Exploring how headline topics affect A/B testing results on the Upworthy website using Machine Learning

Agathiyan Bragadeesh, Emre Mutlu, Juan J. Vazquez,* and Chun H. Yip

We firstly determined through a Mann-Whitney U test, that the median percentage increase in click rate (of articles) induced by changing the headlines of packages was significantly greater than the median percentage increase in click rate induced by changing the images of packages. Within package tests which involved solely the variation of headlines, the article with the maximum click rate amongst all others of the test had a median 32.4% greater click rate than the average click rate of all the articles in the test. In contrast, for package tests which involved solely the variation of images, this figure was 25.4%, 7% less than that for headline tests. In other words, we can conclude that the headline of the package is slightly, but significantly, more important in determining how popular it is, as varying the headline causes a greater median increase in the click rate of a package.

We found that there are statistically significant differences in the *click_rate*, *first_place*, and winner variables for the different headline categories. For example, a test rejected the hypothesis that the mean *click_rate* is the same for the 'HEALTHY LIVING' and 'WELLNESS' categories. Overall, this means that headline categories have a sizable impact on both user behaviour (whether or not they click on the package) and organisational behaviour (whether editors choose to publish that version of the package). However, how large that effect is compared to other factors such as the time when the tests were taken, or the detailed difference between headlines in the same topic, is yet to be investigated.

I. INTRODUCTION

We set out to see how headline topics affected various aspects of A/B testing, using data from the Upworthy Research Archive. The dataset held information about A/B tests conducted using packages (each package corresponding to a version of the article), and statistics relating to each packages performance in testing;

We labelled the headlines in the dataset using a classification model, and then using these labels we conducted a statistical analysis on 3 columns in the dataset: *click_rate*, *first_place*, *winner*.

- *click_rate*: ratio of clicks/impressions, where impressions is the total number of viewers of the package.
- *first_place*: whether package was recommended to Upworthy editors to publish on the website (1 per test week).
- *winner*: whether a package was published on the Upworthy website after the testing.

Furthermore, we used a clustering model to deduce similarities among the classified headlines within a category and among categories. This allows us to compare specific pairs of subsets which could showcase similar properties (click rate, etc..) e.g. health/wellness, science/green. These similarities can then be compared via hypothesis testing and the confusion matrix from the classification model.

* vazquezj@tcd.ie

II. GENERAL OVERVIEW OF THE DATASETS

A. Introduction of variables

As mentioned above, the first important step in our analysis was to create the variable *click_rate* as

$$click\_rate = \frac{clicks}{impressions} \tag{1}$$

which is to be used as a benchmark of the popularity or success of a package. Hereafter, we mainly compare the data fields *click_rate*, *first_place*, and *winners*.

The variables clicks and impressions have pretty clear meanings: they denote the raw test results. However, the meanings of *first_place* and *winner* are less clear. We would assume that *first_place* is chosen for each test, denoting the package with the highest *click_rate*. *Winner* is stated to be the packages chosen by editors. It is unclear how much information editors have, except that they knew which packages were labelled *first_place*. We do some exploratory data analysis to look at the *first_place* and *winner* variables.

• We note that *first_place* is selected for most (98.9%) but not all of the tests. Looking at the tests without a *first_place*, it seems like they are not significantly different to the tests with one, so this may be due to a data entry error.

In 87.4% of the cases, the *first_place* package corresponds to the one with maximum *click_rate* in a test. Looking at the data, at times when the *first_place* does not correspond to the one with highest *click_rate*, the two *click_rates* are pretty similar, so perhaps Upworthy staff exercised their own judgement in making recommendations when the test results were not clear-cut.

• As for *winner* field, we have the following table:

Number of *winners*	Number of tests	Percentage
0	24834	76.4%
1	7642	23.5%
2	11	0.0%

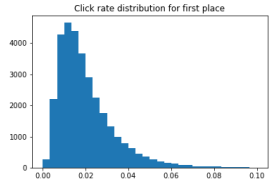
So most tests actually don't result in a *winner*. This shows that editors are more picky when choosing articles to post, perhaps because there is a limit in terms of how many articles they can post on the Upworthy site.

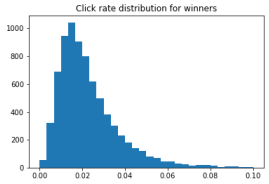
We also consider whether editors take into account the test results when making a decision, in particular the *first_place* variable. We observe that 74.6% of *winners* are *first_place* in their tests as well, showing that editors factor in the test results quite heavily. Looking at the *winners* that are not *first_place*, it seems that they have the same *headline*, but different *excerpt* or *lede*. This means the test results do not reflect the differences of the packages, since *excerpt* and *lede* are not shown in tests.

