Facebook Comments Prediction

by

Abhishek Patria (903511102)
Jackson Michalski (903528016)
Reynaldo Peña (903539197)

H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology
December 2019

Table of Contents

Summary

Our goal in this project is to predict the number of comments a Facebook would get within a certain number of hours after posting. We tried linear and linear models including Lasso Poisson regression, forward stepwise regression, PCA poisson regression and random forests. PCA Poisson Regression outperformed all models including the random forests models in terms of our key metric RMSE. This is a bit surprising given that neither the linearity nor normality assumptions hold and all of our goodness of fit tests suggest a bad model fit for our linear models. Further feature engineering such as log transformations and more advanced model techniques such as quasi-poisson are probably necessary in order to get more accurate predictions. Please note for the visual analytics we purposefully only included very few of the plots due to the high number of features. The information in these plots about the model assumptions can be gleaned from just looking at the plots provided. We did generate all of the plots, which can be seen in our code.

Motivation

The motivation for this project stems from the desire to predict the number of facebook comments on a post within a certain amount of time after posting. Knowing how to predict comments on a post can be seen as a proxy for user engagement which would be interesting to people running ads on Facebook. For example, a company choosing among several potential Facebook posts to promote their product may be interested in knowing which post is most likely to get user engagement in terms of Facebook comments. We also wanted to compare our findings to that of a paper we found written about the same dataset using machine learning techniques. By comparing these techniques to traditional regression techniques, we can gauge if the extra computational complexity used in the online research papers yield significantly better results than our simpler linear methods.

Data Description

Variables

1. Page Popularity/likes: Defines the popularity or support for the source of the document.
2. Page Check-ins: Describes how many individuals so far visited this place. This feature is only associated with the places eg:some institution, place, theater etc.
3. Page talking about: Defines the daily interest of individuals towards source of the document/ Post. The people who actually come back to the page, after liking the page. This includes activities such as comments, likes to a post, shares, etc by visitors to the page.
4. Page Category: Defines the category of the source of the document eg: place, institution, brand etc.
5. Features 5 - 29: Derived. These features are aggregated by page, by calculating min, max, average, median and standard deviation of essential features.
6. CC1: The total number of comments before selected base date/time.
7. CC2: The number of comments in the last 24 hours, relative to base date/time.
8. CC3: The number of comments in last 48 to last 24 hours relative to base date/time.
9. CC4: The number of comments in the first 24 hours after the publication of post but before base date/time.
10. CC5: The difference between CC2 and CC3.
11. Base time: Selected time in order to simulate the scenario.
12. Post length: Character count in the post.
13. Post Share Count: This features counts the no of shares of the post, that how many people had shared this post on their timeline.
14. Post Promotion Status: To reach more people with posts in News Feed, individual promote their post and this features tells that whether the post is promoted(1) or not(0).
15. H Local: This describes the H hrs, for which we have the target variable/ comments received.
16. Post published weekday: This represents the day(Sunday...Saturday) on which the post was published.
17. Base DateTime weekday: This represents the day(Sunday...Saturday) on selected base Date/Time.
18. Target Variable: The no of comments in next H hrs(H is given in Feature no 39).

Format

We had a training data with 49,049 rows and a testing data with 10,043 rows . After feature engineering and cummy encoding, we ended up with 41 columns. Some columns were very ambiguous, especially the ones named “Derived”, since these are columns composed of unspecified metrics within a Facebook page. We have no clue what these metrics actually are, however, so it would be hard to interpret them in our model.

Distribution

We also looked at the distribution of our response and some of our predictors. One of our columns corresponds to the time period (in hours) after a Facebook post was posted in which we count the number of comments on the post. In this column, 98% of values are 24 (hours). Similarly, we see that we have a lot of posts in our dataset (55%) who end up receiving no comments at all. This is important to keep in mind as we may have to correct for such overrepresentation of the 0 value in the response.

Correlation

We next looked at the correlation between our predictors. We notice the following: Page_Talking_About is highly correlated (0.62) with Page_Popularity, and many of the derived variables are extremely correlated with each other (some having correlations of 0.99). This suggests that many of the unknown "Derived" variables are actually measuring extremely similar metrics between them. This means that variable selection will be essential before trying out any linear regression model. Interestingly, our target variable does not have any strong correlation with any of the other predictors, with all correlations magnitudes being less than 0.4.

Data Cleaning

In order to clean our data, we first labeled all of our columns with the column names in the pdf so we could distinguish our columns as our data had no header. There were over 100 categories in the 'Page Category' column, so we aggregated them based on common themes (we included Business, Entertainment, and Miscellaneous). This gives us 3 factor columns with some values not belonging to these three categories so we don't have multicollinearity. We also can omit the na values safely as we only had one row that had NA values. We had one category variable for day of the week in which the Facebook post was posted. In order to reduce the number of variables after dummy encoding, we changed this column into a binary factor denoting whether or not the day was a weekend or weekday. Next, we scaled our predictors for PCA and Lasso regression.

Analysis

We tried a multitude of different models on the dataset including both regression models and forest based models. Most of our efforts went into variable selection for our models in order to find the best combination of features. We first tried running multivariate poisson regression model using all of our predictors before we did our data manipulation. This raised singularity warnings due to high multicollinearity, which means that our model wouldn't be reliable. Since this approach wasn't feasible, we knew we would need to focus on variable selection methods detailed below. Finally, notice that we used offset in all our models given that we are interested in predicting a standardized Facebook comment count. Since the response variable was measured on different time periods, we needed to use offset to account for this.

