Predicting the popularity of online articles with random forests

Gitte Vanwinckelen and Wannes Meert
Department of Computer Science
KU Leuven, Belgium
firstname.lastname@cs.kuleuven.be

ABSTRACT

In this paper, we describe our submission to the predictive web analytics Discovery Challenge at ECML/PKDD 2014. The main goal of the challenge was to predict the number of visitors of a web page 48 hours in the future after observing this web page for an hour. An additional goal was to predict the number of times the URL appeared in a tweet on Twitter and the number of times a Facebook message containing the URL was liked. We present an analysis of the time series data generated by the Chartbeat web analytics engine, which was made available for this competition, and the approach we used to predict page visits. Our model is based on random forest regression and learned on a set of features derived from the given time series data to capture the expected amount of visits, rate of change and temporal effect. Our approach won second place for predicting the number of visitors and the number of Facebook likes, and first place for predicting the number of tweets.

1. INTRODUCTION

Along with the increasing importance of visibility and impact on the Internet, prediction of the popularity of online content has become an important research topic. Interest in this topic is fueled by prospects of better catering to the user's interests, and of increasing revenue by allowing for more focused advertising. Besides predicting the popularity of the actual content (e.g., news articles, blog posts, YouTube videos), the problem can be broadened to predicting the popularity on social media sites. Nowadays, content is linked and discussed on sites such as Facebook and Twitter, increasing its exposure and therefore in turn its popularity. Understanding these interactions and also predicting the popularity of an article on social media sites contributes to the appeal of this research topic.

In the literature, a number of different approaches can be found to tackle the problem. The most popular prediction model is a linear regression model that predicts the logarithmically transformed future popularity from the popularity after an initial period of time [4, 11, 15, 19, 20]. These models also often incorporate features expressing the popularity of the content on social media sites [14]. Alternatively, Lee et al. use a model from survival analysis, not predicting the exact visitor count, but instead the likelihood that a site receives at least a certain number of visitors [13]. One can also use a nearest neighbors like approach, where the prediction happens based on matching similarly shaped time series shaped time series [7]. Finally, besides time series data, the content of a web page can also be used to predict its popularity [12, 2]. Some authors do not focus on the exact visitor count. For instance, Ahmed et al. use a dynamic clustering approach to predict the evolution of popularity of a web page over time, and Kim et al. categorize articles into four categories expressing their popularity [1, 11].

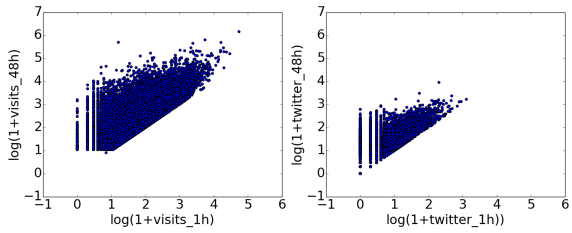
This paper takes the approach of learning a model to predict the popularity of a web page based on the evolution of the number of page views. Additionally, we learn models for predicting the article's popularity on Facebook and Twitter. Our approach is based on random forest regression and learned on a set of features derived from the given time series data to capture the expected amount of visits, rate of change and temporal effect. Specifically, we generate new features by applying the Fourier transform on the time series data. While this transformation is best known for being able to extract periodic trends from signals, our approach is an illustration of how it is also able to capture meaningful properties of non periodic signals.

2. DATASET AND TASK DESCRIPTION

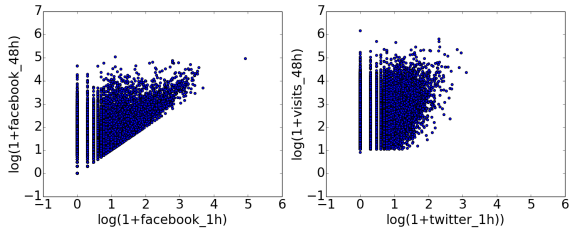
The data consists of a collection of time series that was collected by the real-time analytics engine Chartbeat. For each of a set of hundred websites, a collection of 600 URLs was monitored during 48 hours. Every five minutes, information is collected about the number of visitors in that interval, the number of times the URL appeared in a *tweet* on Twitter, the number of times a Facebook message that contained the URL was *liked*, and the average time a visitor was active on the page. Additionally, the website's ID, and the weekday and hour the URL was posted is available.[1] The prediction task is defined as follows: Based on the time series data from the first hour, predict for each URL the total number of visitors, tweets, and likes after 48 hours. It was ensured that each URL had at least ten visits.