• Finally we look at the *click_rate* distribution of winner and *first_place* packages.

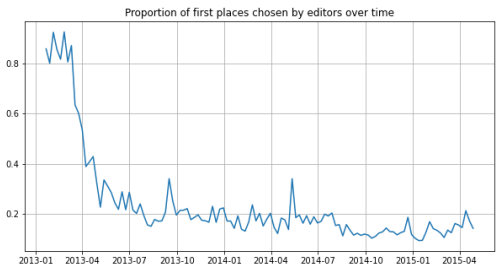
Packages	Mean *click_rate*
All	0.016
first_place	0.020
winner	0.023

Here we present histograms for *first_place* and *winner* packages:





We note that *winner* packages have a higher mean and median *click_rate*. This suggests that editors may have access to the *click_rate* data as well, hence they factor this into their decision-making. Alternatively editors actually have sufficient experience to predict the *click_rate* of a package such that they can choose the ones with higher *click_rate*. We also consider the proportion of *first_place* packages selected by editors over time.



Observe that the proportion starts being quite high, then rapidly decreases over the next few months, and stays constant at around 0.2 afterwards. This is most probably because more packages were tested as time advanced, and editors could only put a fixed number of packages on each week. (For reference, 1942 packages were tested in January-March 2013, and 7690 packages were tested in April-June 2013.)

The code for the above analysis is included in the exploratory.py file.

B. Impact of Headline vs Image Tests

In order to motivate our exploration of the impact of the category and type of a *headline* on a package's popularity, we decided to first explore the overall impacts of the A/B testing undertaken by Upworthy. We wanted to determine whether the variation of headlines did in fact result in significant differences in outcomes for the popularity and *click_rate* of the same article. Noting that users only saw the *headline* and image during the packages testing, we also decided to compare 'tests' which only varied the *headline* ("headline tests") with 'tests' which only varied the *image* ("image tests"). We did this in order to pinpoint the impact of solely varying the *headline* in testing on the popularity of an article, as well as to explore whether changing a *headline* or changing the *image* of a package were more likely to have an influence on its popularity in testing.

A single article "test" was determined by grouping all packages with *test_id*'s of the same value. A group of packages with the same *test_id* were designated to be a "headline test" if at least 2 of the headlines of the packages within a 'test' differed, while all the images for every package within this 'test' were kept the same. A group of packages with the same *test_id* were designated to be an "image test" if at least 2 of the images of the packages within a 'test' differed, while all the headlines for every package within this 'test' were kept the same. Groups of packages in which both the *image* and the *headline* were changed were designated to be "combination tests".

In order to quantify the impact of varying the images or headlines within a group of packages which represented the same article, we used the **clickrateIncrFromAverage** variable defined to be the percentage increase of the *click_rate* of the most popular package within a test (the package with the highest *click_rate*) from the mean *click_rate* of the packages within a test. This was done for every "test" in each of the three groups "headline tests", "image tests" and "combination tests". The percentage increases within each category of test were then ranked and their distributions were examined and compared. The percentage increases within each category of test were not normally distributed, and were heavily left-tailed, seen in the histograms in the Appendix.

