Predicting Buzz in Social Media

Md. Anees Parwez (BS 1704)

Bhaskar Ray (BS 1717)

Souhardya Sengupta (BS 1726)

Indian Statistical Institute, Kolkata

Abstract

In this article, we develop methods to assess the popularity of a topic on Twitter, based on a relevant dataset. We present methods for assessing the significance of various attributes that might affect popularity and develop powerful prediction models for the same. The article attempts a thorough discussion of the dataset and the challenges involved that might hinder conventional analysis, the prominent of them being multicollinearity and high-dimensionality. We identify two groups of variables depending upon their correlation pattern with the response, showing distinctive within-group behavior. Finally, based on our observations, we propose a semi-parametric model for prediction and assess it accuracy against the classical linear model.

Keywords: Twitter, Buzz Prediction, Multicollinearity, Variable Selection, Semi Parametric Modelling

1 Introduction

With the advent of social media, the phenomenon of 'overnight fame' is more frequent than ever before. It is often seen that certain events/topics gain sudden popularity. These are said to create a *'buzz in social media'*, and is characterized by a large number of discussions over a short period of time. A challenge is to be able to predict whether a topic will become a *buzz* in coming days based on some metric measured during the initial days the topic starts attracting discussions. Our discussion pertains to the dataset available in [2]. Our analysis faces two major challenges: High-dimensionality and Multicollinearity. Also, we will see that traditional ANCOVA based methods won't be helpful in establishing significance. This article summarizes our method of variable selection and of prediction, which we achieve using a semi-parametric approach.

This article is organized as follows: In the next section, we present a description of the dataset along with visualisation of various attributes. In Section 3, we discuss methods to assess the significance of these variables and classify them into two groups with distinctive behavior. In Section 4, we try to develop prediction models and propose our semi-parametric approach. We conclude the report with comparison of different models and discussions in Section 5. The R-codes used in our analysis are available in Appendix A.

2 The Buzz in Social Media (Twitter) Dataset

Our analysis pertains to the 'Buzz in social media' dataset, collected from the UCI Machine learning Repository [1]. It contains various characteristics collected for a topic(keyword), forming a single instance. We have **583250** instances. The task is to predict a 'buzz score' (measuring popularity) for each topic.

In order to understand the characteristic for each keyword, [2] develops a mathematical framework of social networking sites. However, for our purpose, we only present a summary of the features in Table 1. For each keyword z, each of the following 11 characteristics were collected at 7 time points. Thus for each row, there are $7 \times 11 = 77$ predictors. Note that the data matrix has 78 columns. The last column forming the response. To reiterate,for each row (ie. for each keyword) 11 different characteristics are noted at 7 time points. Different rows corresponds to different keyword and are independent.

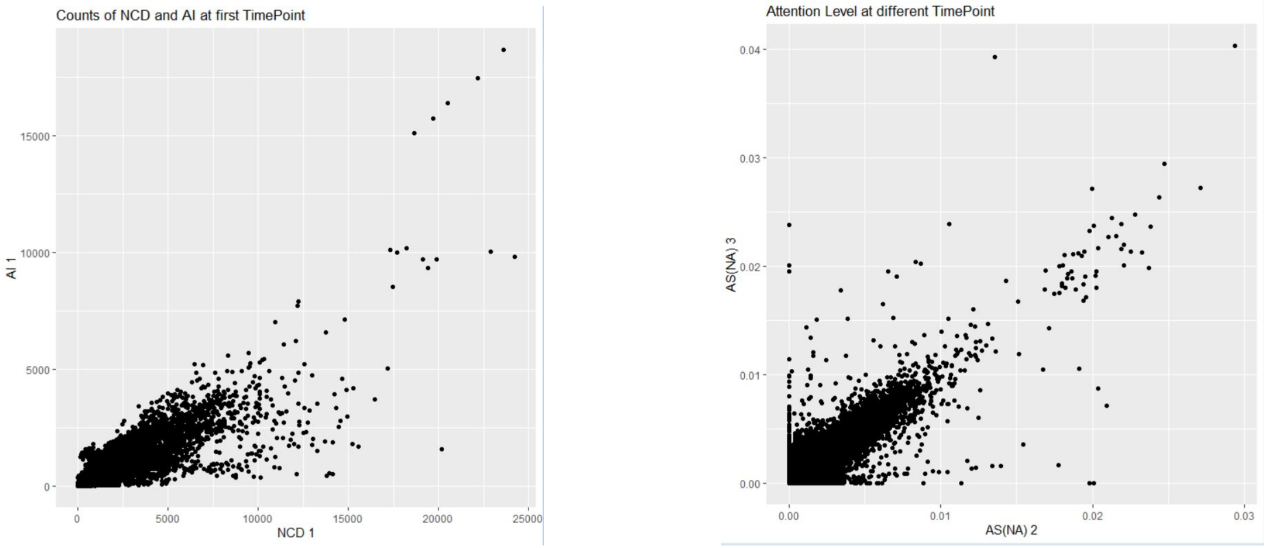
Name	Type	Description
NCD (t,z)	Count	The number of threads created at time step t which involve keyword z
AI(t,z)	Count	Number of new authors interacting on the keyword z at time t
AS(NA) (t,z)	Proportion	Attention level (measured with number of authors) of keyword z at time t
BL(t,z)	Proportion	Ratio of NCD(t,z) and NAD (t,z)
NAC (t,z)	Count	Number of tweets generated until time t for keyword z
AS(NAC) (t,z)	Proportion	Attention level (measured with number of tweets) of keyword z at time t
CS(t,z)	Proportion	Spreading of contributions over thread for the keyword z at time t
AT(t,z)	Count	Average number of authors interacting on the keyword z within a tread.
NA(t,z)	Count	Number of authors interacting on the keyword z at time t
ADL(t,z)	Count	Average length of a thread belonging to the keyword z at time t.
NAD(t,z)	Count	Number of active threads involving the keyword z until time t

