration(non-explicit advertisement for product).
2. Page total likes (numerical variable) - number of people who like the company's page over the year of data collection.
3. Type (categorical variable) - different types of posts, including photo, status, link, and video.
4. Month, hour, weekday(categorical variables) - these three variables refers to the specific time and date of the individual post.
5. Paid (categorical variable) - whether the company paid Facebook for the advertisement of products.
6. Total Interaction (numerical variable) - sum of likes, shares and comments of the posts, indicating popularity of the post.

3 Cleaning the Data

As a first step to analyzing the model, we try to clean the dataset from any possible outliers and bad leverage points. We discovered one specific point that is a significant outlier and six missing values with NA marked from our dataset(two of them in the same row). Because the dataset is fairly large, we decided to eliminate these five observations, knowing it would not drastically affect the rest of the model. Also, we used Boxplot to detect and delete any potential outliers in Total Interactions.



Data point [245] - This individual photo post was published on a Wednesday of July at around 5am. It was a paid advertisement specifically categorized as "promotion", and received 372 comments, 5172 likes, and 790 shares. After fitting our initial OLS regression model, and examining the residual plot shown , point 245 has its residual value and square root of residual value well above all the other data points. It also significantly deviates from the normal line of the Normal Q-Q plot, and has a high cook's distance value. This specific post could have been the company's effort in trying to promote a new cosmetics product for the summer. The research journal from Moro et Al also mentioned the possibility of a promotional post as "contest". This individual post could have been a contest to win a popular product from the company, which could have been the reason why it received the highest amount of total interaction from consumers. The data point negatively affected our result as it significantly skewed the normality of our dataset, we have concluded that it is a possible outlier, and eliminated the point from our dataset. Data point [112,121,125,165] All four data point has missing values in either "like" or "share" column within the total interactions variable. We removed them for our analysis.

As our goal is to find the optimal model to predict the y-response variable, Total Interaction, we need to verify the accuracy of our prediction. In order to do this, we split the dataset into 50 percent training dataset, and 50 percent testing dataset. Now, as we finish data processing, we are ready to do an initial OLS examination, followed by various modeling techniques to find optimal model.

4 Modeling and Analysis

Our modeling process explored four major models:

1. Ordinary least squares model
2. Poisson General Linear Model
3. Negative Binomial General Linear Model
4. Ridge Regression.

4.1 OLS

We first go to the scatter plots and look at the relationships between the variables. Looking at the scatter plots doesn't tell us much about what kind of model to create, being that nearly all of them appear random. The reason for this is most likely because most of the variables are categorical, and for this reason we run an OLS model and work from there to try and create a valid model. After running an OLS model for predicting

```
Call:
lm(formula = Total.Interactions ~ Category + Page.total.likes +
    Type + Post.Month + Post.Hour + Post.Weekday + factor(Paid))

Residuals:
   Min     1Q Median     3Q    Max
-344.0 -137.3  -71.3   13.7 6004.1

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      -2.898e+02  2.908e+02  -0.997   0.3194
Category          5.312e+01  2.096e+01   2.535   0.0116 *
Page.total.likes  3.790e-03  3.176e-03   1.194   0.2332
TypePhoto         6.598e+01  8.626e+01   0.765   0.4447
TypeStatus        6.190e+01  1.028e+02   0.602   0.5472
TypeVideo         1.506e+02  1.664e+02   0.905   0.3662
Post.Month       -1.358e+01  1.571e+01  -0.864   0.3878
Post.Hour        -5.415e-01  4.040e+00  -0.134   0.8934
Post.Weekday     -1.258e+01  8.428e+00  -1.492   0.1362
factor(Paid)1     8.726e+01  3.816e+01   2.287   0.0226 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 377.6 on 485 degrees of freedom
Multiple R-squared:  0.03924,   Adjusted R-squared:  0.02141
F-statistic: 2.201 on 9 and 485 DF,  p-value: 0.02088
```

Total Interactions (with the training set), we found that most of the predictor variables are insignificant. Intuitively, this does not make much sense, but just to confirm this, we'll go through the diagnostics. Looking at the diagnostic plots, we can see a pattern in the standard residual plot and an inconsistency in the Normal Q-Q plot, both of these indicating a violation of the normality assumption, confirming that this model is invalid. Our next step was to try and transform the model to make it valid. We tried the Box- Cox transformations (using inverse and log). Leaving alone the fact that we had to omit all observations with zero Total Interactions from the dataset, these transformations did not make the variables significant. With this in mind, we checked the histogram of the Total Interaction to grasp the overall shape of distribution.