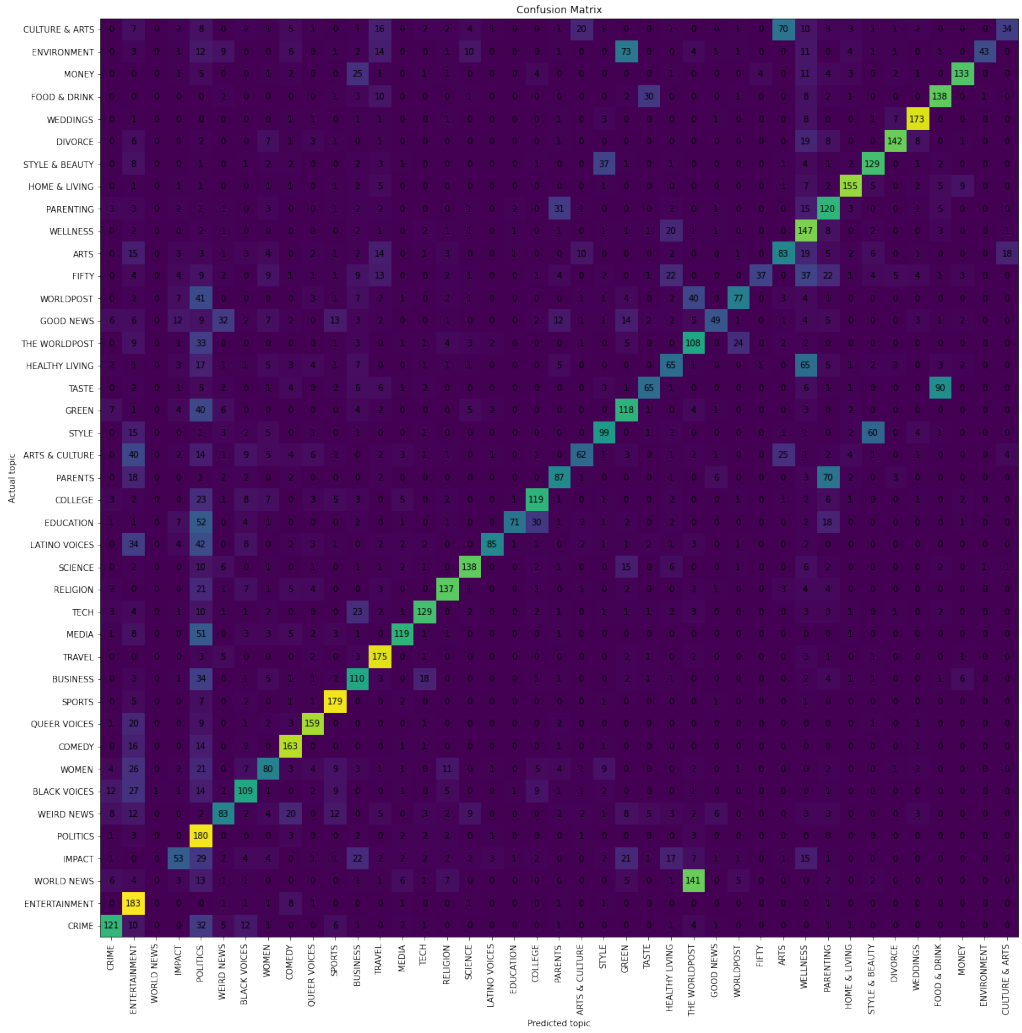


FIG. 1: Confusion Matrix from the Categories Model

III. METHODOLOGY

A. Classifying Headlines

To label headlines, we decided to use NLP to classify them. We utilised transfer learning, first using existing model BERT, and then fine tuning the model for our purpose over a dataset of labelled headlines []. Using this model we could then label our data.

BERT is a model originally trained on the tasks of language modelling and next sentence prediction. It utilises bidirectional transformers which helps the machine to read text in both directions, rather than typical RNNs which usually consider text unidirectionally. The pretrained BERT model includes only the transformer blocks, then we add a classification layer on top of a suitable shape for our task. BERT also uses a subword tokenizer, which allows it to handle text it may not have seen before; this is useful as headlines may use names of people and slang.

We split the News Category Dataset into an 80% training and 20% validation using a random split. The model was trained using an AdamW optimiser:

Since we were only fine tuning the model, to prevent overfitting we trained the model over 2 epochs.

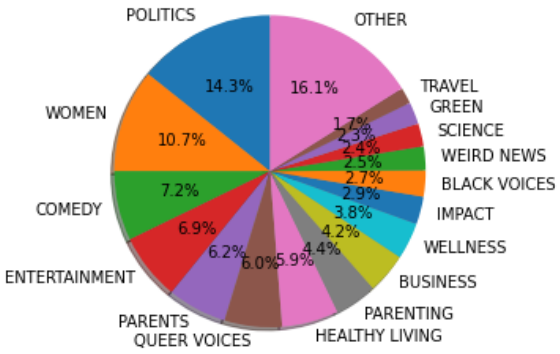
Epoch	Training loss	Validation loss	Accuracy
1	1.65	1.31	0.63
2	1.19	1.25	0.64

We also noted some of the *headline* categories were similar so we ran the model over 200 headlines of each topic, and generated a confusion matrix to see what the model got wrong when making predictions, and we could identify how we could use the model. See the confusion matrix in Fig NUMBER the next page.

From the confusion matrix above we note some things that stand out in particular:

- "ENTERTAINMENT" and "POLITICS" were overpredicted a lot, most likely because these are such broad topics encapsulating others, and inspecting the kaggle dataset we see they were also proportionally a large part of the dataset:
  - POLITICS ≈ 16.3% of data
  - ENTERTAINMENT ≈ 8.0% of data
- Similar topics are frequently incorrectly predicted as each other, most notably out of 200 "WORLD NEWS", 141 of them were predicted as "THE WORLDPOST". Similar relationships are found between:
  - "PARENTS" and "PARENTING"
  - "ARTS" and "CULTURE AND ARTS"

Then running the model over the packages we had, we can label all the data and see a general distribution of all the headline topics that were labelled:



B. Hypothesis Testing

We grouped the categories with not many headlines into one 'OTHER' category, so that sample sizes were not too small when carrying out statistical tests. Then we looked at the mean statistics for each category. As mentioned in the introduction, we carry out analysis on three variables: *first_place*, winner, and *click_rate*.

