QUOIN

WE SIMPLY BUILD GREAT SOFTWARE

PublicRelay Case Study - Toyota ~~and CES~~

Presented by
Bill Mitchell
Feb 9, 2017

PublicRelay Case Study - Toyota and CES

MIT students analyzed 241,000 tweets relating to Toyota hybrids.

Their goal was to see if they could get algorithmic results that matched the human analysis of content. They performed **Topic Modeling** using Latent Dirichlet Allocation and BiTerm Topic Models, then used analytics to perform Tag Prediction. Their Tag Prediction was based on Labelled data, Feature Extraction, Training and Testing. Their final task was to perform **Reach Analysis**, to find out if they could predict the value of a tweet based on metadata, the importance of the twitter user, prior history, etc. This work was done as a Supervised Machine Learning problem, as a Linear Regression model, but also they experimented with Random Forest models.

Text Analysis, Twitter, Machine Learning

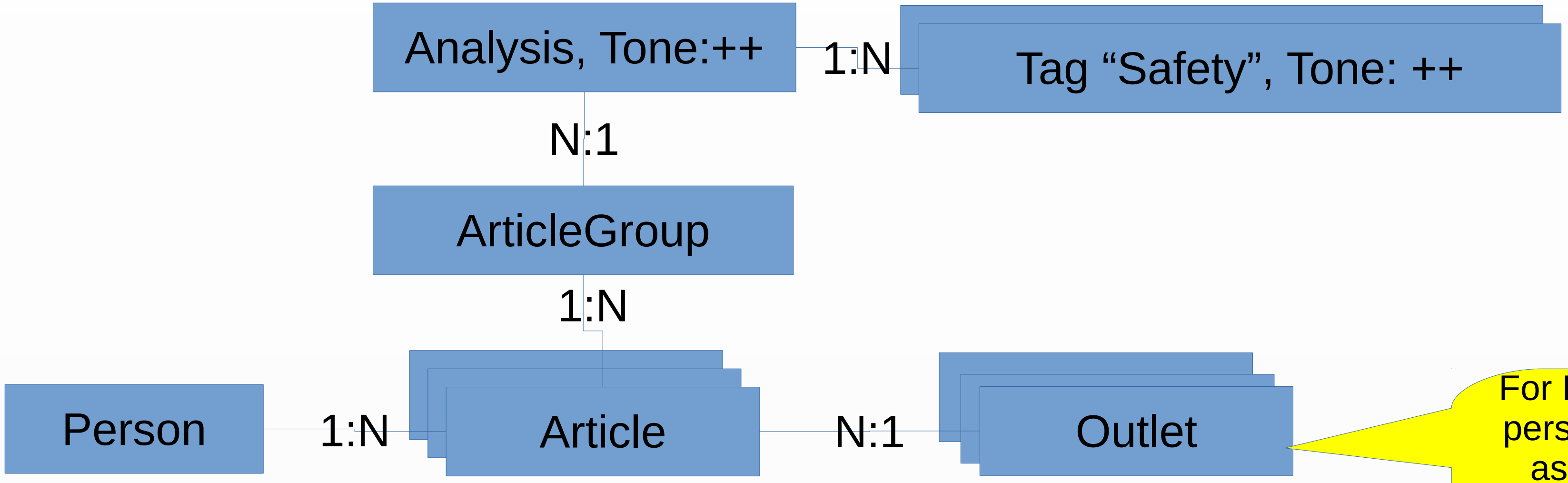
Extracting features

- The first challenge with the dataset was to extract meaning. The dataset provided consisted of labelled data, with RELEVANT and IRRELEVANT samples.
- Furthermore, the data contained examples of RETWEETS. The RETWEETS were interesting for the reach analysis, but not for the topic modeling perspective.
- The students had to do a fair amount of data wrangling just to normalize the data set and get ready to do the first part.

What was in the dataset we gave them

Feature(s)	Notes
Article ID	Unique ID of every tweet in the dataset. Retweets do *not* share Article IDs.
Article Group ID	Unique ID of every *unique* tweet in the dataset. Original tweet and all retweets share this value.
Article Title	Actual raw tweet text.
Other Article Info	Other article information includes publication dates, and others
Outlet Name	Authors' Twitter handles.
Outlet Reach/Circulation	Follower counts of each Tweet author at the time of publication.
Other Outlet/Author Info	Much of these remaining fields were blank or meant to be disregarded.
Topic Tags	Tags for 163 topics or subjects, including relevance.

Very quick look at an (abbreviated) data schema



For PublicRelay, we treat the person who made the Tweet as an ‘outlet’, rather than allowing all tweets to vault Twitter.com into the #1 outlet of all time

Topic Modeling

Latent Dirichlet Allocation

BiTerm Topic Models

TF-IDF (term frequency–inverse document frequency)

Latent Dirichlet Allocation

In natural language processing, Latent Dirichlet Allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar.