[1]The time recorded is presumed to be server side time but this was not explicitely stated in the original data description.

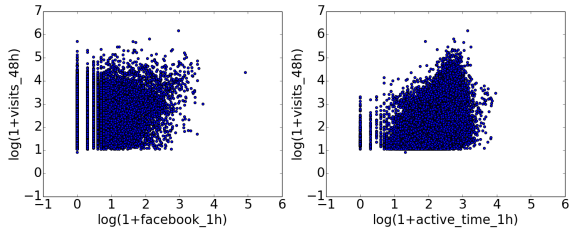
ECML/PKDD Discovery Challenge on Predictive Web Analytics.
Nancy, France, September 2014.



(a) $\rho = 0.73$ (b) $\rho = 0.82$



(c) $\rho = 0.72$ (d) $\rho = 0.32$



(e) $\rho = 0.32$ (f) $\rho = 0.43$

Figure 1: The first three scatter plots show $\log1p(x_{1h})$ **versus** $\log1p(x_{48h})$ **with respectively** x **equal to** *visits*, *twitter*, **and** *facebook*. **The last three scatter plots show the** $\log1p(visits48h)$ **versus** $\log1p(x_{1h})$.

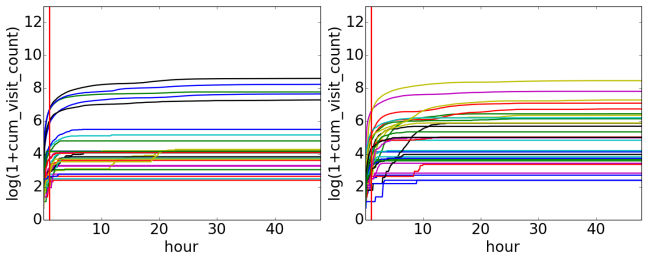
For each of the hundred websites that were monitored, the data from 300 URLs were fully disclosed to the participants. This data was to be used for training and evaluation purposes. The other 300 URLs were part of the secret test set; only the data from the first hour was available. For these URLs, three targets had to be predicted 48 hours in the future: Total number of visitors, tweets, and likes. The error of a solution is measured by the mean squared error (MSE) of $\log1p(x) = \log(x + 1)$, with x one of the three targets. The participants are first ranked in terms of the MSE for the number of visitors, so priority is given to this target.

3. EXPLORATORY DATA ANALYSIS

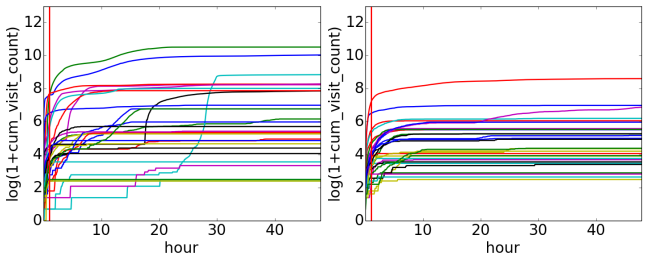
This section discusses the structure of the data as a motivation for the design choices made during the development of the predictive model.