Using the confusion matrix, we determined pairs of categories which are 'similar' in nature and conducted statistical tests to see whether the variables for these categories are different. The point of these tests is that we wanted to see whether user and editor behaviour were influenced by the category of a news *headline*. Some news articles may have similar content, but a different way of framing the content results in a different category, hence attracting more or less people to click on them, or leading to editors picking these articles more or less often.

For the *first_place* and *winners* variables, we assume that for each category these variables follow a Binomial distribution (since they are either True or False for each package). Now assume category A has probability p_a being $True$ and category B has probability p_b being $True$. Let there be $n-a$ samples in category A and n_b samples in category B. With a 5% significance level, we test for:

$$\begin{aligned} \text{Null hypothesis}: &\quad p_a = p_b \\ \text{Alternative hypothesis}: &\quad p_a \neq p_b \end{aligned}$$

We assume the packages were sampled randomly (as part of the Upworthy dataset). Under the null hypothesis, we can estimate the overall probability of a variable being True as the weighted mean, i.e.

$$p = \frac{n_a p_a + n_b p_b}{n_a + n_b}$$

Also, since the sample sizes are large enough, we can apply the Central Limit Theorem to say that the means of the samples follow a Normal distribution centered at p with their respective standard error

$$\sqrt{\frac{p(1-p)}{n_a}} \qquad \sqrt{\frac{p(1-p)}{n_b}}$$

Hence $p_a - p_b$ follows a Normal distribution with mean 0 and standard error

$$\sqrt{p(1-p)\left(\frac{1}{n_a}+\frac{1}{n_b}\right)},$$

and the z-statistic can be calculated using the formula

$$\frac{p_a - p_b}{\sqrt{p(1-p)\left(\frac{1}{n_a}+\frac{1}{n_b}\right)}}$$

Note that the z-statistic follows a standard Normal distribution under the null hypothesis.

We carry out this hypothesis test manually.

For the *click_rate* variable, we first carry out a log transformation to make the distribution of the variable approximately Normal in each category. Then we calculate the mean and estimate the standard error for *click_rate* in each category. Let m_a and m_b be the mean log *click_rate* for category A and B respectively, let SE_a and SE_b be the standard error of the means, and n_a and n_b be the number of samples in each category. We carry out the following hypothesis test with a 5% significance level:

$$\begin{aligned} \text{Null hypothesis}: &\quad m_a = m_b \\ \text{Alternative hypothesis}: &\quad m_a \neq m_b \end{aligned}$$

Assuming the samples are random and Normally distributed, we can use the Welch t-test, where the t-statistic is calculated by

$$\frac{m_a - m_b}{\sqrt{\mathrm{SE}_a^2 + \mathrm{SE}_b^2}}$$

This follows a t-distribution with degrees of freedom approximated by

$$\frac{\left(\mathrm{SE}_a^2 + \mathrm{SE}_b^2\right)}{\frac{\mathrm{SE}_a^4}{n_a^{-1}} + \frac{\mathrm{SE}_b^4}{n_b^{-1}}}$$

We carry out this hypothesis test using SciPy.

C. Finding similarities among categories

Once each package *headline* has been categorised, we analyse the sentence similarity for each *headline*. This will allow us to find and visualise links among same-category headlines, or the complete opposite, an observation of how headlines belonging to specific categories do not share much common value.

We also seek to understand how the content of each category interacts with the rest of the dataset (all other categories). Thus allowing us to observe clear links among categories and their properties (*impressions*, *clicks*), which are also present in the confusion matrix, and in the hypothesis testing.

We found the most optimal method to find these links among categories was using a machine learning model via sentence embeddings with the Siamese BERT-network. The SentenceTransformers python framework provides a straightforward method to compute sentence embedding which can then be visualised using dimension reduction schemes such as t-SNE and UMAP.