Table 1: Summary of Dataset Attributes

2.1 Visualization

As each attribute is collected across multiple time points, there might be a possibility of multicollinearity. We explore this by correlation matrices and scatter plot of two randomly chosen attributes at a time point and two randomly chosen time points of an attribute. The scatterplots are presented in Figure 1:

(a) `NCD: timepoint-1 v/s AI: timepoint-1`
(b) `AS(NA): timepoint-2 v/s AS(NA): timepoint-3`

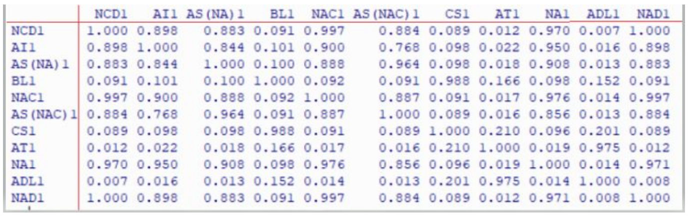


(a) Same timepoint

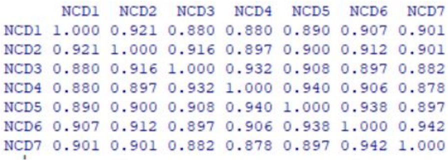
(b) Same characteristic

Figure 1: Possibility of Multicollinearity.

As the Figure 1 suggests, there seems to be a strong association between both (a) two characteristic at one time point, (b) same characteristic at different time point. A further support of multicollinearity is observed in the correlation matrix. Certain off diagonal entries are of order 0.8-1 suggesting strong dependence in variables.



	NCD1	AI1	AS(NA)1	BL1	NAC1	AS(NAC)1	CS1	AT1	NA1	ADL1	NAD1
NCD1	1.000	0.898	0.883	0.091	0.997	0.884	0.089	0.012	0.970	0.007	1.000
AI1	0.898	1.000	0.844	0.101	0.900	0.768	0.098	0.022	0.950	0.016	0.898
AS(NA)1	0.883	0.844	1.000	0.100	0.888	0.964	0.098	0.018	0.908	0.013	0.883
BL1	0.091	0.101	0.100	1.000	0.092	0.091	0.988	0.166	0.098	0.152	0.091
NAC1	0.997	0.900	0.888	0.092	1.000	0.887	0.091	0.017	0.976	0.014	0.997
AS(NAC)1	0.884	0.768	0.964	0.091	0.887	1.000	0.089	0.016	0.856	0.013	0.884
CS1	0.089	0.098	0.098	0.988	0.091	0.089	1.000	0.210	0.096	0.201	0.089
AT1	0.012	0.022	0.018	0.166	0.017	0.016	0.210	1.000	0.019	0.975	0.012
NA1	0.970	0.950	0.908	0.098	0.976	0.856	0.096	0.019	1.000	0.014	0.971
ADL1	0.007	0.016	0.013	0.152	0.014	0.013	0.201	0.975	0.014	1.000	0.008
NAD1	1.000	0.898	0.883	0.091	0.997	0.884	0.089	0.012	0.971	0.008	1.000



	NCD1	NCD2	NCD3	NCD4	NCD5	NCD6	NCD7
NCD1	1.000	0.921	0.880	0.880	0.890	0.907	0.901
NCD2	0.921	1.000	0.916	0.897	0.900	0.912	0.901
NCD3	0.880	0.916	1.000	0.932	0.908	0.897	0.882
NCD4	0.880	0.897	0.932	1.000	0.940	0.906	0.878
NCD5	0.890	0.900	0.908	0.940	1.000	0.938	0.897
NCD6	0.907	0.912	0.897	0.906	0.938	1.000	0.942
NCD7	0.901	0.901	0.882	0.878	0.897	0.942	1.000

We observe enough evidences suggesting strong multicollinearity among variables. We now turn into the question of establishing the significance of the variables, which is presented in the next section.

3 Significance of Variables

As pointed out in earlier sections, we have 11 attributes, each collected over 7 time points. These 77 instances form our covariates. Now a very obvious question can be to establish the effect of these attributes (and/or the time points). Note that we cannot resort to traditional ANCOVA-based methods for establishing the significance of the attributes, with regard to the time point effects. It is so, as we are collecting all the attributes at each of the time points and thus heterogeneity in *treatment assignment* isn't available across the observations. Each variable is a combination of an attribute and a time point and thus answering the question of significance of an attribute is tricky. In this section we try to answer this question and present our method of variable selection and establishment of significance. We will only try to establish the significance of variables and not the time points.

3.1 LASSO for variable selection

To begin with, we apply variable selection techniques on the entire dataset. Given the dimensionality of the data and presence of multicollinearity, LASSO is a natural preference for both regression and variable selection. This is because LASSO is prone to set coefficients of some of the correlated predictors to 0. Here particularly, we are interested in variables selected by LASSO. For response vector $\boldsymbol{Y}$, data matrix $\boldsymbol{X}$, we estimate the model parameters $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_{77})^T$ from the objective function

$$\arg\min_{\boldsymbol{\beta}} \left[(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \sum_{j=1}^{77} |\beta_j| \right]$$

To choose the optimal value of λ, we have used 10-fold cross-validation using MSE cost to obtain $\hat{\lambda}_{cv}$. To obtain training-test prediction power, we perform 5 fold cross validation, using $\hat{\lambda}_{cv}$ and report average training and average test R^2. We denote this quantity as *Predicted* R^2.