4.2 Poisson Initial Approach

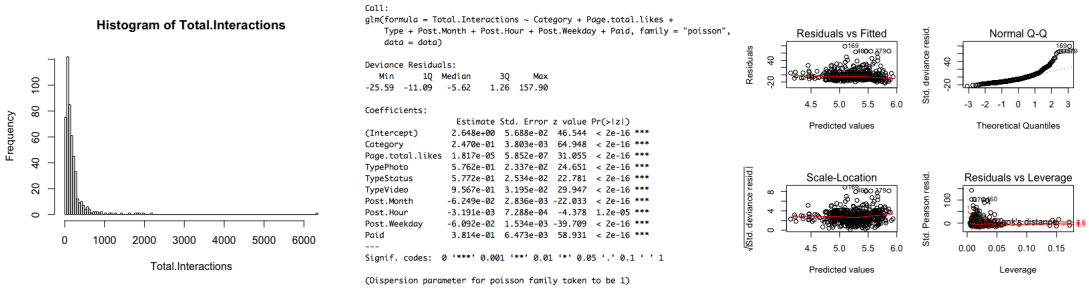
In deciding upon an initial approach for a generalized linear model, we first restricted our search to counting distributions as need support for positive integers. First, we looked at the histogram of the y-response variable Total Interactions. The shape of the graph appears to resemble the shape of a Poisson distribution. We now explore the mathematical motivation for using a Poisson process to model the total number of interactions for a particular post. Lets consider for now a single user on Facebook U. Suppose we are interested in modeling whether or not U_i sees some post P_j. Let X_{ij} be a random variable such that

$$X_{ij} = \begin{cases} 1 & \text{if } U_i \text{ interacts with } P_j \\ 0 & \text{if } U_i \text{ does not interact with } P_j \end{cases}$$

It is not unreasonable to model this random variable experiment as a Bernoulli random variable. We extent this and assume there are now $n \in \mathbb{N}$ users on Facebook $U_1, \ldots, U_n$. It turns out that when a random variable X has a binomial distribution $B(n, p)$ where n is large and p is small, then we can approximate the distribution of that Binomial random variable with a **Poisson Distribution** (see appendix for derivation). Since all the observations are positive integers, we decided to construct a Poisson generalized linear model. Under this model, we make the assumption that

$$\lambda = E[Y|X] = exp(\beta^T X)$$

The goal is to find the β that maximizes the log likelihood. Since there it turns out there is no analytical solution, the derivation is left out.




```
Call:
glm(formula = Total.Interactions ~ Category + Page.total.likes +
    Type + Post.Month + Post.Hour + Post.Weekday + Paid, family = "poisson",
    data = data)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-25.59  -11.09   -5.62    1.26  157.90

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     2.648e+00  5.688e-02  46.544  < 2e-16 ***
Category        2.470e-01  3.803e-03  64.948  < 2e-16 ***
Page.total.likes  1.817e-05  5.852e-07  31.055  < 2e-16 ***
TypePhoto       5.762e-01  2.337e-02  24.651  < 2e-16 ***
TypeStatus      5.772e-01  2.534e-02  22.781  < 2e-16 ***
TypeVideo       9.567e-01  3.195e-02  29.947  < 2e-16 ***
Post.Month     -6.249e-02  2.836e-03 -22.033  < 2e-16 ***
Post.Hour      -3.191e-03  7.288e-04  -4.378  1.2e-05 ***
Post.Weekday   -6.092e-02  1.534e-03 -39.709  < 2e-16 ***
Paid            3.814e-01  6.473e-03  58.931  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)
```

Our initial fit of the Poisson model proves that it is a pretty good approach to our analysis. As shown from the summary table, all the variables turns out to be significant.

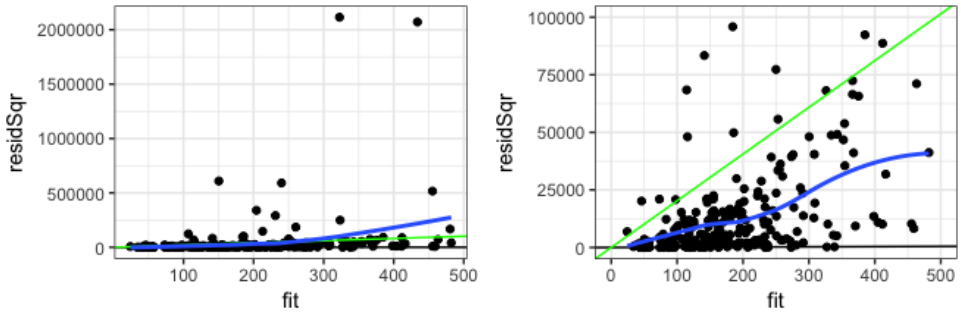
Looking at the diagnostics plots of the Poisson model, the residual vs. predicted values plot shows a significantly smaller range in values than the OLS model, yet the standardized residual plot shows evidence of a slight increase in variance. The normal Q-Q plot has a heavy right tail. Even though all the main effects seems to be significant in the initial glm model, we want to look into the model more carefully since the initial diagnostics shows some problems with the model. The first indication that something is wrong comes form the test mean squared error being significantly higher (2 vs 230 RMSE) than the training rmse. This leads us to reexamine how good a fit the Poisosn GLM is. To check the goodness of fit to this model, we need to first check our assumptions. In particular, we show in the appendix that if $X \sim Poiss(\lambda)$, then

$$E[X] = Var[X] = \lambda$$

The mean total number of interactions turns out to be 185.834710743802 and the variance is 53821.250. As can be seen, this assumption does not hold. The deviance of residuals for this GLM model comes out to be 32228.6 on 208 degree of freedom, associated with a p-value equals to 0. This result indicates the existence of overdispersion and the model is inappropriate. To overcome this problem, we first use T-test to see if any term can be dropped from the model. Three terms are shown to be insignificant: Type, Day and Hour. After we exclude the three terms from the full model, we carry out the same residual deviance test as above. Yet, the problem of overdispersion still remains. We then consider using the quasi-likelihood estimation and negative binomial distribution.