LASSO Regression

We first tried a lasso model on scaled data in order to remove predictors which might be collinear and reduce our model size. We ran the lasso poisson regression using the number of hours after the Facebook post as offset and saw that we reduce the number of features significantly. Doing outlier analysis using Cook's distance we see that we have a substantial amount of outliers (arbout 16% of the data set), with one point standing out as having an exceptionally large Cook's distance. We removed all of the points with Cook's distance greater than 10 before rerunning the final model.

Model

log(Number of Comments/hour) = -6.684129e-01 + Page_Talking_About* -8.675371e-07 + Page_CategoryBusiness* -4.292467e-01+ Page_CategoryEntertainment * 3.718186e-01 +Derived_1* 9.176004e-04 +Derived_2* 5.029431e-04 + Derived_5 *2.729482e-03 + Derived_10 * 1.147006e-03 + Derived_12 * -4.123703e-05 + Derived_14* -1.871495e-03 +Derived_21* -6.080661e-04 + Derived_22* 5.090138e-05 + Derived_25 *-1.564784e-04 +CC1* 1.744149e-03 +CC2* 8.565070e-05 + Base_Time* -8.211152e-02 + Post_Share_Count * 6.878028e-05

Next, you can see our plot of Cook's distances and how the coefficients' change according to each lambda. Notice how a bunch of the predictors go to 0 fairly quickly which makes sense given that we have issues with multicollinearity so many redundant features go to zero.

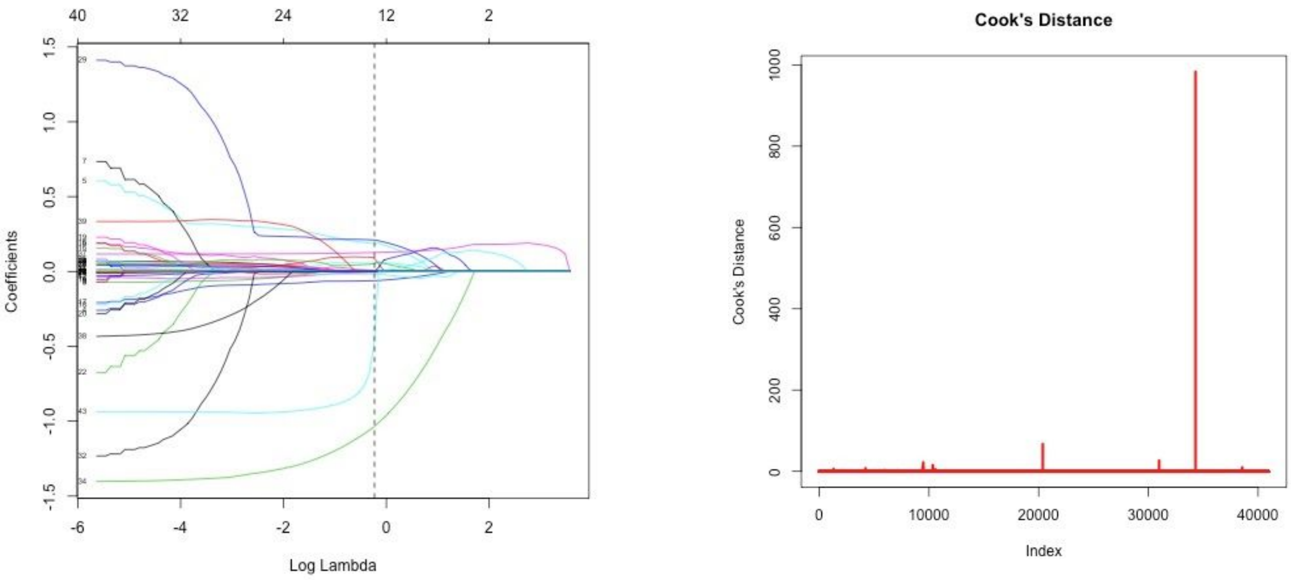

Figure 1: Regression coefficient path - Lasso Regression (left), Cook's Distance Plot for outlier ID (right)

Elastic Net Regression

We also attempted to run an Elastic Net model on the data, but unfortunately were not able to get results. The Elastic Net either crashed R or kept running indefinitely so it was not possible to get it to finish. We decided our time was better spent elsewhere and moved on from the model.

Variable selection using Forward Stepwise Method

To also combat multicollinearity, we try using forward stepwise model selection to choose the best features. We chose forward over backward selection because we have over 40 predictors so it is computationally much more efficient to do forward selection. The final model ends up having 23 features so we end up cutting about half of the features. We also notice that we have a relatively small number of outliers compared to the other models using Cook's distances. We see that only around 2% of our points are outliers so it looks like our model fits the data a lot better. We will stick with only removing data points that have Cook's distances larger than 10 because we still see some big spikes.