In order to facilitate computation, we used 4 different models pre-trained on different large sets of data:

- All-mpnet-base-v2
- Msmarco-distilroberta-base-v2
- Paraphrase-distilroberta-base-v1
- stsb-roberta-large

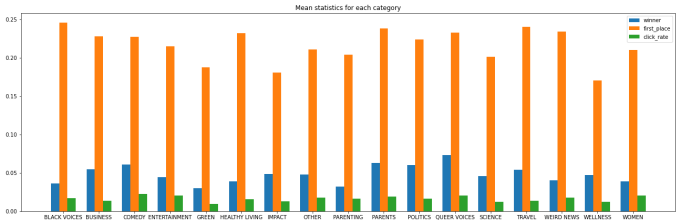
The output from the model was then processed using the dimension reduction algorithm Uniform Manifold Approximation and Projection (UMAP). Using this scheme we found the low dimensional embedding of the headlines to prepare the data for visualisation. Note that 30 neighbouring sample points and an effective minimum distance of 0.02 was used for the UMAP computation.

The data was then distributed into 22 bins and plotted. Note that this was found to be the best parameter value for visualisation purposes.

IV. RESULTS

A. Statistical analysis of categories

We first look at summary statistics for each variable across the categories.



Looking at the table below, we deduce that overall there is less variance in *first_place* than in the other two variables. This may be because for each test, the headlines usually belong to the same category despite being different (since they correspond to the same article). Also each test has 4 or 5 choices so the ratio comes out to be around 20% for all categories.

Variable	Mean	Standard Deviation	Highest Category	Highest Category Value	Lowest Category	Lowest Category Value
first_place	0.217	0.0213	BLACK VOICES	0.246	WELLNESS	0.171
winner	0.0483	0.0114	QUEER VOICES	0.0737	PARENTING	0.0321
click_rate	0.0166	0.00355	COMEDY	0.0231	GREEN	0.00977

We look at three pairs of variables which are often mis-categorised according to the confusion matrix: 'HEALTHY LIVING' and 'WELLNESS', 'ENTERTAINMENT' and 'COMEDY', and 'PARENTING' and 'PARENTS'. The confusion matrix for these pairs is as follows:

	Predicted Topic		
Actual Topic		HEALTHY LIVING	WELLNESS
	HEALTHY LIVING	65	65
	WELLNESS	20	147

	Predicted Topic		
Actual Topic		ENTERTAINMENT	COMEDY
	ENTERTAINMENT	183	8
	COMEDY	16	163

	Predicted Topic		
Actual Topic		PARENTING	PARENTS
	PARENTING	120	31
	PARENTS	70	87

The whole confusion matrix can be found in text_classification_confusion_matrix.ipynb. The results of the statistical tests are as follows:

Pairs of Topics	z-statistic for *winner*	z-statistic for *first_place*	t-statistic for *click_rate*
HEALTHY LIVING / WELLNESS	-0.628 Not significant	2.30 Significant	4.99 Significant
ENTERTAINMENT / COMEDY	-1.36 Not significant	-0.550 Not significant	-1.74 Not significant
PARENTING / PARENTS	-2.28 Significant	-1.32 Not significant	-4.22 Significant

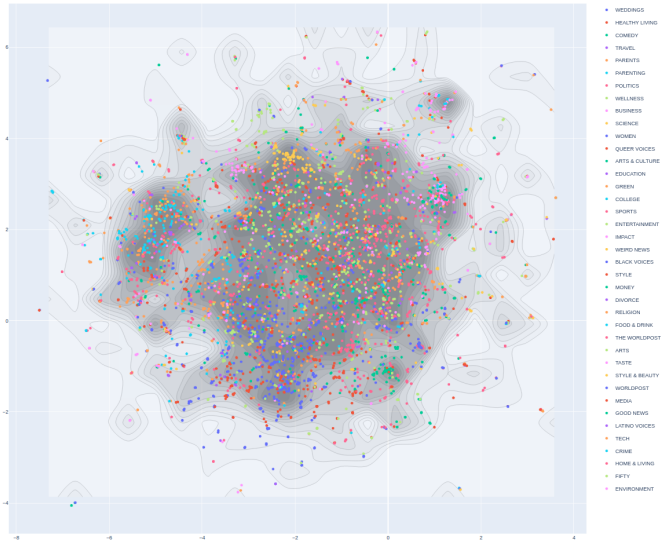
First consider the *winners* data. We see that the packages in the 'PARENTS' category are more likely to be chosen by editors than those in the 'PARENTING' category. A cursory glance at the headlines in each category suggests that the 'PARENTS' category refers to reflections on the parenting one received, while 'PARENTING' refers to actual parenting tips. Hence we could explain this difference in data by saying that the headlines in the 'PARENTS' category are relevant to a larger proportion of the population (not just new parents), hence editors are more inclined to choose these articles.