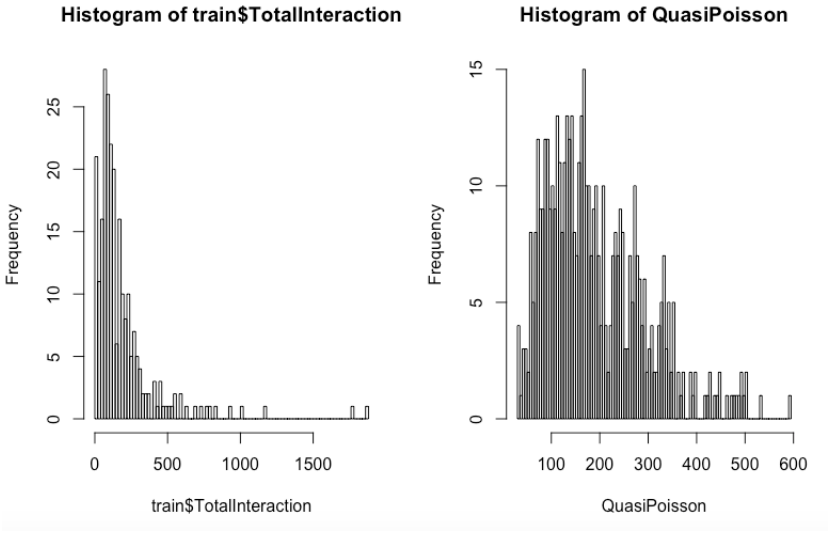
4.3 Quasi-Poisson Approach

The output from R shows that there is no change in the estimated coefficients between the Quasi-Poisson fit and the poisson fit. However, the number of significant terms drops dramatically, with only 10 out of 39 factor levels remain significant. The dispersion parameter is estimated to be 202.975. Applying the "drop-one" F-test, Type, Day and Hour again shown as insignificant. Even removing these three terms, still less than half of the rest factor levels are significant. Intuitively, it is also not quite make sense to exclude Type and Hour from our model, as we think these two factors will have major effect on Total Interactions.



The above scatter plots also indicate quasi-Poisson model is not appropriate. The black line at the bottom is for the Poisson assumed variance, the green line is for the Quasi-Poisson assumed variance, and the blue curve is for the smoothed residual mean square. Ideally the blue curve would be straight and it would be

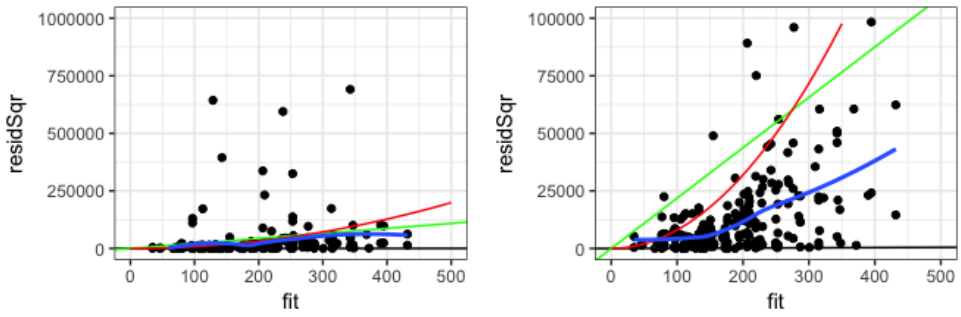
collinear with the green line for the Quasi-Poisson variance. The greater the deviation from the green line the greater the concern is about the proportionality of the variance to the mean. Here we have some indication that the variance may not be proportional to the mean.



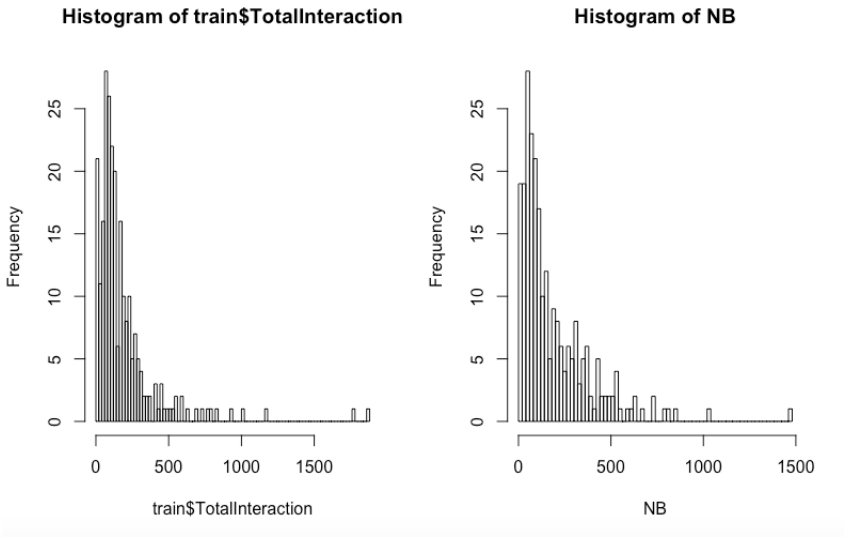
Also, as shown from the figure above, the simulation of counts by this model seems to be overrated, and it does not capture the feature of large Total Interactions. Therefore, we conclude that the Quasi-Poisson model is not appropriate.