3.1.1 Variable selection and prediction power

The primary aim to do LASSO is to see the variables selected by it. We however, for the sake of completeness discuss it's predictive power as well.

- Using 10-fold cross-validation, we obtained $\hat{\lambda}_{CV} = 0.0773$.
- Using the value of $\hat{\lambda}_{CV}$, the model parameters are estimated. Since we are not interested in developing a prediction model here, we do not present the exact estimated values of the parameters for now.
- LASSO, when applied to our dataset drops 29 variables indicating towards their insignificance or redundancy. For each of the 11 attributes, we summarize the number of corresponding parameters (each parameter corresponding to a distinct time point) that were significant in LASSO in Table 3.

The prediction power, measured in terms of *Predicted* R^2 using the model obtained are in Table 2. The plot of number of model parameters and MSE cost against $\log(\lambda)$ are available is Figure 2.

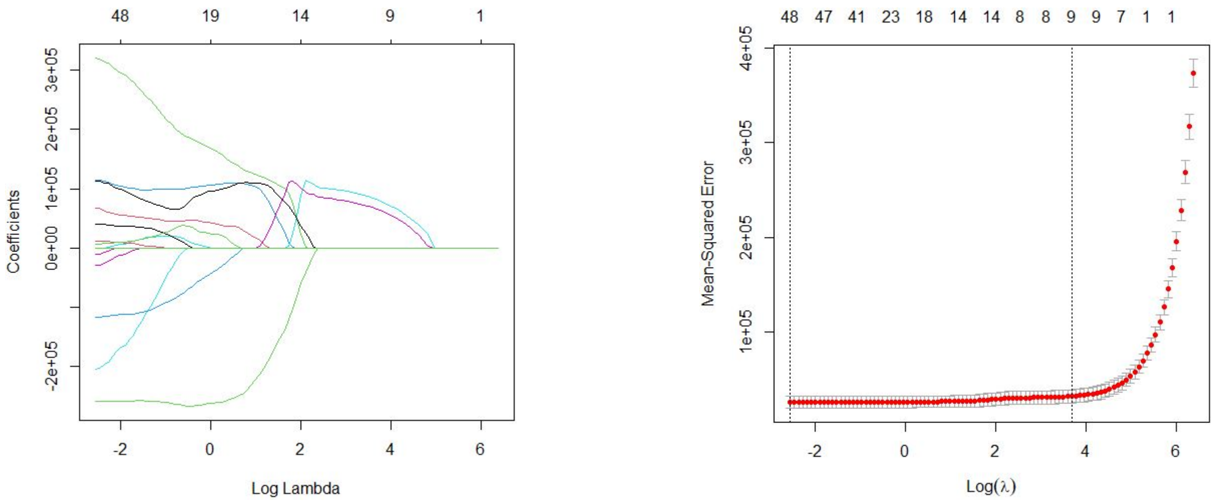


Figure 2: Plots of model parameters and MSE cost against $\log(\lambda)$

Dataset	***Predicted*** R^2
Training	0.935
Test	0.93

Table 2: LASSO Prediction Results

3.2 Variable selection using Linear Model

We take up the same problem of establishing the variable significance in the classical linear regression setup. We take the entire dataset and perform t-test to establish significance of the 77 variables.
We fit an additive linear model on the dataset. 30 parameters turn out insignificant at 10% level of significance. The number of significant parameters corresponding to each attribute is summarized in Table 3.

3.3 Observations

The summarized results of variables chosen by LASSO and Linear Regression for attribute are tabulated in Table 3. We observe that for each of the seven attributes, viz., NCD, AI, AS(NA), NAC, AS(NAC), NA, NAD), the parameters over at least 4 time points are selected in both LASSO and Linear Regression. The number of significant parameters are smaller for BL, CS and AT, with a striking difference between the two methods in case of the attribute ADL. This suggests towards the significance of the former attributes (as most of the time point effects were held significant/selected by the two methods). To further strengthen our assumption, we present a visual inspection of correlation between the variables and the response in the next

Variables	**LASSO**	**Linear Regression**
NCD	4	5
AI	7	7
AS(NA)	7	5
BL	3	3
NAC	4	7
AS(NAC)	6	7
CS	2	3
AT	1	0
NA	5	6
ADL	4	0
NAD	5	4

Table 3: Summary of the number of significant parameters corresponding to each attribute

section, that would also give some interesting insight into the behavior of these attributes.

3.4 Visual inspection of correlation

Observations from LASSO and linear regression prompt us to look at the correlations of the predictors with the response. We begin with searching for some trend in the correlation between each attribute and the response over the 7 time points. The plot is provided in Figure 3. The R implementation can be found in Appendix A.1.

The plot reveals a few interesting features:

- 7 attributes (NCD, AI, AS(NA), NAC, AS(NAC), NA, NAD) are highly correlated with the response over all 7 time points while the other 4 have very low correlation over all time points. We shall refer to the former group as influential attributes and the latter as non-influential attributes, in the sense of correlation.
- For the 7 influential attributes, the trends in correlation over the time points are nearly similar.
- The 4 non-influential attributes can be subdivided into two clusters consisting of BL, CS and AT, ADL. The trends for all 4 are similar, with the maximum correlation at time point 3.
- It is interesting to observe that the 7 attributes that have at least 4 selected time points corresponding to them for both LASSO and linear regression, are precisely the influential attributes. The others comprise the non-influential attributes.