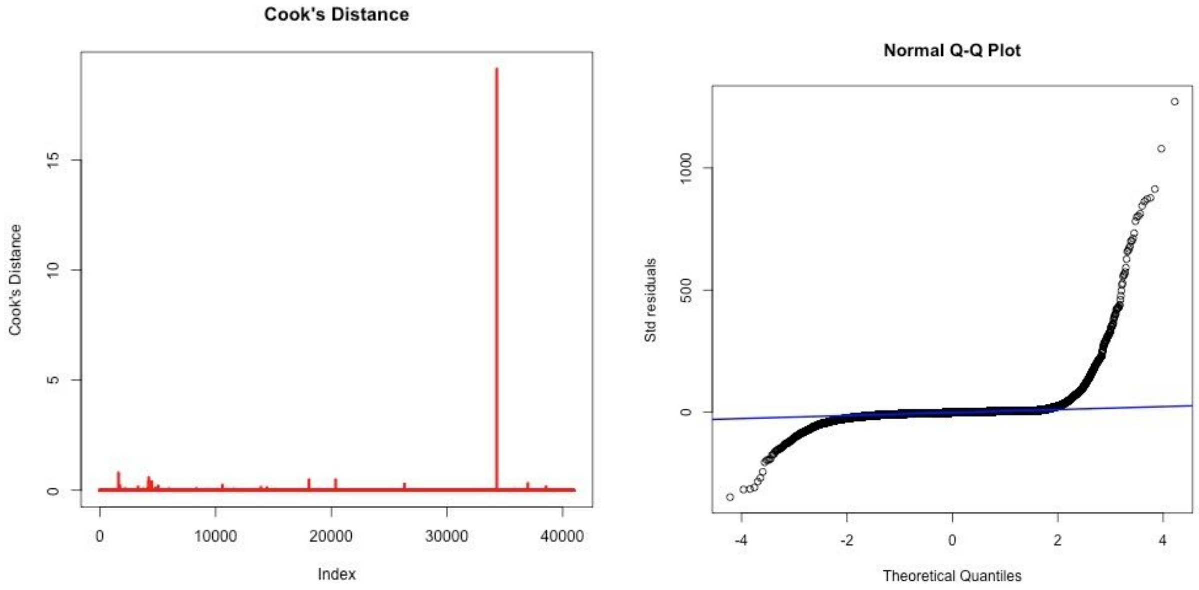

Figure 2: Cook’s Distance Plot for outlier ID for Forward Stepwise (left), QQ-Plot for Forward Stepwise (right)

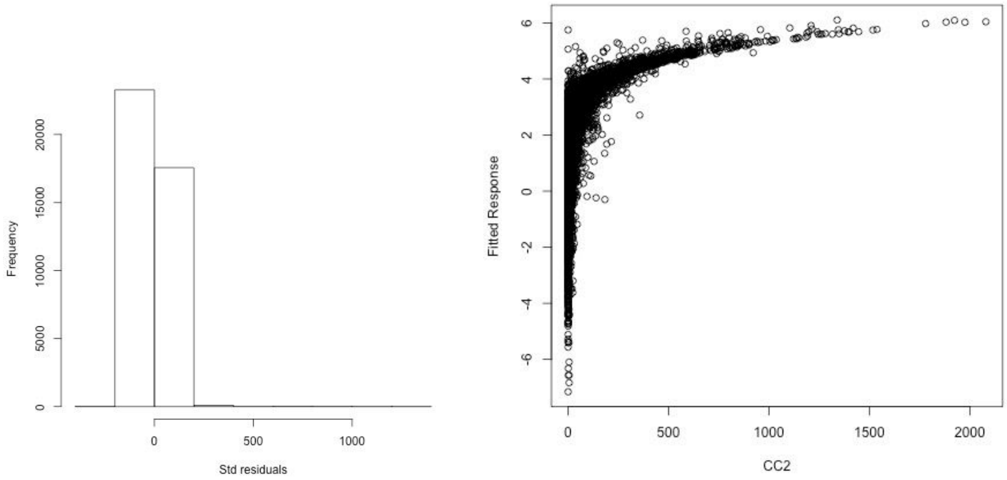

Figure 3: Residuals Histogram for Forward Stepwise (left), Response vs CC2 for Forward Stepwise (right)

We see a strong right skewed deviance distribution and clusters of data points in the response vs predictor plot, which makes sense given that we have such an overwhelming number of response data with 0 values and a few very large response values. Note that we tried log transforming our response data, but given that 55% of the response data is 0, it makes the usual log transformation useless.

Below, we see that variable selection differs substantially from Lasso regression. For example, the variable on page category is not selected here, while in Lasso this category had the largest coefficient value.

Model:

log(Number of Comments / hour) = 6.9065966913+ CC2 *0.2197822794+ Base_Time * -0.1939119262+ Derived_3 *0.8265577892 + CC4 *-0.0661573193+ Derived_18 * -0.8780674798+ Post_Share_Count *0.0029475318+Derived_8 * 0.1567580094+ Page_Talking_About * -0.0000188974+ Derived_1 *-0.4711498901 + CC3 * -0.0232379525+ Derived_9* 0.1244719358+ Derived_12 * 0.0115746670+ Derived_14 * -0.0079203331+ Derived_6 *-0.1185672978+ Derived_16 *0.4233607563 + CC1*0.042452006 + Derived_19* -0.2113272028 +Derived_15*-0.1357136162+Derived_4*0.1843953553+ Derived_7* -0.0031193675 + Derived_13* 0.1286187261+ Derived_20 * 0.0250157705+ Page_Checkins *-0.0000124908

PCA Regression

We also tried PCA Poisson Regression as a means to reduce both multicollinearity and dimensionality. First, we ran PCA and then graphed the percentage of the variability explained vs number of components so we could choose which k to use. Normally we would use AIC to choose models, but as we add more components we will always get a decreasing AIC value so we need to choose the value qualitatively instead. As we increase the number of components, our AIC goes down, but our overdispersion value increases so we want to balance the tradeoff. In the end, we picked 15 as it is in the range of decreasing marginal variability explained by components so it seems like a good compromise.