4.4 Negative Binomial Poisson Approach

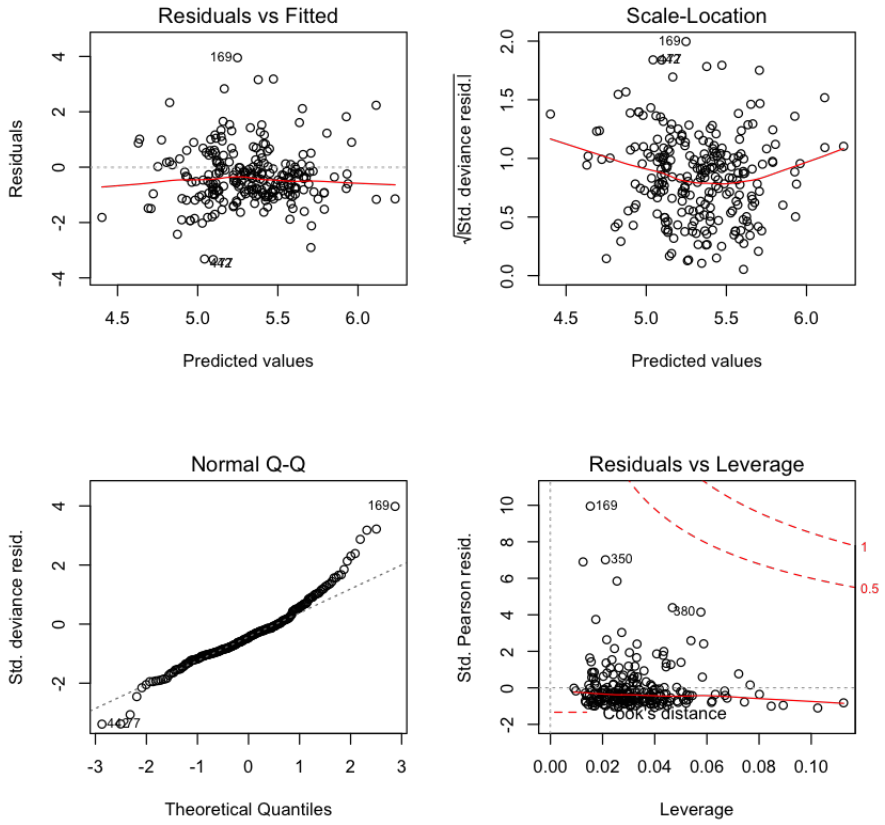
We now look to see if a Negative Binomial model might be a better fit. 15 out of 39 factor levels are significant, which is definitely an improvement from the Quasi-Poisson model. The residual deviance is 281.325 on 229 degree of freedom, comparing to Quasi-Poisson 35792.38 on 231. A likelihood ratio test is then applied for variable selection. Page Total Likes, Day and Hour are not significant, so we remove them from the model. Now, the retained factors in our Negative Binomial Poisson model are: Type, Month, Category & Paid, which all seems reasonable. We will repeat to check the variance of residuals as we have done for the Quasi-Poisson model by adding a red line for predicted variance from the Negative Binomial fit.



The Negative Binomial model is closer to the loss line than to the Quasi-Poisson model. Though the range of values predicted by both models tend to be overstated, the Negative Binomial model is preferred. To check if the Negative Binomial model captures the distribution of Total Interactions, we simulate a sample data from the log rates estimated by the model.

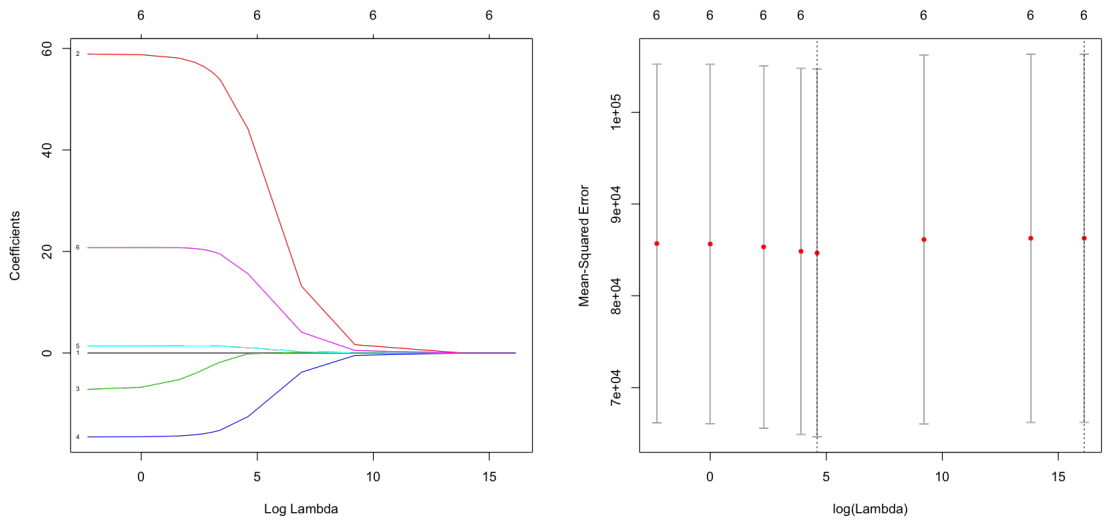


The shape of our simulation lines up with the shape of our target value. With some experimentation with polynomial feature transformation, we arrive at a final model consisting of a temporal interaction term. From there, we check the diagnostic plots of the mode. The residual plots show a random pattern, the Q-Q plot follows a relative straight line, and there is no significant outlier. These all adding up to further confirm the linear relationship between the explaining factors and the log rates of the count.