Now consider the *first_place* data. We would like to point out that headlines of the same article, despite being different, are likely to be grouped into the same category, hence the usefulness of the *first_place* variable grouped by category may not be as high as the other two variables. However, as stated in the Methodology section, re-framing a headline may affect the category, given these categories are so similar to each other. In this case, we observe that 'HEALTHY LIVING' headlines are more likely to be clicked on by users than 'WELLNESS' headlines. A cursory glance at the data suggests that 'HEALTHY LIVING' refers to mostly physical health, while 'WELLNESS' mostly refers to mental health. Again, the former is applicable to a larger proportion of the population, so may be clicked on by more users.

Finally, we look at the *click_rate* data. Both pairs of topics mentioned above yielded significant statistical test results. Note that the results were as expected in both topics, in that 'PARENTS' headlines received clicks more often than 'PARENTING' headlines, and 'HEALTHY LIVING' headlines received clicks more often than 'WELLNESS' headlines. Since winner and *first_place* have a strong positive relationship with the *click_rate* (as detailed in the Introduction section), this result is to be expected.

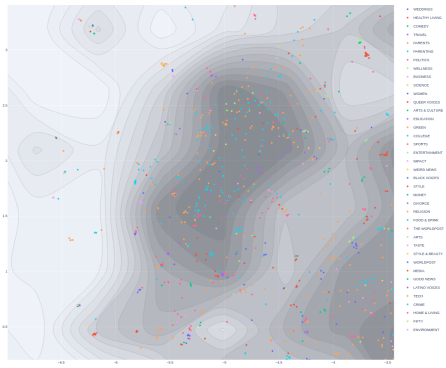
B. Visual analysis of categories

The visualisation of the UMAP plots showcasing the similarities among all categories can be found in the appendix section A. Here we present the UMAP plot for the pre-trained model **stsb-roberta-large**:

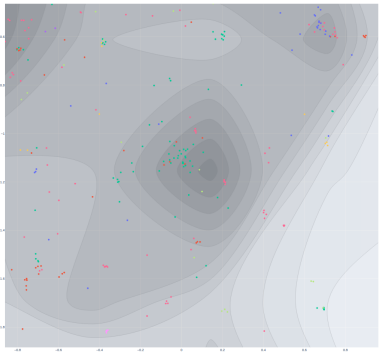


From this plot, we see the formation of certain clusters not necessarily pertaining to one category alone:

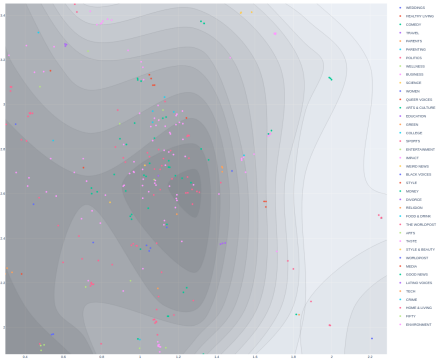
• Cluster 1: We have PARENTS, PARENTING, EDUCATION, POLITICS grouped:



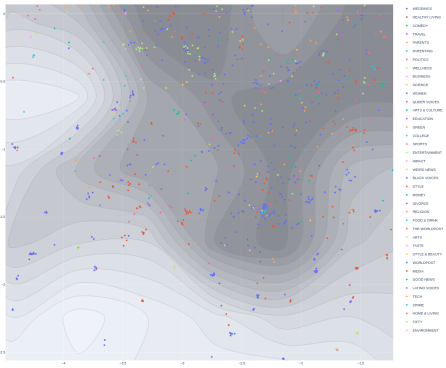
• Cluster 2: We have COMEDY, POLITICS, ENTERTAINMENT grouped:



• Cluster 3: We have BUSINESS, POLITICS, MONEY grouped:



• Cluster 4: We have WOMEN, QUEER VOICES, COLLEGE grouped:



This agrees with the confusion matrix and has given rise to some of the subsets of data studied above in the statistical analysis section. One of the shortcomings of this report was not being able to identify a suitable method to analyse the discrepancy or likeliness of some properties for the categories pertaining to different clusters, an example is how the curves for QUEER VOICES and WOMEN click data which appear to have similar content from the UMAP analysis.