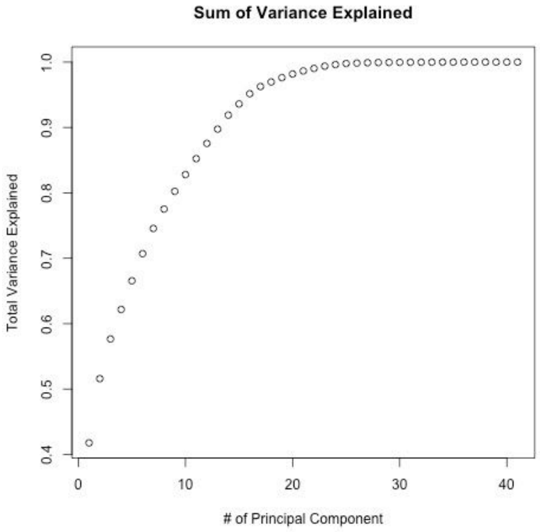

Figure 4: Plot explaining the cumulative variance explained by each Principal Component

Looking for potential outliers, we see that 18% of our data is considered an outlier by Cook's distance. This suggests that we have a large proportion of potential outliers and thus we can not simply remove them. We do have a single point that has a much larger Cook's distance than any of the others so we will remove it to see how our model accuracy changes.

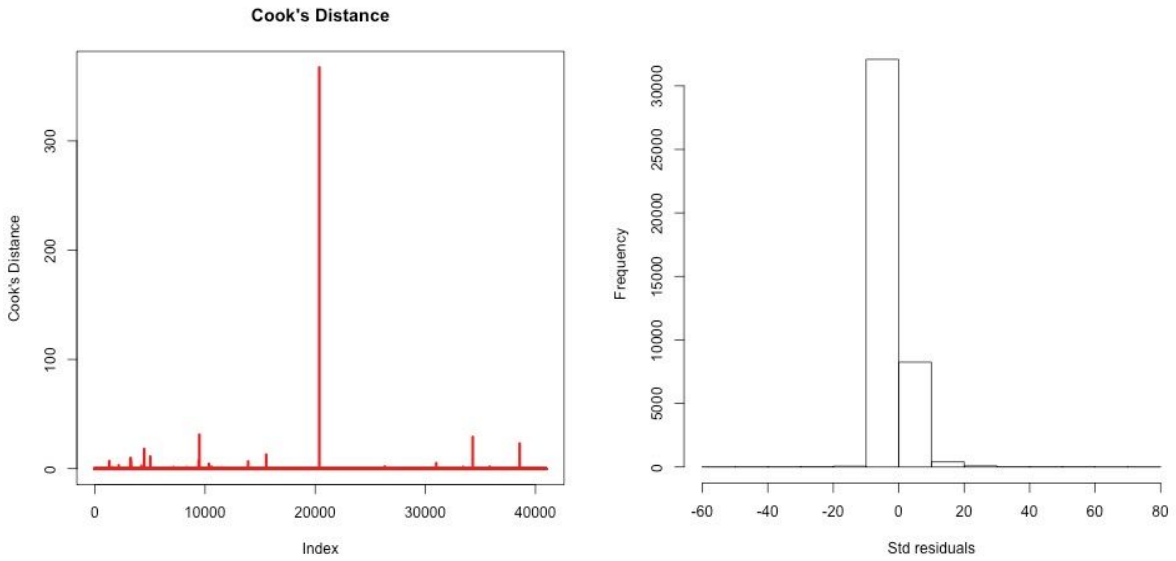

Figure 5: Cook's Plot Outlier ID using PCA Regression Model (left), Residuals Histogram for PCA Regression (right)