4.5 Ridge Regression

The final model we employ in this paper is Ridge Regression. Ridge regression uses L2 regularization to penalize residuals when the parameters of a regression model are fit . It is comparable to OLS, in that the optimization still involves learning coefficients to minimize the residual sum of squares. The only difference is instead of taking the global optimum, the space of possible coefficients is restricted to a ball centered at the origin. The exact radius of that ball is determined by Lambda, the regularization parameter. When lambda is chosen effectively, the outcome is typically a model that fits the training data worse than OLS but generalizes better because it is less sensitive to extreme variance in the data such as outliers. In particular, we are constructing a biased estimator for Beta in the hopes of greatly reducing the variance of predictions. In this particular case, it seems to be a reasonable choice, as most models up until this point of suffered from low training RMSE but significantly higher testing RMSE.



As usual, we use 5-fold cross validation to find an optimal choice of lambda with respect to mean squared error. From there, we fit a Ridge regression model using glmnet and score the testing set with respect to RMSE. A value of lambda =10 is chosen, and the resulting testing RMSE is 227.748286775604.

5 Prediction

With the previous sections of this paper extensively discussing the validation of our models, now we use RMSE as a performance metric to measure the prediction accuracy of the variable of interest: Total Interactions. Since our main focus of interest is how much our predictions deviate from the ground truth in the testing set rather than the proportion of the variance in the dependent variable that is predictable from the independent variables in the training set, RMSE would be the ideal metric to use. We started with the initial Poisson Model with an RMSE of 290.782. After balancing the training/testing sets based on Type, we were able optimize our model and reduce the RMSE down to 230.684. However, although this model generated the smallest RMSE, it violated the assumption that the Poisson model should have equal mean and variance. The next logical counting distribution to try was a Negative Binomial Poisson(with no interaction terms) approach to overcome overdispersion.

As seen in the table below, the RMSE for the third Model had increased substantially compared to the second model, which made us try to add interaction terms to better predict to the response variable (Total Interactions). After trying several different combinations, we came across a fourth model which returned the lowest RMSE value so far. Even though Model 4 seems to have the best prediction accuracy (proven by it's low RMSE), there is another issue in that most of the variables become insignificant, which can lead us to believe that Model 4 might not be the best model but is preferable to the Poisson Models. The final model is the ridge regression model. Unlike the other models which has extremely small RMSE on the training set

Table 1: Predictive Results

Model	RMSE
Poisson Model Initial Approach	290.782
Poisson Model Balanced	230.684
Negative Binomial W/O Interaction Term	296.307
Negative Binomial Final	227.364
Ridge Regression	228.436

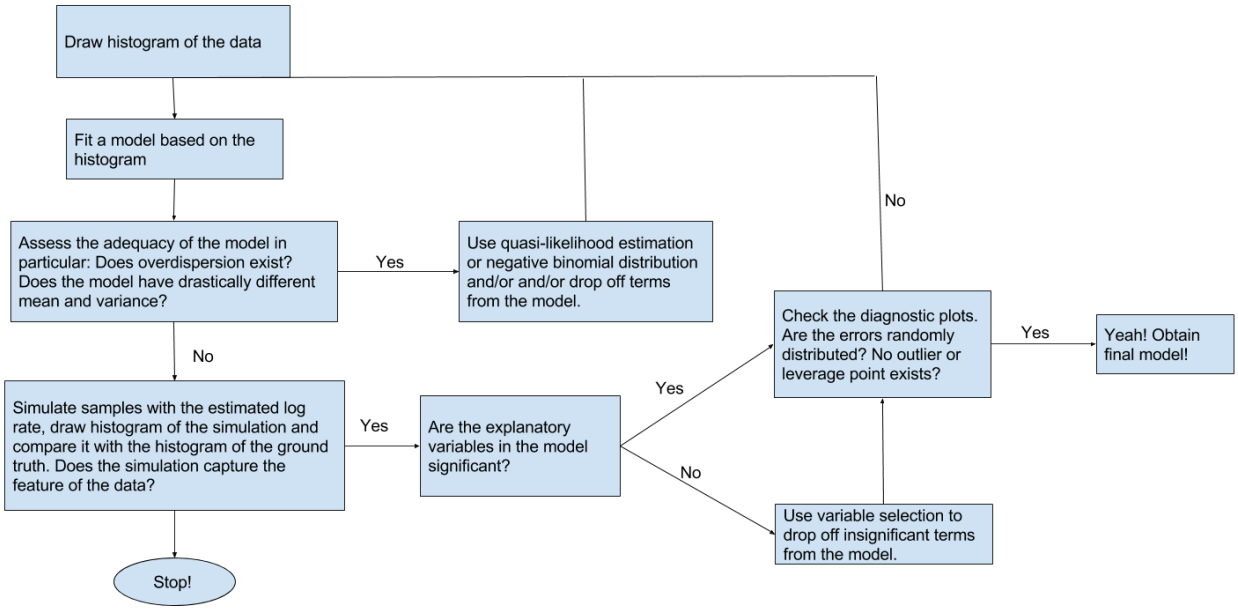
and significantly larger RMSE on the testing set, the RMSE on the CV set and testing set were comparable. This is usually a good indication that the model will generalize well.