3.1 Transformation to log space

It has been demonstrated before that for online content, a linear relationship exists between the log visitor counts at two different moments in time [4, 19]. We investigate whether this also holds for our problem. Note that instead of the log transform, we use the log1p transform to avoid the problem of zero counts. Figure 1a plots the log1p



(a) Website 31 (b) Website 49



(c) Website 69 (d) Website 76

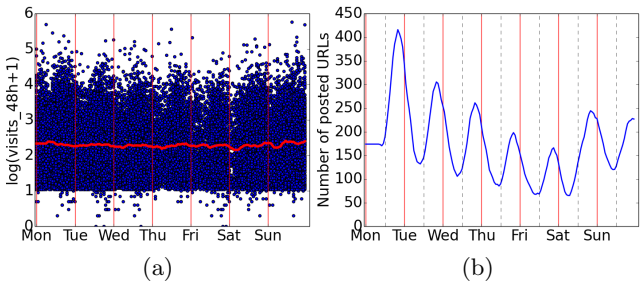
Figure 2: The cumulative visitor count time series for 30 random URLs from four different websites.

transform of the total number of visitors after one hour, $\log1p(visits1h)$, versus the total number of visitors after 48 hours, $\log1p(visits48h)$. Similar scatter plots are constructed for the number of tweets and likes, shown in Figures 1b and 1c. For each target, we also computed the Spearman rank correlation (shown below each plot). This correlation coefficient can take values from -1 to 1, with values close to |1| indicating a strong correlation. We see that indeed, for each of these variables, there is a linear relationship between the count after one hour and that after 48 hours. The correlation is highest (0.82) for the number of tweets. This may explain why our random forest model has the highest accuracy for predicting the number of tweets (see Section 4.3). Finally, we observe that for $\log1p(visits48h)$ with respectively $\log1p(twitter1h)$, $\log1p(facebook1h)$, and $\log1p(time1h)$, the correlation is much lower.

3.2 Cumulative visitor count

Figure 2 shows the 48-hour cumulative time series for the number of visits for 30 random URLs from four different websites. The first hour after the URL was posted is indicated by a vertical red line.

While these figures show just a small sample of URLs, the plots nevertheless hint that the characteristics of the curves may be different for each website. For instance, we observe a difference in minimum and maximum number of visitors, and in the spread of the final total visitor count. Additionally, we observe a difference in the shapes of the curves. For instance, in these samples, for websites 31, 49 and 76, many curves saturate fairly quickly, after which the number of visitors stagnates. Whereas for website 69, we observe 'bumps' in some of the growth curves. Such a bump can be caused for instance by the website using a model were popular stories are promoted to a front page, thus suddenly experiencing a



(a) (b)

Figure 3: (a) Scatter plot of log1p(*visits48h*) **versus posted hour and weekday, together with the mean visitor count plotted in red. (b) Smoothed time series of the mean number of posted URLs.**

surge in visitors because the URL gets more exposure [14]. Notice that, as these bumps occur after the 12th datapoint (the boundary of the input data), the final visitor count will probably be difficult to predict accurately for such cases.

3.3 Time dependence

Because the day and hour a URL was posted is known, we investigate whether the time series for the visitor count exhibits periodicity. Figure 3a shows a scatter plot of the log1p of the total visitor count after 48 hours in function of the day and time the URL was posted, together with the mean number of visitors, indicated in red. A new day starts at midnight and is indicated by a red vertical line; noon is indicated by a black dotted line. We expected to find a trend in the popularity of the URLs, with for instance URLs posted at noon being much more popular than URLs posted at 4am [19]. However, no significant periodicity was detected in this signal that would indicate such a daily trend.

Another aspect is the number of URLs posted at each point in time, shown in Figure 3b. In this case, a trend *is* observed, which can be visualized using an exponential moving average with a window size of 6 hours. We see that regardless of the day, most URLs are posted somewhere during the evening, and few URLs are posted before noon. Furthermore, we see that most URLs are posted on Monday, and after that there is a decreasing trend with the least number of URLs posted on Friday.

These results suggest that most content providers are from the same region, whereas the readers are from various regions. However, when constructing the same plots for smaller, random subsets of websites, we noticed periodicity in the visitor count in some of the subsets. This indicates that readers of a single website are in fact often from the same region. Nevertheless, when the data from the different sites is merged, the trends are obfuscated due to the presence of time zones. As time could be an important predictive feature, an easy solution to still incorporate it as a feature could be to learn a separate model for each website.