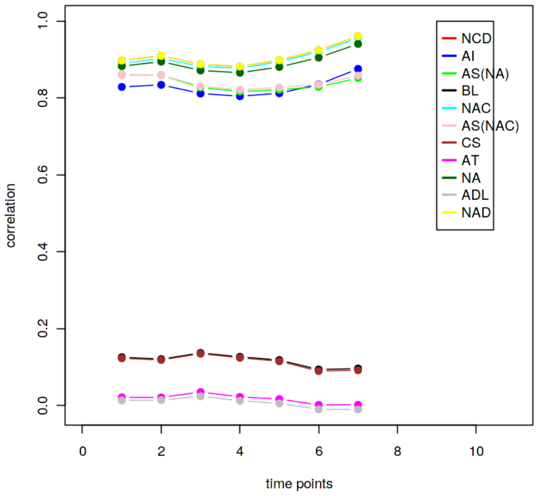


Figure 3: Plot of correlations of predictors with response

To explain this behaviour, we need to look into the precise definition of the attributes. The mathematical definition of the variables are beyond the scope of this article. We thus refer the reader to [2] for the same. Here, however, we give a brief motivation for this phenomenon. We find that the variables BL, AT, ADL are defined as ratio of two variables, both of which increase for a buzz event. So, unless we know the increment rates of numerator and the denominator, we cannot guess how the resultant ratio would behave for buzz events(response). Thus when a topic becomes 'buzz', the quantities in both the numerator as well as the denominator increases, which doesn't result in much increase in the fraction. A similar thing happens if the topic is 'non-buzz'. For example, consider BL, which is defined as the ratio of NCD and NAD (Table 1). For a buzz event, both NAC and NCD increases, which doesn't result in much change in BL, as compared to the same for a non-buzz event.

This may be the reason, variables CS, BL, AT, ADL are *non influential.* The other variables, by definition, tend to increase for buzz events. Consequently, they exhibit high correlation with response for which we termed them as *influential.*

3.5 Our proposal of variable selection

Before proceeding any further, we remove outliers from the dataset, that might hinder our inference. In that regard, we use Cook's distance to identify high leverage points. The Cook's distance for a data point x_i is:

$$d(x_i) = \frac{(\widehat{\beta} - \widehat{\beta}_{-i})^T (X^T X)(\widehat{\beta} - \widehat{\beta}_{-i})}{p\widehat{\sigma}^2},$$

where $p = 53$ is the number of predictors here. We remove all those points for which $d(x_i) > 4/n$, where n is the number of observations. We consider those points as high leverage points. The relevant R-codes are

available in Appendix A.1.
Based on the previous results, especially the correlations, we propose an additive linear model with reduced number of variables and investigate the model thus obtained. We consider the following variables for the model:

- For each of the influential attributes, we keep all 7 parameters corresponding to the 7 distinct time points as correlations for each with the response is quite high.
- For each of the non-influential attributes, we keep the parameter that has the highest correlation with the response. For each attribute, this corresponds to the third time point.

We now have 53 variables instead of 77. A quick glimpse at the BIC and multiple R^2 values for the model compared to that of the complete additive linear model with 77 parameters provided in Table 4, suggests that the reduced model is at par with, if not better than, the complete linear model. Thus, we can use the reduced model for testing significance of the variables. Establishing the significance of the attributes is not straightforward here. For the non-influencing attributes, we have only one parameter (corresponding to the one time point) under consideration, which can be used to assess it's significance. However, for the influencing attributes, we have seven time points under consideration, and hence no single parameter that would establish their significance. We performed t-tests to establish significance of the coefficients and found that for the influencing attributes, coefficients corresponding to at least four of the seven time points were significant, while for the non-influencing attributes, the effects of AT and ADL turned out insignificant, while that of CS and BL were significant. See Appendix A.2 for the R-implementation. Hence, we conclude **the attributes NCD, AI, AS(NA), BL, NAC, AS(NAC), CS, NA and NAD bear significant effect on the response while the effects of AT and ADL are insignificant.**

Model	**Predictors**	**Multiple R^2**	**BIC**
Full Model	77	0.9546	6172785.77
Reduced Model	53	0.9546	6172695.21