6 Discussion

6.1 Reflection

Reflecting back on the project, the biggest challenge that we encountered was finding valid model to fit our data. We tried models ranging form OLS to AR models, LASSO etc., but most models had validity issues. Another challenge encountered had to do with the nature of the data. In particular, 6 of the 7 predictors are dummy variables/factors, and each dummy variable included at least 3 different categories. Trying out various polynomial combination of features and several transformations did not yield promising early results. This lead us to seek methods such as Poisson and NB regression.

Even though we have not covered Poisson regression throughout the semester, we find that the approaches for model analysis are similar for all regressions. The figure belows contains a flowchart that summarizes our steps in developing a Poisson regression model. We will see that it has a similar flavor as the development of multiple linear regression model.



6.2 Revisiting Goals

Our primary goal was two fold. The first goal was to determine which features were important. For the regression models, since the response variable is a positive integer, we can naively interpret feature importance based on the size of the coefficient, and the size of the input feature. The motivation to seek an explanation

for why some posts receive more interactions than others is simple: the reason companies have social media pages is to cheaply and effectively disseminate information. Whether it be an advertisement for a new product, a sale or some other announcement, Facebook provides an excellent platform to reach consumers. In most of our models, we found that the binary attribute paid was indeed statistically significant form zero. This is not surprising, as if we look at the average number of total interactions for a paid post, and an unpaid post in the training set, we see paid mean of 227.183 and an unpaid mean of 210.302. The impact of this project is we can now see with some confidence how this company can reach more people.

The second goal was to find the model with the best prediction RMSE. Even though our final model was able to increase the amount of significant variables drastically, we still were not able to reduce the RMSE to a satisfactory level- Since our knowledge models is limited, and the scope of the class excluded many other possible models we could have tried on the data set.

7 Appendix

Author Contributions: modeling: K.B, M.R, J.J, prediction and analysis: A.B, S.X, Poisson modeling and proofs: K.B, writing the paper: A.B, S.X, J.J, M.R, K.B