Most people are familiar with TF-IDF for text corpora analysis – where you extract terms, then find the terms which are most prevalent, identify which terms are most unique to discern individual documents.

Latent Dirichlet Allocation

LDA, on the other hand, uses SVD (singular value decomposition) to identify a subspace of the TF-IDF to extract ‘themes’ from the terms.

The MIT students used the generated TERMS from the Tweet corpus that was supplied to them, and then fed those terms into a LDA engine, to automatically extract groupings of terms that represented detected themes.

Their code was written in Python, and leveraged the following libraries:

Matplotlib : Python plotting package (to visualize the results)

Numpy : NumPy: array processing for numbers, strings, records, and objects.

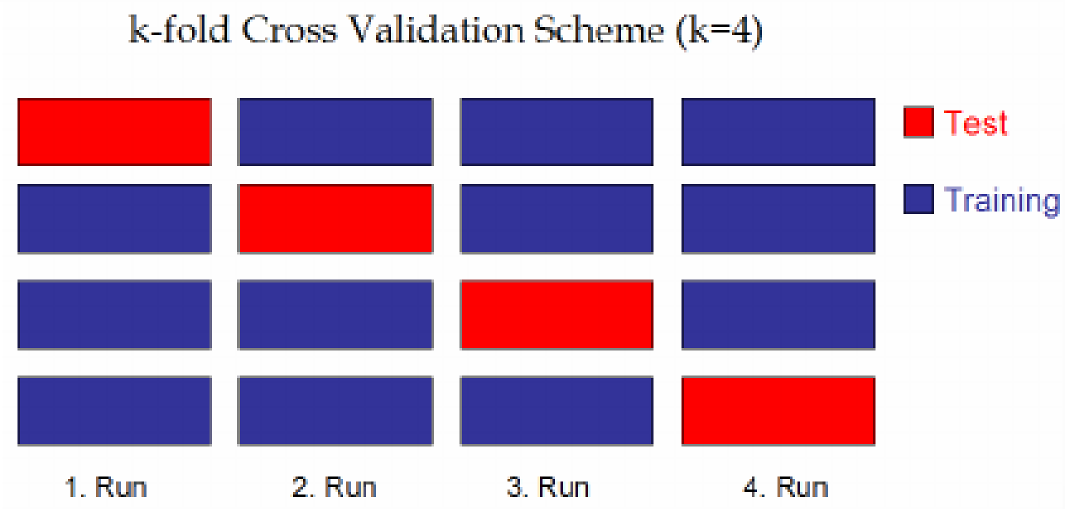
Pandas : Powerful data structures for data analysis, time series,and statistics

Latent Dirichlet Allocation

Unlike other clustering models, LDA allows a document to be associated with multiple topics.

In order to maximize utilization of the dataset provided (typical datasets for training systems are usually larger), a 10-fold cross validation approach was used.

- *In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data.*



BiTerm Topic Models

BiTerm Topic Modeling splits the input text into BiTerms (groupings of 2 words at a time), and attempts to match those grouping frequencies with projected outputs.

- Since the algorithm is using pairs of words, it doesn’t lose context the same way, “good car” has a different semantic meaning than {“good”,…,”car”} in a bag of words collection of terms.

(The team did not use the more recent Discriminative Bi-Term Topic Modeling approach, which places a filter on the words that are used to construct the Bi-Terms, allowing the algorithm to focus on higher value tuples.)

Reach Analysis

The team's challenge:

Identifying "significant" tweets based on topic inference. [..] This challenge has two goals: identify tweets that demand further tracking or direct engagement, or formulate messaging that Toyota might use to drive social media conversations themselves. Trained analysts currently perform this classification task. Machine identification would ideally reduce this cost and free human resources to apply judgment to tasks that no machine approach can yet handle.