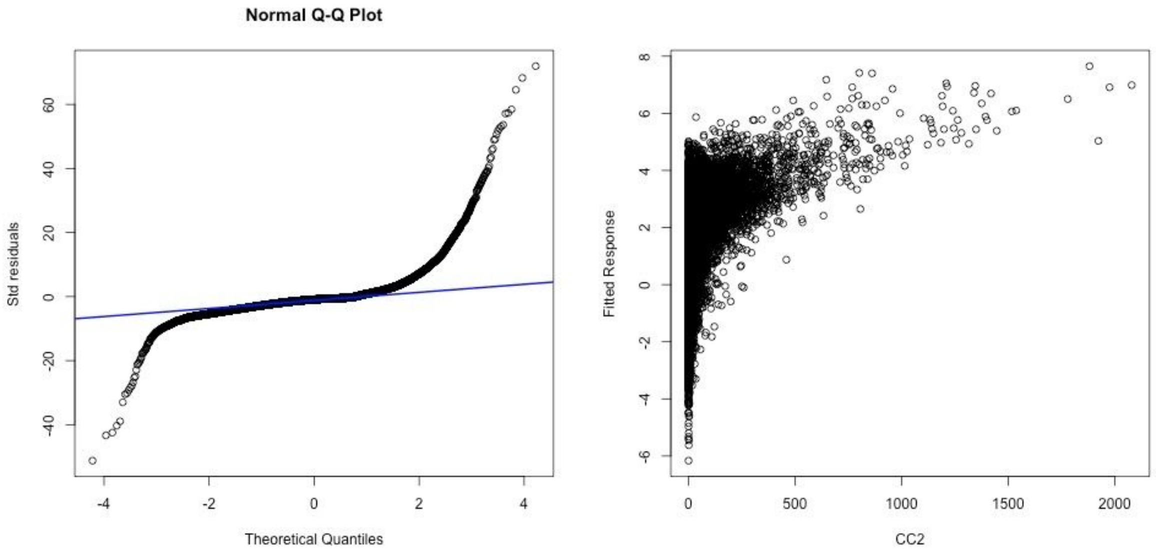

Figure 6: QQ Plot for PCA Regression (left), Response vs CC2 for PCA Regression (right)

As seen above, deviance residuals are also right skewed and clustered around 0. It seems that it is something that will negatively impact our predictions in all of our linear models. Once we convert back our principal components to original variables, we get the following equation:

Model:

log(Number of Comments / hour) = -2.701770421+Page_Popularity * 0.006050946+Page_Checkins
*0.014926932+Page_Talking_About * 0.039655689 +Page_CategoryBusiness *-0.239677078
Page_CategoryEntertainment 0.177537137+Page_CategoryMiscellaneous *-0.009880113+Derived_1 *
-0.003115674+Derived_2 *0.062791588+Derived_3 * 0.011455646+Derived_4 *
-0.010619001+Derived_5 *0.048502672+Derived_6 * -0.008302338+Derived_7 *
0.044281107+Derived_8 *0.010668980+Derived_9 * 0.003776550+Derived_10 *
0.026398887+Derived_11 * 0.048858489+Derived_12 * 0.068660947+Derived_13 *
0.025016120+Derived_14 *-0.032369176+ Derived_15 * 0.054614608+Derived_16
*-0.004938402+Derived_17 *0.060209364+Derived_18 * 0.012127710+Derived_19
*-0.010392198+Derived_20 *0.048536723+Derived_21 * -0.060386741+Derived_22*
0.039036613+Derived_23*-0.021877663+Derived_24 * 0.024012850+Derived_25*
0.031698991+CC1*-0.035094168+CC2 *0.167980099+CC3 * 0.158409043+CC4 * -0.019948773+CC5
*0.018002107+Base_Time * -1.598274789+Post_Length * 0.043017332+Post_Share_Count
*0.008855442+Post_Day_Weekend1 * -0.001418564+Base_Day_Weekend1* -0.007019663

Random Forest

In addition to fitting the above linear models, we also tried a random forest model on our data.

Initially, we built a default model to decide the maximum number of trees that our model should have because having too many trees substantially increases the computational time for fitting.

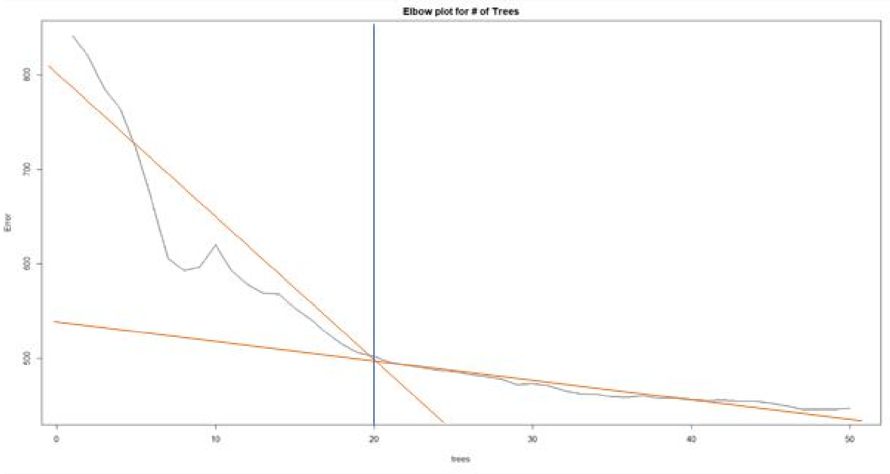

Figure 7: Elbow Plot for Error vs Number of Trees

We see that after 20 trees there is hardly any benefit to increasing the tree number, so we set the maximum number of trees to be 20. Then, we tuned our model by tweaking the ‘mtry’ parameter: the number of variables to be randomly sampled while building a decision tree.