7.1 Code

```
#Poisson Model
m1 <- glm(TotalInteraction ~ Followers + Type+Month+Category+Day+Paid+Hour, family="poisson", data=train)
summary(m1)
#res.deviance   df  p
#31823.16      206  0
with(m1, cbind(res.deviance = deviance, df = df.residual,p = pchisq(deviance, df.residual, lower.tail=FALSE)))

drop1(m1, test="F")

#Drop Type, Day and Hour
#res.deviance  df  p
#35792.38      231 0
m2 <- glm(TotalInteraction ~ Followers +Month+Category+Paid, family="poisson", data=train)
summary(m2)
with(m2, cbind(res.deviance = deviance, df = df.residual,p = pchisq(deviance, df.residual, lower.tail=FALSE)))

#Negative binomial model
nb1 <- glm.nb(TotalInteraction ~Followers+Type+Month+Category+Day+Paid+Hour, data=train)
summary(nb1)

drop1(nb1, test="LRT")
#Day & Hour not significant
#Retain everything but Hour &Day in nb2

nb2 <- glm.nb(TotalInteraction ~ Type + Month  + Category + Paid, data=train)
summary(nb2)   #theta1.259
plot(nb2)

#281.325 229 0.0104212
with(nb2, cbind(res.deviance = deviance, df = df.residual,p = pchisq(deviance, df.residual, lower.tail=FALSE)))

NB <- rnegbin(fitted(nb2), theta = nb2$theta)
hist(train$TotalInteraction,100)
hist(NB,100)
```

```r
#Quasi-Poisson Model
quasi1 <- glm(TotalInteraction ~ Followers+Category+Paid, family="quasipoisson", data = train)
summary(quasi1)  #dispersion: 250.3399

par(mfrow = c(1, 2))
QuasiPoisson <- rnbinom(n = 492, mu = fitted(play), size = 5)
hist(train$TotalInteraction,100)
hist(QuasiPoisson, 100)

m1Diag <- data.frame(train,
                     link=predict(m1, type="link"),
                     fit=predict(m1, type="response"),
                     pearson=residuals(m1,type="pearson"),
                     resid=residuals(m1,type="response"),
                     residSqr=residuals(m1,type="response")^2
)
ggplot(data=m1Diag, aes(x=fit, y=residSqr)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  geom_abline(intercept = 0, slope = summary(quasi1)$dispersion,
              color="green") +
  stat_smooth(method="loess", se = FALSE) +
  theme_bw() + xlim(0, 500) +ylim(0, 1e+05)
```

```r

library(gamlss)
library(DAAG)
require(glmnet)
library(alr3)
require(caTools)
library(car)
library(ggfortify)
library(bootstrap)
library(MASS)
library(zoo)
library(ggplot2)

facebook  = read.csv("Data/dataset_Facebook.csv")

ind = 0
z = 0
for(i in 0:500) {
    if(anyNA(facebook[i,])){
        print(i)
    }
}
NA_rows = c(112,121,125,165,500)
NA_rows = c(112,121,125,165,500)
clean_fb <- facebook[-NA_rows,]
anyNA(clean_fb)
outlier_row = c(241)
clean_fb = clean_fb[-c(241) , ]
dim(clean_fb)
attach(clean_fb)

rmse_mod <- function(mod, data){
    mean((data$Total.Interactions - predict(mod, data) )^2)^.5
}
mse <- function(sm)
    mean(sm$residuals^2)

paid_split <- function(data){
    data_paid = data[data$Paid == 1,]
    data_unpaid = data[data$Paid == 0,]
     list(data_paid, data_unpaid)
    }

train_test_split <- function(data){
    set.seed(4242)
    index = sample(1:nrow(data), size = 0.5* nrow(data))
```

```
    train = data[index,]
    test  = data[-index,]
    list(train, test)
}

category_split <- function(data){
    data_link = data[data$Type == 'Link',]
    data_photo = data[data$Type == 'Photo',]
    data_status = data[data$Type == 'Status',]
    list(data_link, data_photo, data_status)
    }



tmp <- train_test_split(clean_fb)
train <- tmp[[1]]
test <- tmp[[2]]

split_data <- category_split(train)
split_test <-  category_split(test)


train_balance  <- rbind(split_data[[1]], split_data[[2]],split_data[[3]])
test_balance  <- rbind(split_test[[1]], split_test[[2]],split_test[[3]])

paid_split <- function(data){
    data_paid = data[data$Paid == 1,]
    data_unpaid = data[data$Paid == 0,]
     list(data_paid, data_unpaid)
    }

paid_unpaid <- paid_split(train_balance)
mean(paid_unpaid[[1]]$Total.Interactions)
mean(paid_unpaid[[2]]$Total.Interactions)

m1 <- lm( log(Total.Interactions + 1)  ~   Category  +
                  Page.total.likes  + Post.Month + Post.Weekday + Post.Hour + Paid,, data=
    train_balance)
mse(m1)
summary(m1)

preds_m1 <- exp(predict(m1, test_balance)) - 1
mean((test_balance$Total.Interactions - preds_m1)^2)^.5

m2 <-  glm(Total.Interactions  ~   Category  +
                  Page.total.likes  + Post.Month + Post.Weekday + Post.Hour + Paid, family="
    poisson", data=train_balance)
mse(m2)
summary(m2)

confint(m2)

mean(m2$residuals)
var(m2$residuals)

preds = exp(predict.glm(m2, test_balance))

mean((test_balance$Total.Interactions -  preds) ^ 2)^.5

# Good RMSE, but notice
mean(test_balance$Total.Interactions )
var(test_balance$Total.Interactions )
# Can't possibley be poisosn

m3 <-  glm.nb(Total.Interactions  ~ Page.total.likes  + (Post.Month + Post.Weekday * Post.
    Hour)
              + Paid+ Category, data=train_balance)
mse(m3)
```

```
summary(m3)

layout(matrix(c(1,2,3,4),2,2))
plot(m3)

# Promising
m3_preds <- exp(predict(m3, test_balance))
mean(( test_balance$Total.Interactions - m3_preds )^2)^.5

lm_link <- glm.nb(Total.Interactions  ~
                  Page.total.likes  + Post.Month + Post.Weekday + Post.Hour + Paid,data=
    split_data[[1]])
lm_photo <- glm.nb(Total.Interactions  ~   Category  +
                  Page.total.likes  + Post.Month + Post.Weekday + Post.Hour + Paid, data=
    split_data[[2]])

lm_status <- glm.nb(Total.Interactions  ~   Category  +
                  Page.total.likes  + Post.Month + Post.Weekday + Post.Hour + Paid, data=
    split_data[[3]])

mse(lm_link)
mse(lm_photo)
mse(lm_status)

rss_link <- (split_data[[1]]$Total.Interactions -  exp(predict.glm(lm_link, split_data[[1]])
    ) ) ^ 2
rss_photo <- (split_data[[2]]$Total.Interactions - exp(predict.glm(lm_photo, split_data
    [[2]]))) ^ 2
rss_status <- (split_data[[3]]$Total.Interactions - exp(predict.glm(lm_status, split_data
    [[3]]))) ^ 2

mean(rss_link)^.5
mean(rss_photo)^.5
mean(rss_status)^.5

mean(split_data[[2]]$Total.Interactions)
var(split_data[[2]]$Total.Interactions)^.5



plot(density(x = split_data[[2]]$Total.Interactions), main="Density estimate for Photo
    Training Set")

plot(density(x = split_test[[2]]$Total.Interactions), main="Density estimate for Photo
    Training Set")

length(split_test[[1]]$Total.Interactions)  + length(split_test[[2]]$Total.Interactions) +
    length(split_test[[3]]$Total.Interactions)

(sum(rss_link + rss_photo + rss_status)/ 242)^.5

features <- c('Page.total.likes', 'Category', 'Post.Month', 'Post.Weekday', 'Post.Hour', '
    Paid')
target <- c('Total.Interactions')

x <- as.matrix(train_balance[features])
y <- as.matrix(train_balance[target])

tx <- as.matrix(test_balance[features])
ty <- as.matrix(test_balance[target])

cv.ridge = cv.glmnet(x, y, alpha = 0, lambda = c(.1,1,10,50,100,10000,1000000, 10000000),
                     nfolds = 3, type.measure = 'mse')

plot(cv.ridge$glmnet.fit, xvar="lambda", label=TRUE, )

plot(cv.ridge)
cv.ridge$lambda.min
```

```
cv.ridge$lambda.1se
coef(cv.ridge, s=cv.ridge$lambda.min)

#Lambda of 5 works
fit = glmnet(x, y, alpha = 0,lambda =cv.ridge$lambda.min)
mean((y - predict(fit, x))^2)^.5

coef(fit)

print("RMSE Ridge")
mean((ty - predict(fit, tx))^2)^.5

m1.disp <- glm(TotalInteraction ~ Followers + Type+Month+Category+Day+Paid+Hour, family="
    quasipoisson", data =d)
summary(m1.disp)
summary(m1.disp)$dispersion   #264.5635


m2.disp <- glm(TotalInteraction ~ Followers+Type+Category+Day+Paid+Hour, family="
    quasipoisson", data =d)
summary(m2.disp)
summary(m2.disp)$dispersion   #276.1557

m3.disp <-glm(TotalInteraction ~Type+Month+Category+Day+Paid+Hour, family="quasipoisson",
    data =d)
summary(m3.disp)
summary(m3.disp)$dispersion   #264.655
```