Reach Analysis

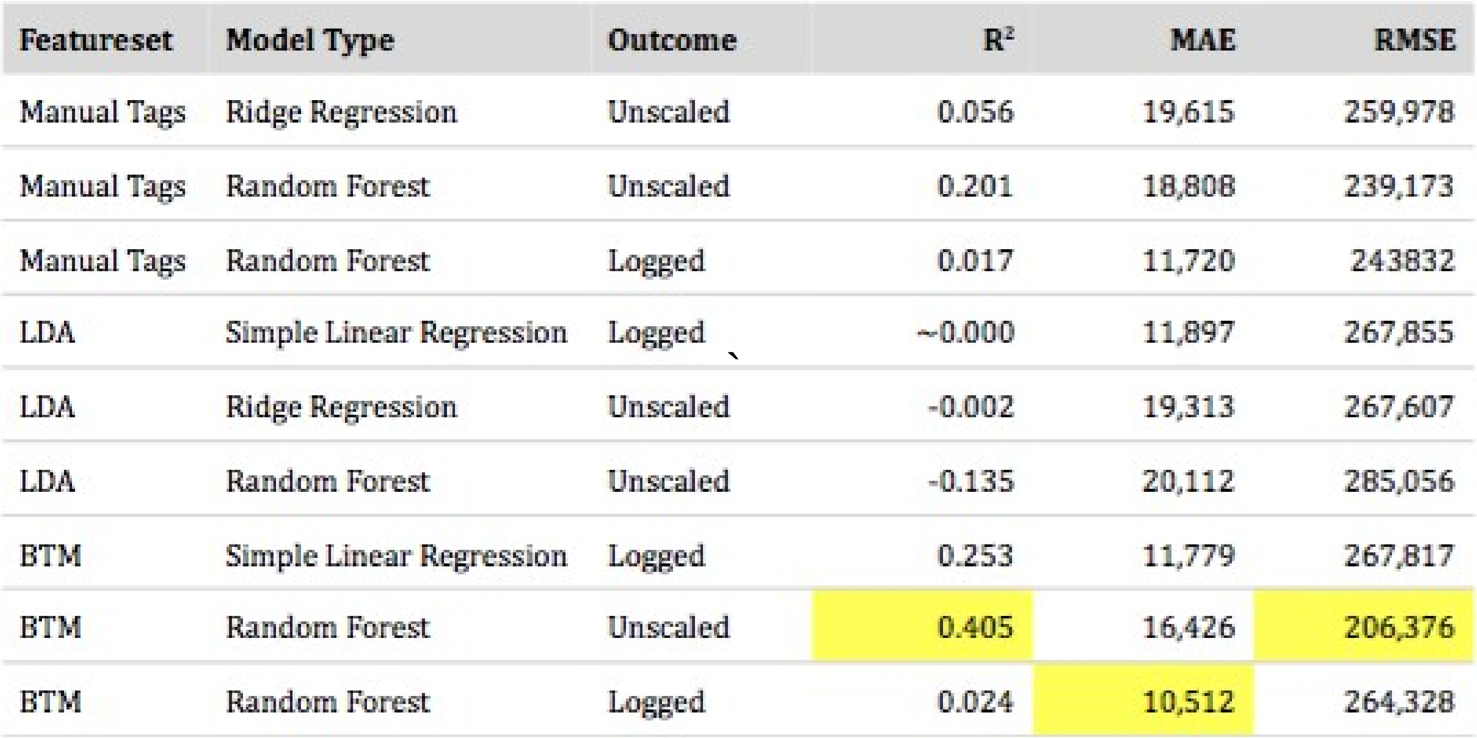
The team's result:

[..]focussed our efforts on linear regression models (both standard OLS and ridge–regularized) for "baseline" performance. Such models tend to train quickly and learn easily–interpretable parameters (feature weights).

Hypothesizing that models that detect "conditional" inluence of multiple topics would perform better, and that generating a large set of interaction features would lead to overitting in linear regression, we also experimented with Random Forest models.

Reach Analysis results

- R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.
- The R^2 result is interpreted as a % that the response variable was predicted by a linear model. In this case, only with the Random Forest Model Type did they achieve over a 40% accuracy.



Featureset	Model Type	Outcome	R^2	MAE	RMSE
Manual Tags	Ridge Regression	Unscaled	0.056	19,615	259,978
Manual Tags	Random Forest	Unscaled	0.201	18,808	239,173
Manual Tags	Random Forest	Logged	0.017	11,720	243832
LDA	Simple Linear Regression	Logged	~0.000	11,897	267,855
LDA	Ridge Regression	Unscaled	-0.002	19,313	267,607
LDA	Random Forest	Unscaled	-0.135	20,112	285,056
BTM	Simple Linear Regression	Logged	0.253	11,779	267,817
BTM	Random Forest	Unscaled	0.405	16,426	206,376
BTM	Random Forest	Logged	0.024	10,512	264,328

Their final conclusions

Text processing is a challenge:

- Overzealous cleaning risks removing important context in a terse medium.

Topic modeling can automatically discover latent topics from plain tweets:

- “Learned” topics are interpretable in some sense and reflect the true context.
- Help to reduce heavily labor-intensive analysis (manual tagging).
- May detect specific news stories—information that is useful but perhaps not generalizable.

Learned topics show some relationship to PublicRelay’s manual tags:

- TF-IDF has better prediction accuracy but is slower.
- LDA and BTM are faster but at the expense of accuracy.

Topic modeling generated useful predictors of tweet reach:

- The BTM approach captured some meaningful information that manual tags did not.
- Full interpretation of the Random Forest results remains elusive.