3.4 Website statistics

The websites present in the data set exhibit a large variety in the absolute number of visitors. This is shown in Figure 4 by means of a box plots for log1p(*visits48h*) for a random sample of 30 websites, ordered according to the mean number of visitors. The mean is indicated by a blue dot and the median by a red line. The upper and lower limits of the box represent respectively the first and third quartile. The length of the whiskers corresponds to 1.5 times the interquartile range (IQR), and outliers are indicated individually by '+'.

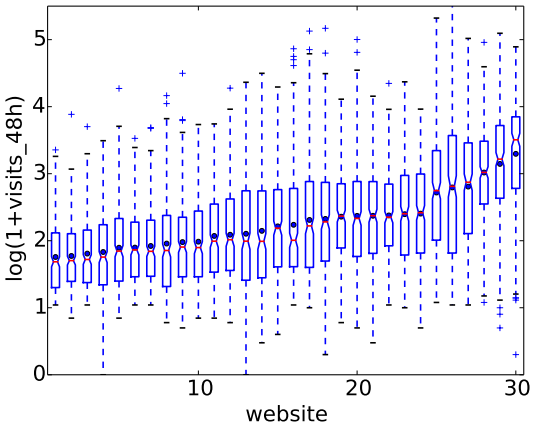


Figure 4: Box plots of log1p(*visits48h*) **for a random selection of 30 websites, ordered by increasing average visitor count.**

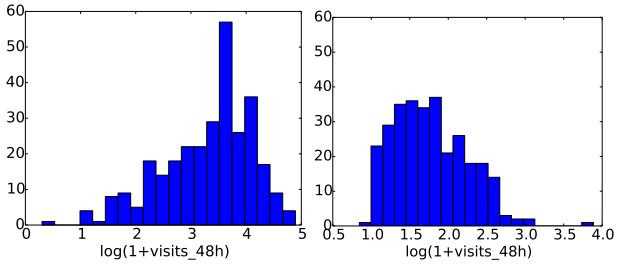
Figure 4 shows some interesting differences between the websites. First, we see that for websites with a small mean number of visitors, the median number of visitors is typically smaller than the mean. This could suggest a left skewed distribution where most URLs receive few visitors, and a few URLs receive a large number of visitors. Oppositely, for websites with a large mean number of visitors, the median is typically larger than the mean. This could suggesting a right skewed distribution. In fact, this is confirmed by Figure 5 showing the histograms of log1p(*visits48h*) of respectively the website with (a) the smallest mean visitor count, and (b) the largest mean visitor count. Second, we also notice a difference in the spread of the visitor counts. For websites with a small mean number of visitors, the spread is smaller than for websites with a large mean number of visitors. This is especially true for the visitor counts in the first quartile. These observations show that the distributional properties of the visitor count are different for each website, so that it could be useful to learn a model for each website separately.

4. PREDICTION METHODOLOGY

4.1 Feature generation

Our model was trained on a dataset that consisted of four types of features. First, for each of the four original time series (visits, tweets, likes, and active time), we computed the log1p of the sum of the counts for the first hour of data.

Second, we extracted information about the time the URL was posted. Although our initial data analysis showed that no periodic trend existed in the data for the 100 websites overall, we also found that this is not necessarily true for the individual websites. We therefore included the hour a URL was posted as a feature. We divided the 24 hour interval into four intervals: [0h,6h), [6h,12h), [12h,18h), and [18h,24h).



(a) (b)

Figure 5: Histograms of log1p(*visits48h*) **of respectively the website with (a) the smallest mean visitor count, and with (b) the largest mean visitor count.**

These intervals represent respectively night, morning, afternoon, and evening. We did not discriminate on weekday, as often, for a given website, there was no data available for each weekday.