7.2 Poisson Derivations

If Y is a discrete Poisson random variable, then Y has range $Range(Y) = \{0, 1, 2, ..., \infty\}$. (This makes sense, since it approximates the Binomial distribution, which has range $\{0, 1, 2, ..., n\}$ when $n \to \infty$.) It's PDF is

$$\Pr[Y = k] = \frac{e^{-\lambda}\lambda^k}{k!}$$

Proposition 7.1.

$$\lim_{n\to\infty} \binom{n}{k} \Big(\frac{\lambda}{n}\Big)^k \Big(1 - \frac{\lambda}{n}\Big)^{n-k} = \frac{e^{-\lambda}\lambda^k}{k!}$$

In particular, $X \sim Pois(\lambda)$ where the PMF is

$$f_X(k) = \frac{e^{-\lambda}\lambda^k}{k!}$$

Proof. By definition,

$$\binom{n}{k} = \frac{n(n-1)\dots(n-x+1)}{k!}$$

Then

$$\begin{aligned}\lim_{n\to\infty} \binom{n}{k} (\frac{\lambda}{n})^k (1 - \frac{\lambda}{n}^{n-k} &= \lim_{n\to\infty} \frac{n(n-1)\dots(n-x+1)}{k!} (\frac{\lambda}{n})^k (1 - \frac{\lambda}{n}^{n-k} \\ &= \lim_{n\to\infty} \frac{n}{n} \cdot \frac{n-1}{n} \dots \frac{n-k+1}{n} \frac{\lambda^x}{k!} \Big(1 - \frac{\lambda}{n}\Big)^n \Big(1 - \frac{\lambda}{n}\Big)^{-k}\end{aligned}$$

Now use the limit formula for e^x, and notice that $\lim_{n\to\infty} \left(1 - \frac{\lambda}{n}\right)^{-k} = 1$. This gives us the result

$$\lim_{n\to\infty} \frac{n}{n} \dots \frac{n-1}{n} \dots \frac{n-k+1}{n} \frac{\lambda^x}{k!} \Big(1 - \frac{\lambda}{n}\Big)^n \Big(1 - \frac{\lambda}{n}\Big)^{-k} = \frac{e^{-\lambda}\lambda^x}{x!}$$

■

Proposition 7.2. *For $X \sim Pos(\lambda)$, $E(X) = \lambda$.*

$$\sum_{k=0}^{\infty} \frac{e^{-\lambda}\lambda^k}{k!} k = \lambda.$$

Proof.

$$\begin{aligned}\sum_{k=0}^{\infty} \frac{e^{-\lambda}\lambda^k}{k!} k &= \sum_{n=0}^{\infty} \frac{e^{-\lambda}\lambda^k}{(k-1)!} \\ &= \lambda e^{-\lambda} \sum_{n=0}^{\infty} \frac{\lambda^{k-1}}{(k-1)!} = e^{-\lambda}\lambda e^{\lambda} = \lambda\end{aligned}$$

■

Proposition 7.3. *For $X \sim Poiss(\lambda)$, $Var(X) = \lambda$*

Proof. Let our random variable X have Poisson distribution with parameter λ. We start from the familiar Maclaurin series

$$e^{\lambda} = 1 + \frac{\lambda}{1!} + \frac{\lambda^2}{2!} + \frac{\lambda^3}{3!} + \frac{\lambda^4}{4!} + \frac{\lambda^5}{5!} + \cdots.$$

Differentiate twice, and scale by $\lambda^2 e^{-\lambda}$. This gives

$$\lambda^2 = (2)(1)e^{-\lambda}\frac{\lambda^2}{2!} + (3)(2)e^{-\lambda}\frac{\lambda^3}{3!} + (4)(3)e^{-\lambda}\frac{\lambda^4}{4!} + (5)(4)e^{-\lambda}\frac{\lambda^5}{5!} + \cdots.$$

We recognize the right-hand side as $E(X(X-1))$. So $E(X^2) = \lambda^2 + \lambda$. Hence,

$$\mathrm{Var}(X) = E(X^2) - (E(X))^2 = \lambda^2 + \lambda - \lambda^2 = \lambda$$

■

References

[1] Moro et al., 2016) Moro, S., Rita, P., & Vala, B. (2016). Predicting social media performance metrics and evaluation of the impact on brand building: A data mining approach. Journal of Business Research, 69(9), 3341-3351