Table 4: Model comparison: Full vs. the model with dropped variables

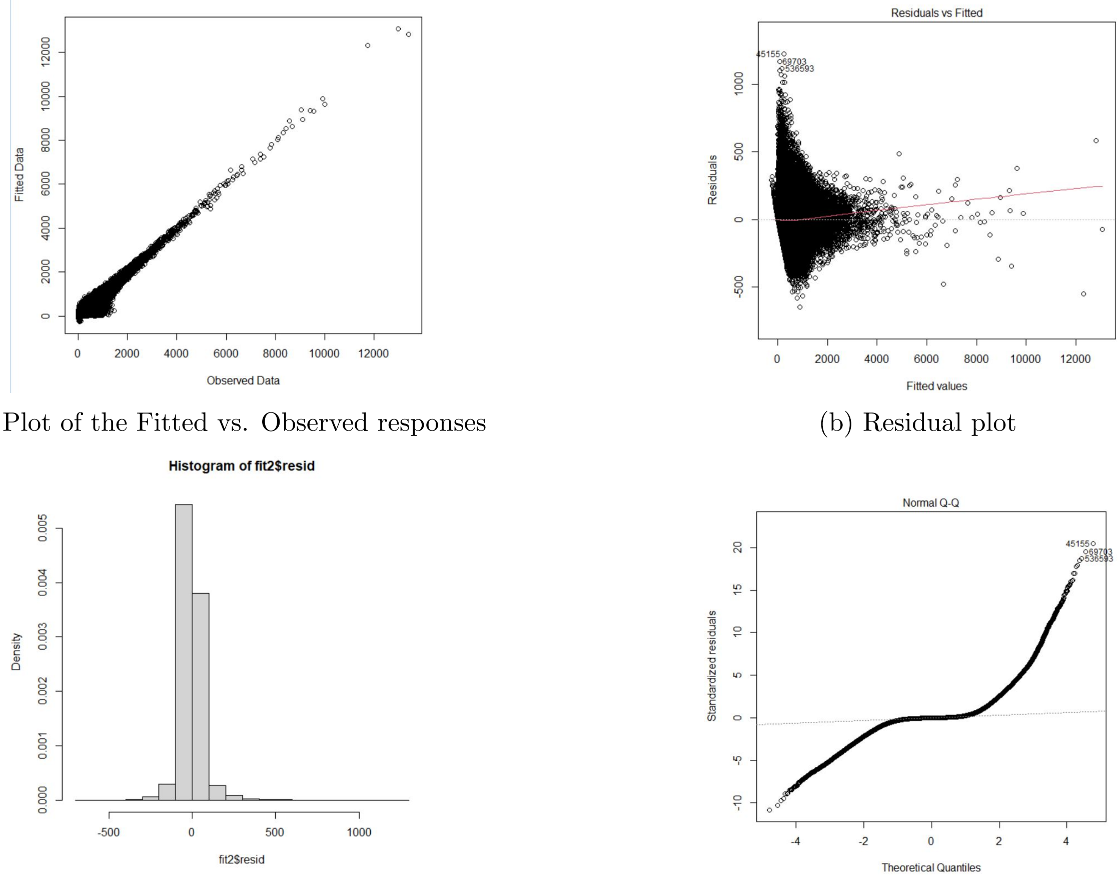
4 Developing a prediction model

We have already considered the question of establishing the significance of various variables. Now, we turn our attention to the problem of developing a prediction model. We put forward two proposals: The Reduced Linear Model we developed and a Semi Parametric Model which will described in Section 4.2.

4.1 A Reduced Linear Model

We decide to proceed with the model proposed in Section 3.5. We have already established its efficiency in terms of multiple R^2 value and BIC. Although AT and ADL are insignificant, we wish to keep them for the sake of prediction.
We first look into the diagnostic plots of the fitted model in Figure 4.

Figure 4a shows the plot between the fitted and observed response values. The plot resembles a straight line approximately, showing that our model has a good predictive power. However, the other plots analyzing the error do not look satisfactory. The residual plot in Figure 4b clearly shows that our assumption of homoscedasticity is under question as they don't appear to be a random scatter and the correlation slightly deviates from zero. From Figures 4c and 4d, it is clear that our assumption of normality doesn't hold. Both the histogram as well as the QQ-plot of the errors show that there is significant deviation at the tails from normality.
With these observations, one might not be very convincing to use the linear model. Though it is true that this model happens to have a good predictive power, as can be determined from the R^2 value in table 4, it should however be kept in mind that these values are prone to overfitting and might be misleading sometimes. As the assumptions under which we propose a linear model doesn't seem to hold good, we will propose our approach to form a prediction model in the hope of improving prediction power in the next section. We will use leave-one-out cross validation to compare it's performance with this linear model.



(a) Plot of the Fitted vs. Observed responses

(b) Residual plot

(c) Histogram of the residuals

(d) QQ-plot of the residuals

Figure 4: Diagnostic plots for the Reduced model

4.2 A semi-parametric approach

In spite of the prediction properties, the diagnostic plots the reduced linear model might make it unconvincing to consider a it. Besides, a non-parametric approach always allows for a chance of improvement in prediction accuracy and hence it's only worthwhile to explore this direction. We initially motivate our semi-parametric proposal, discuss a method for fitting and prediction, and subsequently compare it's performance with the general linear model.

4.2.1 Our Proposal

The main motivation behind considering a semi-parametric model is not to let go interpretability completely, but still trying to improvise over the standard linear model. We propose to have a model in which effect of each of the variables is linear in the time points within each one of them and the effect of the variables are combined non-parametrically.
Assume that our data is

$$\left\{\left(y_i, \boldsymbol{x}_i^{(1)}, \ldots, \boldsymbol{x}_i^{(11)}\right) : i = 1, \ldots, n\right\},$$

where $\boldsymbol{x}_i^{(j)}$ is the seven time point covariate values for the j^{th} attribute of the i^{th} observation, making $\boldsymbol{x}_i = (\boldsymbol{x}_i^{(1)}, \ldots, \boldsymbol{x}_i^{(11)})^T$ to be the complete set of predictors for the i^{th} observation.
Assume that $\beta_i^{(j)}$ is the effect of the f^{th} time point in predicting effect of the i^{th} attribute, where $i = 1, \ldots, 7$ for the influential attributes and $i = 3$ for the non-influential attributes. Then set

$$\boldsymbol{\beta}_i = (\beta_i^{(1)}, \ldots, \beta_i^{(7)})^T \text{ or } \beta_i^{(3)},$$

depending on whether the attribute is influential or non-influential respectively. Then set

$$\boldsymbol{f}_i = \boldsymbol{X}_i \boldsymbol{\beta}_i, i = 1, \ldots, 11 \tag{1}$$

to be the effect of the i^{th} attribute. Here $\boldsymbol{X}_i$ is the data matrix corresponding to the i^{th} attribute for the respective time points. For a covariate $\boldsymbol{x} = (\boldsymbol{x}^{(1)}, \ldots \boldsymbol{x}^{(11)})^T$, define

$$\boldsymbol{f}(\boldsymbol{x}) = (\boldsymbol{\beta}_1^T \boldsymbol{x}_1, \ldots, \boldsymbol{\beta}_{11}^T \boldsymbol{x}_{11}). \tag{2}$$

Then, for an unknown covariate $\boldsymbol{x}$, the predicted response is:

$$\widehat{y}(\boldsymbol{x}) = \frac{1}{k} \sum_{i \in N_k(\boldsymbol{f}(\boldsymbol{x}))} y_i, \tag{3}$$

where $N_k(\boldsymbol{f}(\boldsymbol{x}))$ is the set of k nearest neighbors of $\boldsymbol{f}(\boldsymbol{x})$ from the set

$$\{\boldsymbol{f}(\boldsymbol{x}_i) : i = 1, \ldots, n\}.$$

Thus the model has the set of parameters $\{\beta_i^{(j)} : j \in N_i, i = 1, \ldots, 11\}$, where $N_i = \{1, \ldots, 7\}$ for influential attributes and $N_i = \{3\}$ for the non-influential attributes. We also have a tuning parameter k, the number of nearest neighbors to consider.