C. Relationship To User Data and Significance

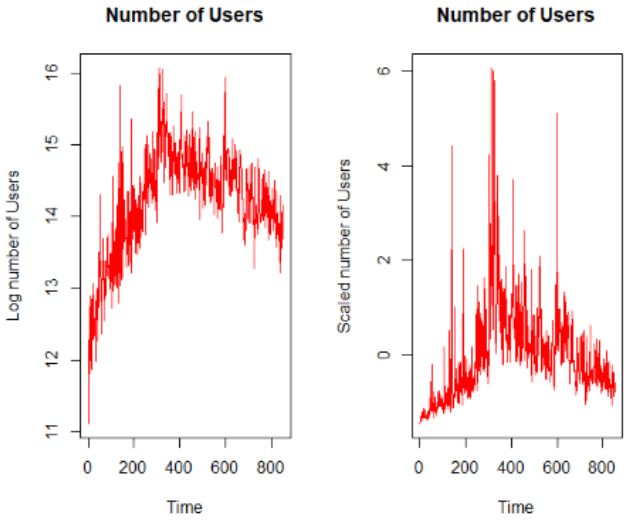
Finally, we wanted to see whether the conclusions drawn previously from the categorization of the headlines and the usage of A/B packages testing data carried any weight with regards to real-world performance. In order to analyze this, we examined the relationship between click rates of articles during the A/B testing and the number of users on the Upworthy website.

We let the max click rate of a day be the "winner" (posted) package with the high click rate amongst all other "winner" packages on that day. Now, we expected that on days with a more popular article in testing, these articles would be more popular in the real world on the Upworthy website and thus generate more users and higher traffic for the website in the time period that followed the testing. We will henceforth refer to the max click rate as the maximal clicks for a given day.

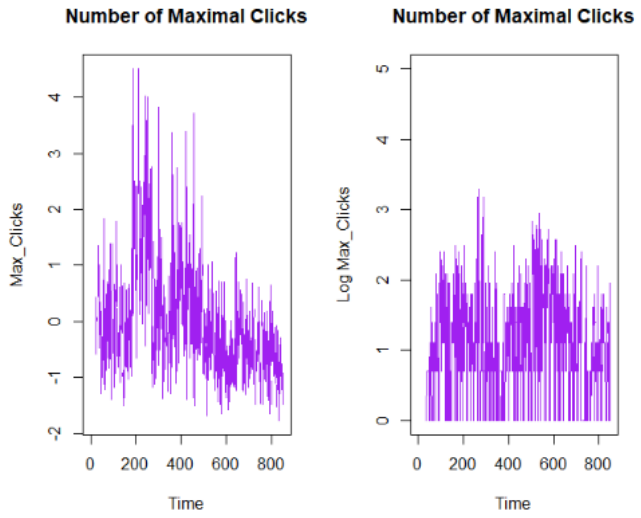
1. *Modelling this relationship*

From the above discussion we note that a time series model is necessary to understand the relationship between maximal clicks and users.

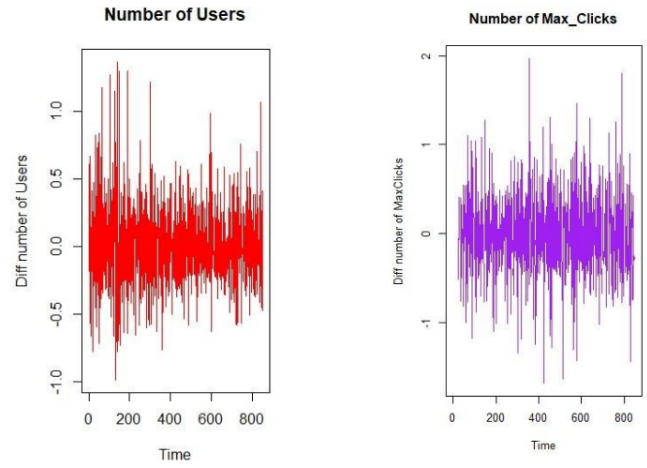
However, the "users" data is highly variable and non-stationary over time. Indeed, sometime after July 2013 there is an increase in the variation of the number of users until probably around August-September of 2014. To reduce this non-stationary time series of "users", we log transformed the number of user data. We then compared the log-transformed-user data to the scaled-user data visually:



There was a significant reduction in the variation over time using the log-transformation. Similarly, we looked at what happens to maximal clicks:



There was a significant reduction in the variation over time using the log-transformation. Observing the graphs, there are a number of shocks within the users data (possibly due to a viral article or two during that time period) creating stochastic trends that have permanent effects. To further achieve a stationary time series and reduce the effect of stochastic trends we used differencing.



We use the KPSS for stationary for both the users and max click time series, with P value threshold 0.1 (10%), which gave the following results:

- For Users - KPSS Level = 0.14925, Truncation lag parameter = 6, p-value > 0.1; so null hypothesis is accepted and hence stationary.
- For *Max_Clicks* = KPSS Level = 0.026118, Truncation lag parameter = 6, p-value > 0.1; so null hypothesis is accepted and hence stationary.