A third type of feature is derived from the Fourier transform of the cumulative time series for visits. As an example, Figure 6 shows three cumulative visitor count curves (black), together with their Fourier approximation based on the DC component, the harmonic with the lowest frequency, and its complex conjugate. The Fourier spectrum for the positive frequencies is shown below. These figures illustrate the reason for choosing to model the cumulative visitor counts, and not the original time series as is typically used to obtain a *periodogram* [17]. Namely, we are not interested in brief fluctuations of the visitor count, but in the growth characteristics of the total visitor count, for instance, its slope. This information is implicit in the Fourier spectrum. Compare for instance, the Fourier spectrum of the second and the third curve: the second curve has a large DC component, with the other coefficients being small. The third curve, on the contrary, has a 'bump', causing the coefficients of the harmonic components to be larger than those of the second curve.

Based on the Fourier transform, we generate three features as follows. For each cumulative time series, we first compute the moduli of the complex coefficients of the discrete Fourier transform. Next, we keep the DC component of the Fourier transform, the mean of the second to fourth coefficient, and that of the fifth to the seventh coefficient.

Finally, in an effort to boost performance, we also included the sum of two previously defined features, namely:

$$\text{log1p}(\sum visits) + \text{log1p}(\sum active\_time).$$

This feature has a high Spearman rank correlation coefficient with respect to the visitor count (0.72), although we expect it to be highly correlated with both log1p($\sum visits$) and log1p($\sum active\_time$).

4.2 Conditional random forest

We considered three different approaches for learning a predictive model, based on different ways of splitting the data. (1) Learning a single predictive model on the data from all

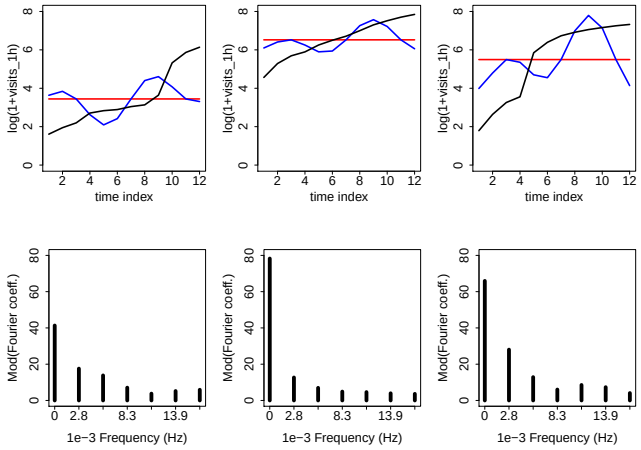


Figure 6: Top: Cumulative 1 hour visitor count (black), DC component of the Fourier transform (red), Fourier series consisting of the DC, the 1st harmonic, and its complex conjugate (blue). Bottom: Frequency spectrum of the Fourier transform of the the time series above.

the websites. (2) Learning a separate model for each website. This is possible because the data in the secret test set originates from the same websites as those in the training set. (3) Combining the previous two approaches by first learning a model on all of the data, after which the residuals of the predictions can be fitted for each site separately. Experiments showed that the second approach was the most successful, therefore, we focus on this strategy.

As a classifier we decided to use a random forest, which we have extensive experience with. This model has shown to be successful in various applications in the past [5, 6]. Furthermore, in several Kaggle competitions the top contribution was based on random forests [3, 8, 16]. This model has the advantages that good performance can be obtained without much parameter tuning, and that results are interpretable by investigating the trees or computing the variable importances [18].

Specifically, we chose the random forest based on conditional inference trees from the R package 'party' [9, 10]. Traditional decision trees are first fully grown based on metrics such as the Gini coefficient or information gain, after which the tree is pruned to avoid overfitting. Conditional inference trees instead use a statistical test for independence between the predictors and the target to do variable selection, after which they again use statistical testing to decide on the split. Furthermore, instead of pruning, a stopping criteria is used. The advantage of this approach is that it avoids overfitting and selection bias towards predictors with many possible splits.