4.2.2 Estimation and tuning of model parameters

We begin with estimation of the model parameters (the time point effects of each of the attributes).

- **Estimation of the time point effect coefficients**:
  The approach we adopt is rather ad hoc. The method might intuitively seem convincing, however we won't provide a theoretical justification. We will see that this choice achieves considerable accuracy.
  We estimate β_i, for $i = 1, \ldots, 11$, by the least square estimator of

  $$\boldsymbol{y} = \boldsymbol{X}_i \boldsymbol{\beta}_i + \boldsymbol{\epsilon}, \tag{4}$$

  where $\boldsymbol{y}$ is the vector of responses and $\boldsymbol{X}_i$ is, as previously defined, the sub-data matrix corresponding to observations of the respective time points for the i^{th} attribte only.
  Let's denote this estimate by $\widehat{\beta_i}$.
  A motivation behind considering the least square estimator of Equation 4 might be that, we can consider the fitted values based on the time points of a particular attribute only, to be the *part of the response* explained by that attribute only. Here the *part* happens to be a vector of observations.
  We now try to consider these 11 many fitted values as predictors and try to see how well they explain the response. We use kNN regression to do non-parametric smoothing and predictions. The next thing we consider is to tune the parameter k.

- **Choice of k:**
  We use a simple leave one out cross validation to determine the choice of k. We iteratively drop each of the observation from the dataset, and train the model on the remaining dataset. We thereafter consider the predicted output for the dropped observation. For each choice of k, we output the PRESS R^2, given by:

  $$\text{PRESS } R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \widehat{y}_{i,-i})^2}{\sum_{i=1}^n (y_i - \bar{y}_{-i})^2}$$

  We considered values of k ranging from 1 to 25. The plot of PRESS R^2 across k and some relevant values of PRESS R^2 for certain k are available in Figure 5.

  We see that, with $k = 17$, we obtain the highest value of PRESS R^2 after which the values start falling. Thus we will use $k = 17$ for training our model.

The estimated coefficients for the time effects specific to the variables are summarized in Table 5. All the relevant R-codes for the methods developed in this section can be found in Appendix A.3.

k	PRESS R^2
12	0.9426566
13	0.9427563
14	0.9428034
15	0.9427653
16	0.9427742
17	**0.942815**
18	0.9427398
19	0.9427204
20	0.9426820
21	0.9427204
22	0.9426820

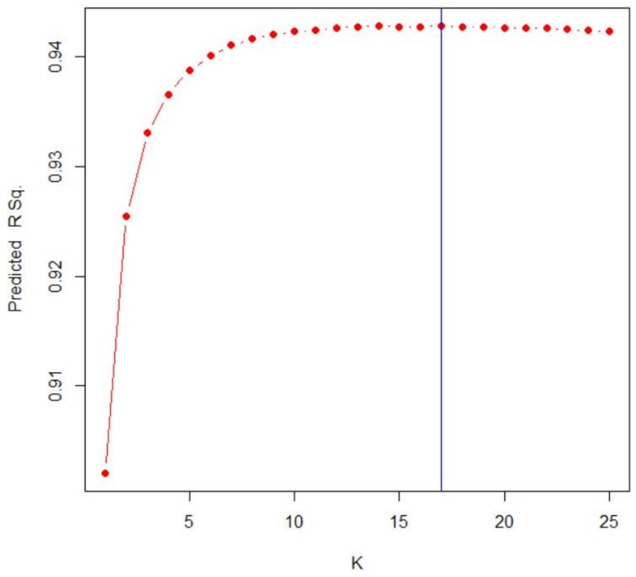


Figure 5: The PRESS R^2 values corresponding to different k's : LOO CV

Variables	T1	T2	T3	T4	T5	T6	T7
NCD	0.0346	0.1145	0.0658	0.0417	0.0614	0.0903	0.5419
AI	0.3018	0.2428	0.1528	0.1261	0.1041	0.1215	0.7166
AS(NA)	2.39×10^5	2.40×10^5	7.74×10^4	3.52×10^4	3.02×10^4	3.732×10^3	$1.42\ \times10^5$
BL	-	-	132.486	-	-	-	-
NAC	0.0008	0.1004	0.0602	0.0379	0.0638	0.0908	0.5306
AS(NAC)	3.79×10^5	3.96×10^5	1.32×10^5	6.69×10^4	5.88×10^4	2.60×10^4	2.98×10^5
CS	-	-	130.851	-	-	-	-
AT	-	-	10.46	-	-	-	-
NA	0.0434	0.1597	0.0848	0.0522	0.0719	0.0939	0.5849
ADL	-	-	6.429	-	-	-	-
NAD	0.0327	0.1142	0.065	0.0415	0.0613	0.0900	0.5423

Table 5: Estimated coefficients of the model for various time points of the respective attributes ($\boldsymbol{Ti}$'s)

4.2.3 Comparison with the general linear model

We now try to see how much we are improvising, if at all, by using this semi-parametric model instead of the reduced additive linear model. In order to do that, we fitted the reduced linear model with same regard to the time points as with our semi-parametric approach, that is, considering all seven time points for the influential attributes and the third one for the non-influential attributes amounting to a total of 53 predictors. We calculated the PRESS R^2 (by performing the exact LOO CV as we did with our model in Section 4.2.2). The results are available in Table 6.