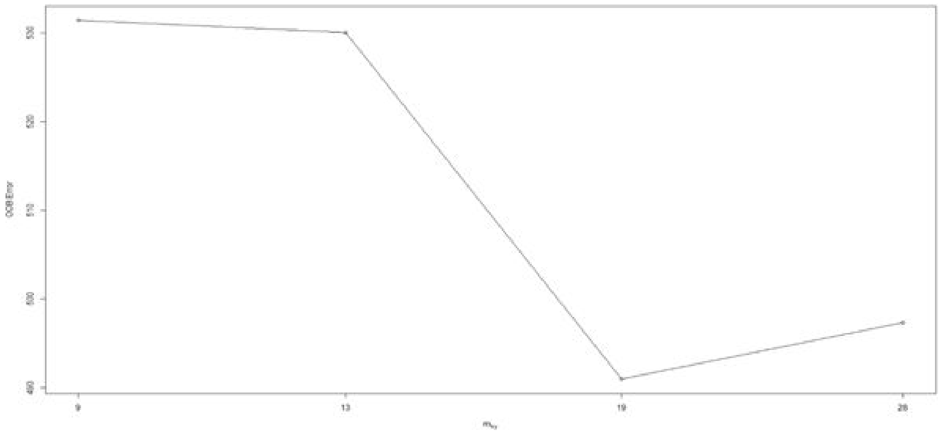

Figure 8: Plot for Out of Bags Error vs Number of Randomly Sampled Variables

From the plot above, we can see that mtry parameter around 19 gives us the least out of bags error. So, we kept the mtry parameter to be 19.

The tuned CART based model had a completely different variable importance as compared to the linear models. We got the following plot using the node purity based on the information-gain in each Occam’s Razor. It is evident that Bast_time, CC5, CC2, and Post_Share_Count are by far the most important predictors in our random forest model.

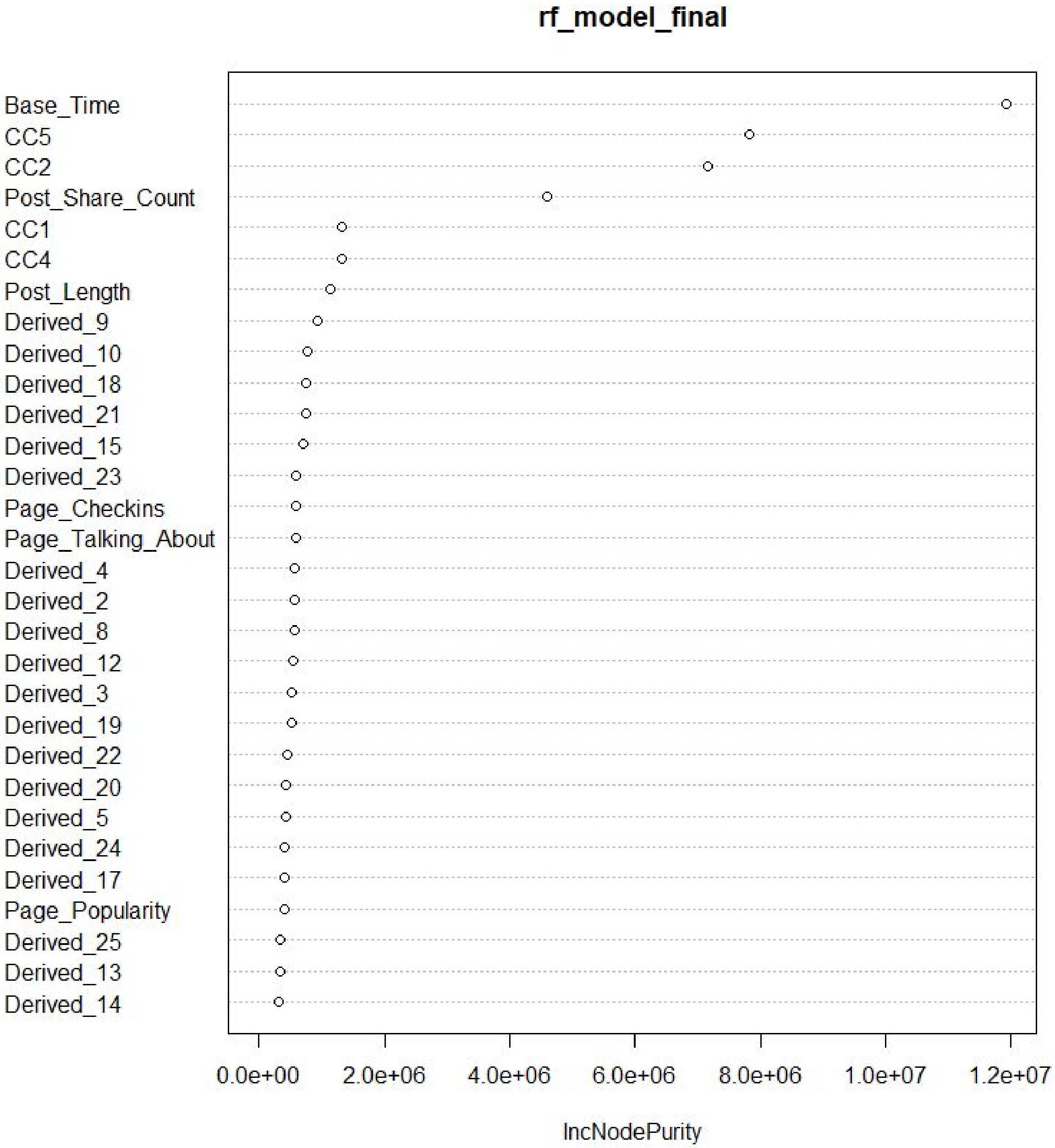

Figure 9: Variable Importance Plot using Occam's Razor method

The following plot suggests that our model has performed fairly well on the training data by minimizing the OOB error with a calculated RMSE of 0.044 which is extremely good.