4.3 Results

To obtain the optimal model, we evaluated a range of configurations on the fully disclosed training data set. Table 1 presents the results in terms of MSE(log1p(x_{48h})) for approach (2) discussed in Section 4.2. The performance of the

target	RHO MSE	test set MSE
visits	0.76	0.99 (2^{nd} place)
tweets	0.35	0.65 (1^{st} place)
likes	1.5	1.38 (2^{nd} place)

Table 1: Performance evaluation results for the model

model was first evaluated with three times repeated hold-out with random resampling (RHO), using respectively 2/3 and 1/3 of the data for the training and test sets. This process is repeated for each of the 100 classifiers, after which the mean squared errors are averaged. We also evaluated the model on the secret test data, of which the score was eventually used to rank the participants.

The results for repeated hold-out should be interpreted with caution. We found that adding new features to the data sometimes decreased the repeated hold-out error, but increased the test set error. Especially, for the modeling of each site separately, we suspect that the model was overfitting because of the small size of the training sets (200 instances). This means that the performance estimated by repeated hold-out was not always a good indicator for the performance of the model on the secret test data.

To interpret our model, we randomly selected one train/test split to compute variable importances of the features [18]. Because we learned a classifier for each website separately, this results in 100 variable importances. We therefore present the results as histograms. Figure 7 shows the results for the features which have the largest mean variable importance when predicting the visitor count: The visitor count, sum of visitor count and active time, posted hour interval, and the mean of the second to fourth Fourier coefficient (the first non-DC term). Interestingly, tweets, likes, and active time were found not to have much predictive value for estimating visitor count. The results for predicting the number of tweets and likes are similar: The one hour data of the predicted feature always has the largest variable importances.

5. CONCLUSION AND FUTURE WORK

This paper describes our solution to the ECML/PKDD 2014 Discovery Challenge, predicting the popularity of a web page based on time series data from the Chartbeat web analytics engine. Our approach consisted of learning a random forest on a set of features derived from the original time series, capturing information about the initial visitor count, the growth rate, and temporal effect. Our results showed that this approach worked well and scored among the top contenders.

In future work, we envision this work to be extended in three potential directions. First, our data analysis suggests that some groups of websites do show similar behavior, but when merging all of them together, specificities are averaged out. A solution to this problem is to learn a single model per website. However, this significantly decreases the size of the training set. An alternative could be to cluster websites that show similar behavior.

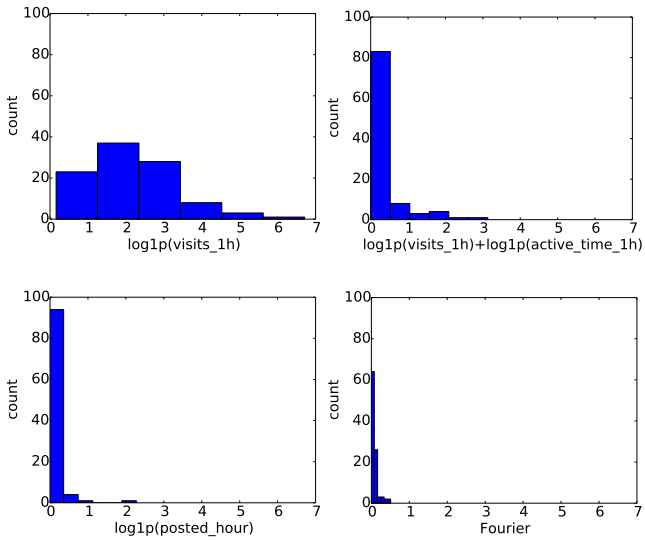


Figure 7: Histograms of the variable importances of the most important predictors for predicting the visitor count, with 'Fourier' the mean of the 2^d to 4^{th} Fourier coefficient.

Second, we generated new features by computing the discrete Fourier transform of the cumulative visitor count series. The Fourier transform provides information about the frequency content of this time series. Alternatively, one could also use a wavelet transform. This could provide information about the shape of the curve both in the time and frequency domain.

Third, since the goal of this challenge was to predict three correlated targets, it might be interesting to look into multivariate prediction.

6. ACKNOWLEDGMENTS

The authors would like to thank prof. Hendrik Blockeel for his support and feedback.