We see that our method provides a marginal improvement in prediction accuracy over the general linear model. Our model achieves about 94.3% accuracy (PRESS R^2) as compared to about 93.3% for the additive general linear model. This is just a mere 1% increase. Thus, our model doesn't perform exceptionally well as compared to what the reduced linear model does. However, this serves as a milestone for the endeavour to achieve better results with the consideration of non-parametric methods. We did not resort to completely non-parametric methods (like complete kNN regression), in which case we would have to let go of interpretation completely.

Method	**PRESS** R^2
Semi-Parametric Model ($k = 17$)	**0.943**
Reduced Linear Model	0.933

Table 6: Performance our model alongside the general additive model

5 Conclusions

In this article, we have attempted a thorough analysis of the 'Buzz in social media' dataset corresponding to the Twitter platform. High dimensionality and multicollinearity were among the prominent challenges in the analysis. We acknowledged that traditional ANCOVA based methods cannot be used to establish significance of the attributes. We performed variable selection using t-tests and LASSO and through a visual inspection of correlation, to form a reduced model with 53 variables performing as good as the original model with 77 variables. Using this model we established the significance of the attributes NCD, AI, AS(NA), BL, NAC, AS(NAC), CS, NA and NAD and also concludedthat AT and ADL are not significant. The visual inspection of correlation also suggested that presence of two groups of attributes, one that bears high correlation with the response across time points and the other that shows low correlation. We tried to give an explanation for this phenomenon. We also saw that the change in correlation with time points show similar pattern within each group, across the attributes of the group.

Finally we developed prediction models. The classical linear model resulted in questionable diagnostic plots prompting us to look for different methods. We proposed a semi-parametric model that improvised over the PRESS R^2 as compared to the reduced linear model marginally, thereby yielding accuracy of 94.3%. We however suggest that more sophisticated models hold potential for further improvising on the accuracy.

In conclusion, we present some drawbacks and scope of improvements over our methods:

- We observed from Figure 3 that the change in correlation with time bears almost the same pattern for all the influential variables, and a different pattern for all the non-influencing variables. We however, did not take this fact into account, in our analysis. We can gain more insight, if we attempt an analysis in this direction.
- As was mentioned, the estimation procedure for our semi-parametric approach was ad hoc and not supported by theoretical results. We can try to develop more advanced techniques for estimation of that part.
- Lastly, we always have room for improvement, if we resort to more sophisticated models for prediction.

References

[1] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[2] Kawala. François, Chouakria. Ahlame, Gaussier. Eric, and Eustache Dimert. Prédictions d'activitédans les réseaux sociaux en ligne, 2013.

A R Codes

The R codes used in our work are listed here. The data can be downloaded from [1]. To run these codes, please install the packages `FNN` and `glmnet`.

A.1 Visualization, Cleaning, etc.

```
#read data
data = read.csv('../input/twitter-buzz/Twitter.data', header = FALSE)#this line
     would change depending on the reader's system
data = as.data.frame(data)
names(data)[78] = 'resp'

#removing points with high leverage (cook's distance)
fit = lm(resp~., data = data)
n = nrow(data)
cd = cooks.distance(fit)
bad = which(cd>4/n)
data_n= data[-bad, ]
#data_n is the dataset after removing the high leverage points

#the correlation plot
plot_time_interact<-function(data,name=NULL){
          #plot of correlation of different variables with the responses
     if(is.null(name)){
          name = as.character(1:11)
     }
     p = ncol(data)
     f<-function(x){
          return(cor(x,data[,p]))
     }
     vec = apply(data,2,f)
     print(length(vec))
     cols = c('red', 'blue', 'green', 'black', 'cyan',
     'pink', 'brown', 'magenta', 'dark green', 'grey', 'yellow')
     ma = max(vec)
     mi = min(vec)
     plot(0, xlim = c(0,11), ylim = c(mi,ma),
     xlab = 'time points', ylab = 'correlation', ty ='n')
     for(i in 1:11){
          #print(length(1:7))
          #print(length(vec[(7*(i-1)+1): 7*i]))
          points(x = 1:7, y = vec[(7*(i-1)+1): (7*i)], pch=19, type ='b', col =
               cols[i] )
     }
     legend(9,1, legend = name, col = cols, lwd = 2)
}
plot_time_interact(data_n, name = c('NCD','AI','AS(NA)','BL','NAC','AS(NAC)',
'CS','AT','NA','ADL','NAD'))
```