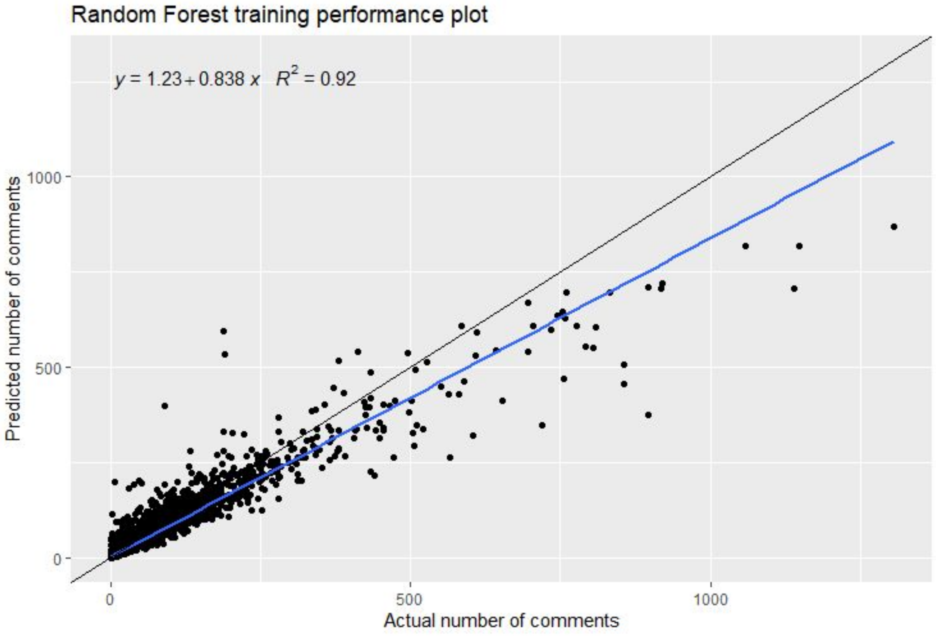

Figure 10: Performance Plot for Random Forests on Training Dataset

But the same hyperparameter tuned model does not perform well on the test dataset as suggested by the following plot. The RMSE on the test dataset is 15.774

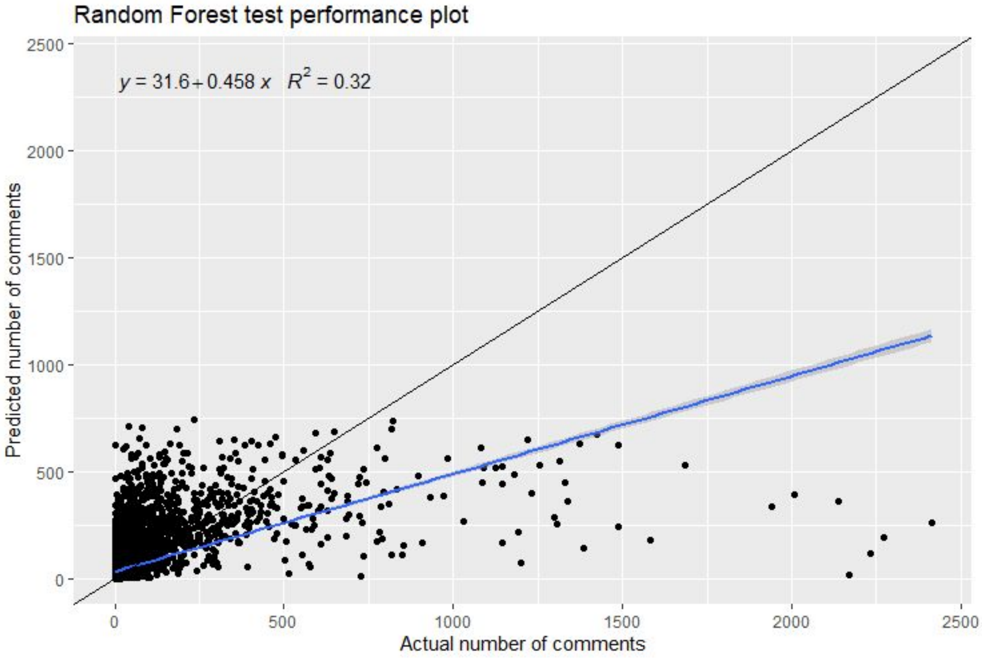

Figure 11: Performance Plot for Random Forests on Test Dataset

Random Forests model is a model based on the aggregation of trees trained through bagging and variable randomization, so we simply cannot comment about the interpretability of our model. For individual trees we can comment about the parameters used for making that decision tree and interpret our model but as we are using Random Forest one particular tree's decision criteria may contradict another one. As a result, the best we can do is estimate the variable importance from the table above.

Moreover, our residual plot shows the following trend of residuals. As mentioned above, it is challenging to transform our response data given the large presence of 0 values. Moreover, random forests are immune to transformations, so transforming the response would likely not improve the predictions.