References

[1] M. Ahmed, S. Spagna, F. Huici, and S. Niccolini. A peek into the future: predicting the evolution of popularity in user generated content. In *Proceedings of the 6th International Conference on Web Search and Data Mining*, pages 607–616. ACM, 2013.

[2] Y. Bei, C. Miao, and K. Linchi. Toward predicting popularity of social marketing messages. In *Proceedings of the 4th International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, volume 6589 of *Lecture Notes in Computer Science*, pages 317–324. Springer, 2011.

[3] H. Ben. Air quality prediction hackathon. `http://blog.kaggle.com/2012/05/01/chucking-everything-into-a-random-forest`, 2012. Accessed: 2014-07-30.

[4] C. Castillo, M. El-Haddad, J. Pfeffer, and M. Stempeck. Characterizing the life cycle of online news stories us-

ing social media reactions. In *Proceedings of the 17th Conference on Computer Supported Cooperative Work*, pages 211–223. ACM, 2014.

[5] E. De Paula Costa, L. Schietgat, R. Cerri, C. Vens, C. N. Fischer, C. M. Carareto, J. Ramon, and H. Blockeel. Transposable element annotation using relational random forests. Benelux Bioinformatics Conference, Dec. 2013.

[6] T. Fannes, E. Vandermarliere, L. Schietgat, S. Degroeve, L. Martens, and J. Ramon. Predicting tryptic cleavage from proteomics data using decision tree ensembles. *Journal of Proteome Research*, 12(5):2253–2259, Apr. 2013.

[7] C. George H., N. Stanislav, and S. Devavrat. A latent source model for nonparametric time series classification. In *Proceedings of the 27th conference on Advances in Neural Information Processing Systems*, pages 1088–1096, 2013.

[8] J. P. González-Brenes and C. Matías. Rta freeway travel time prediction. `http://blog.kaggle.com/2011/03/25/jose-p-gonzalez`, 2011. Accessed: 2014-07-30.

[9] T. Hothorn, K. Hornik, C. Strobl, and A. Zeileis. *party: A Laboratory for Recursive Partytioning*, 2014. R package version 1.0-15.

[10] T. Hothorn, K. Hornik, and A. Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674, 2006.

[11] S.-D. Kim, S.-H. Kim, and H.-G. Cho. Predicting the virtual temperature of web-blog articles as a measurement tool for online popularity. In *Proceedings of the 11th International Conference on Computer and Information Technology*, pages 449–454, 2011.

[12] H. Lakkaraju and J. Ajmera. Attention prediction on social media brand pages. In *Proceedings of the 20th International Conference on Information and Knowledge Management*, pages 2157–2160. ACM, 2011.

[13] J. G. Lee, S. Moon, and K. Salamatian. An approach to model and predict the popularity of online contents with explanatory factors. In *Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology*, pages 623–630. IEEE Computer Society, 2010.

[14] K. Lerman and T. Hogg. Using a model of social dynamics to predict popularity of news. In *Proceedings of the 19th International Conference on World Wide Web*, pages 621–630. ACM, 2010.

[15] H. Pinto, J. M. Almeida, and M. A. Gonçalves. Using early view patterns to predict the popularity of youtube videos. In *Proceedings of the 6th International Conference on Web Search and Data Mining*, pages 365–374. ACM, 2013.

[16] D. Sculley. Results from a semi-supervised feature learning competition. `http://eecs.tufts.edu/~dsculley/papers/semisupervised-feature-learning-competition.pdf`, 2012. Accessed: 2014-07-30.

[17] R. H. Shumway and D. S. Stoffer. *Time Series Analysis and Its Applications*. Springer, 3rd edition, 2011.

[18] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9(307), 2008.

[19] G. Szabo and B. A. Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, Aug. 2010.

[20] A. Tatar, J. Leguay, P. Antoniadis, A. Limbourg, M. D. de Amorim, and S. Fdida. Predicting the popularity of online articles based on user comments. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, pages 67:1–67:8. ACM, 2011.