A.2 Prediction model: LASSO and LM

```
#LASSO
twitter = data
lasso.fit.full<-glmnet(as.matrix(twitter[,1:77]),as.numeric(twitter[,78]),
    family="gaussian",alpha=1)
```

```
cvlasso.full<-cv.glmnet(as.matrix(twitter[,1:77]),as.numeric(twitter[,78]),
    alpha=1,nfolds=10)
plot(lasso.fit.full,xvar="lambda")
plot(cvlasso.full)

#Predicted R^2
pen_cross_val <- function(dat,k_sample,a,lambda)
{
  dat=as.matrix(dat)
  #A permutation of indices. n/k from the beginning for the first group, the
      next form the second group and so on.
  ind = sample(nrow(dat),nrow(dat),replace=F)
  samp_size=floor(nrow(dat))/k_sample
  Rsq=0
  Rsq_t=0
  for (i in 1:(k_sample-1))
  {
    t_ind=ind[((i-1)*samp_size+1):(i*samp_size)]
    y_train=dat[-t_ind,78]
    subfit=glmnet(dat[-t_ind,1:77],y_train,family="gaussian",alpha=a)
    y_hat=predict(subfit, s = lambda, newx = dat[t_ind,1:77])
    y=dat[t_ind,78]
    y_hat_train=predict(subfit, s = lambda, newx = dat[-t_ind,1:77])
    Rsq=Rsq+1-((sum((y-y_hat)^2))/(sum((y-mean(y))^2)))
    Rsq_t=Rsq_t+1-((sum((y_train-y_hat_train)^2))/(sum((y_train-mean(y_train))
        ^2)))
  }
  t_ind=ind[((k_sample-1)*samp_size+1):(nrow(dat))]
  y_train=dat[-t_ind,78]
  subfit=glmnet(dat[-t_ind,1:77],y_train,family="gaussian",alpha=a)
  y_hat=predict(subfit, s = lambda, newx = dat[t_ind,1:77])
  y=dat[t_ind,78]
  y_hat_train=predict(subfit, s = lambda, newx = dat[-t_ind,1:77])
  Rsq=Rsq+1-((sum((y-y_hat)^2))/(sum((y-mean(y))^2)))
  Rsq_t=Rsq_t+1-((sum((y_train-y_hat_train)^2))/(sum((y_train-mean(y_train))^2)
      ))
  Rsq=Rsq/k_sample
  Rsq_t=Rsq_t/k_sample
  cat("Train␣R^2:␣",Rsq,"\nTest␣R^2:␣",Rsq_t)
}
Rsqcv.lasso.5=pen_cross_val(twitter,5,1,cvlasso.full$lambda.min)

#Linear Model
fit = lm(resp~., data = data_n)
toreg = c(1:21, 24, 29:42, 45, 52, 57:63, 66, 71:77)
length(toreg)
dat = data_n[,c(toreg,78)]
names(dat)[54]='resp'
fit2 = lm(resp~., data = dat)   #the reduced model
summary(fit)    #summaries of the two models
summary(fit2)
BIC(fit)    #BICs of the two models
BIC(fit2)

#diagnostic plots for the reduced model
plot(fit2$fitted,fit2$resid)    #residual
plot(dat$resp, fit2$fitted) #observed vs. fitted
hist(fit2$resid)    #histogram
```

```
qqnorm(fit2$resid)
qqline(fit2$resid)  #qq plot
```

A.3 The semi-parametric model

```
#coefficient estimates
fit11 = lm(data_n[,78]~., data = data_n[,1:7])
fit12 = lm(data_n[,78]~., data = data_n[,8:14])
fit13 = lm(data_n[,78]~., data = data_n[,15:21])
fit14 = lm(data_n[,78]~., data = as.data.frame(data_n[,24]) )
fit15 = lm(data_n[,78]~. ,data = data_n[,29:35])
fit16 = lm(data_n[,78]~. ,data = data_n[,36:42])
fit17 = lm(data_n[,78]~. ,data = as.data.frame(data_n[,45]))
fit18 = lm(data_n[,78]~. ,data = as.data.frame(data_n[,52]))
fit19 = lm(data_n[,78]~. ,data = data_n[,55:63])
fit110 = lm(data_n[,78]~. ,data = as.data.frame(data_n[,66]))
fit111 = lm(data_n[,78]~. ,data = data_n[,71:77])

#cross validation
point_score<-function(i){
    lm1 = lm(dat[-i,]$resp~dat[-i,1:7])
    lm2 = lm(dat[-i,]$resp~dat[-i,8:14])
    lm3 = lm(dat[-i,]$resp~dat[-i,15:21])
    lm4 = lm(dat[-i,]$resp~dat[-i,22])
    lm5 = lm(dat[-i,]$resp~dat[-i,23:29])
    lm6 = lm(dat[-i,]$resp~dat[-i,30:36])
    lm7 = lm(dat[-i,]$resp~dat[-i,37])
    lm8 = lm(dat[-i,]$resp~dat[-i,38])
    lm9 = lm(dat[-i,]$resp~dat[-i,39:45])
    lm10 = lm(dat[-i,]$resp~dat[-i,46])
    lm11 = lm(dat[-i,]$resp~dat[-i,47:53])
    dat_temp = cbind(lm1$fitted,lm2$fitted,lm3$fitted,lm4$fitted,lm5$fitted,
    lm6$fitted,lm7$fitted,lm8$fitted,lm9$fitted,lm10$fitted, lm11$fitted , dat$
        resp[-i])
    pattern = c(predict(lm1,dat[i,]),predict(lm2,dat[i,]),predict(lm3,dat[i,]),
    predict(lm4,dat[i,]),predict(lm5,dat[i,]),predict(lm6,dat[i,]),predict(lm7,
        dat[i,]),
    predict(lm8,dat[i,]),predict(lm9,dat[i,]),predict(lm10,dat[i,]),
    predict(lm11,dat[i,]))
    val = as.vector(knn.reg(train = dat_temp, test = pattern, y = dat$resp[i])$
        pred)
    return((val-dat$resp[i])^2)
}
k_score<-function(k){
    #calculates the loo score for a certain k
    vec = 1:nrow(dat)
    return(sum(sapply(vec,point_score)))
}
loo_semiparametric<-function(k_vec){
    #returns a dataframe containing two columns, the k value and the loo score
    denom = 0
    for(i in 1:nrow(dat)){
        denom = denom + (mean(dat$resp[-i])- dat$resp[i])^2
    }
    scores = 1-(sapply(k_vec, k_score)/denom)
    return(data.frame(K = k_vec, R2 = scores))
}
```