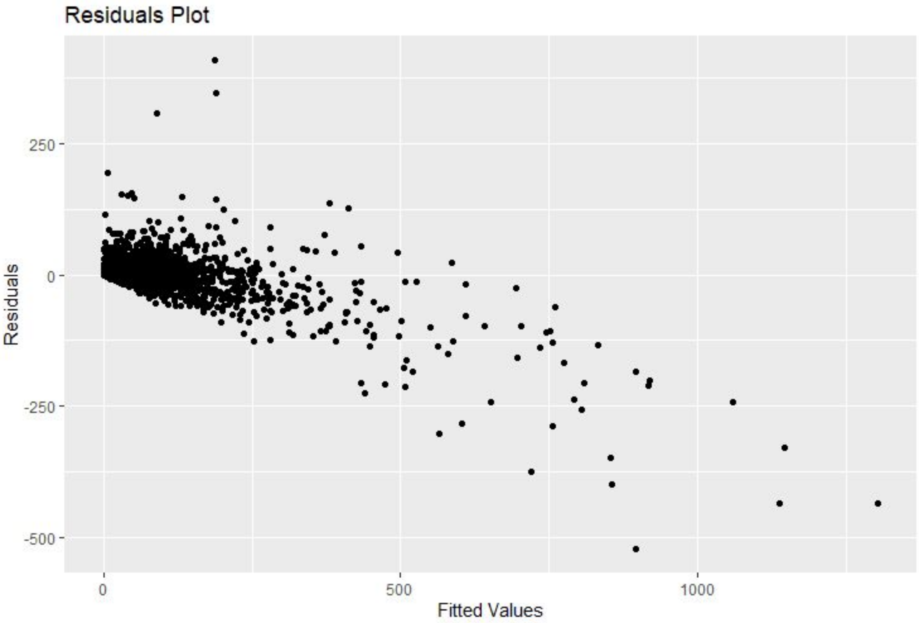

Figure 12: Residuals vs Fitted Values for Random Forests

Comparison

Now that we have defined all of our models, let's look at the results we get and decide on a single model to use as a final model. Below is a table that summarizes our results. Evidently, PCA Regression is the worst performer in terms of AIC, BIC, and Mallow's CP. Forward stepwise regression has a much lower AIC, BIC, and Mallow's CP value compared to Lasso and PCA Regression, which suggests a better model fit. However, we were mostly focused on the predictive accuracy of the model, and thus RMSE was our main focus when choosing a model. By this measure, PCA Regression is a clear winner with the lowest RMSE of 3.6206451. This means that on average, the root mean squared prediction between our model's prediction of the comment rate per hour on a Facebook post and the actual comment rate is 3.620645. This is still a little high given that the average test set comment rate is 2.04. Unsurprisingly, the forward stepwise regression had the highest RMSE value as it starts with the least amount of information and builds up. The random forest was the second clear winner, which suggests that a non linear model also does very well at predicting our poisson response data. The deviance test yields a p-value of 0 for all our models, indicating that our models are a poor fit to the data. Taking this into account, it is a bit surprising that our PCA model still outperformed substantially our random forest model.

Model	Overdispersion	AIC	BIC	Mallow's CP	Deviance Test p-value	RMSE
LASSO	146.7699	516026.1	516172.7	501065.5	0	39.99102
Forward Stepwise	851.7688	392530	392745.5	24	0	55.9071
PCA Regression	36.23505	549103.7	549241.7	83844.51	0	3.620645
Random Forest	NA	NA	NA	NA	NA	15.774

Figure 13: Comparison matrix for the models tested

Implications

Of all our models tested, we found that a regression model actually can work better than the more computationally complex Random Forest model. This shows how powerful a poisson regression regression can be for the right problem. That being said, the data is still clearly not being well represented by the models, but at least one model (PCA regression) is still able to have high predictive accuracy. Compared to the paper where we got this data set from, it seems that these results aren't unexpected as their results are not very good either. This implies that either the task is not very well suited to prediction, that the wrong models are still being used, or that the dataset itself is not very good.

Further Questions

One of the biggest drawbacks of our models was that they were all highly overdispersed, with PCA regression having the lowest overdispersion of 36.23 (when 2 is the usual acceptable limit). One way to deal with overdispersion is to use a quassi-poisson model, where the assumption of mean/variance equality is relaxed to allow the model to estimate the variance from the data. Predictions and estimates remain the same as in a normal poisson model, but inferences are adjusted to the new variance estimate. Given that we only focus on the predictive accuracy of our models and hypothesis testing, we found that running a quasi poisson model was unnecessary.

Additionally, it would be beneficial to explore other possible transformation of the response. It was previously indicated that the usual log transform was not likely to work because of the high number of 0 values. However, we suggest trying other Box-Cox transformations in the future. We believe this to be necessary for many reasons. First, for most of our models we saw an abnormally high amount of outliers according to cook's distances which seems to suggest that the distribution has a sizeable tail. Secondly, all of the QQ plots suggest a strong right skew so it is clear the normality assumption does not hold for any of our models. These also show very drastic tails so we should look into transformations that would account for this. Next, our plots of fitted response vs the predictor suggest non linear relationships between the response and the predictors. This suggests that most of our model assumptions are violated, so we should not expect the model to be a good fit to the data. Finally, all of our Goodness of Fit tests had values of 0 so we have a really bad fit to the data as we want a p-value close to 1. However, this does not mean these models are useless in terms of predictive power, as was evident with the relatively low RMSE of our PCA Poisson model.