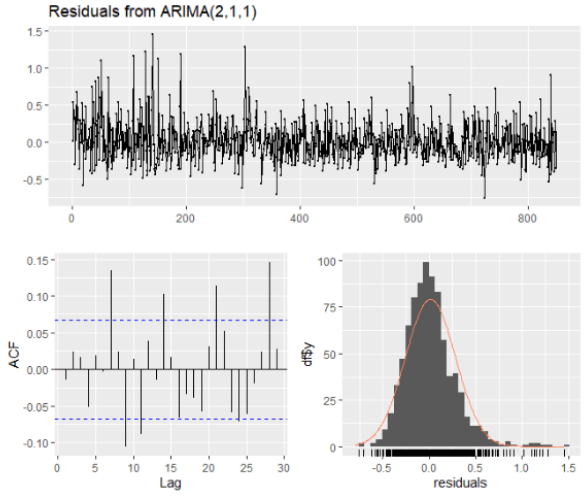
This suggests that differencing removes any changes in the data and makes these variables prewhitened. As such the following analyses using the test-related attribute of maximal clicks and the one outcome of users is expected to be free of correlated errors. This process also leaves out the possibility that a common temporal trend or pattern or other variable is a confounding explanation for any possible observed associations between one the test variable series and the outcome of user numbers.

2. *ARIMA modeling using the forecast package in R*

We then tried to fit complex ARIMA models to each of the 2 log transformed values of variables using the R forecast package that searches for a best fit ARIMA model using auto.arima functionality. We were able to fit an ARIMA model for the users and max clicks using the forecast package from R.

Note that the Log transformed User Model is autocorrelated.

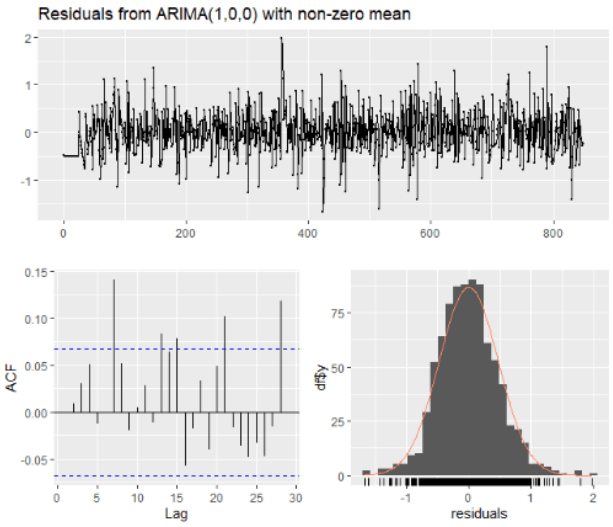
The residual analysis of the ARIMA log-transformed user model showed:



The Ljung-Box test of the residuals from ARIMA(2,1,1) for users showed Q* = 29.346, df = 7, p-value = 0.0001251, Model df: 3. Total lags used: 10.

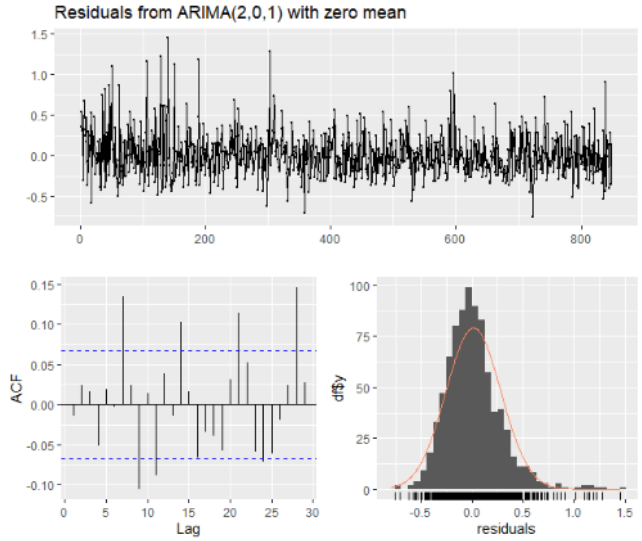
Note that the Log transformed *Max_Click* Model is autocorrelated.

The residual analysis of the ARIMA *max_click* model showed:



The Ljung-Box test of the residuals from ARIMA(1,0,0) with non-zero mean for max_clicks showed Q* = 22.868, df = 8, p-value = 0.003538, Model df: 2. Total lags used: 10. For both ARIMA results there was autocorrelation of the residuals. As such we then used the log transformed and differenced values to remove the autocorrelation. The Log transformed and differenced User Model is also autocorrelated We again were able to fit an ARIMA model for the log-transformed and differenced users and max clicks using the forecast package from R.

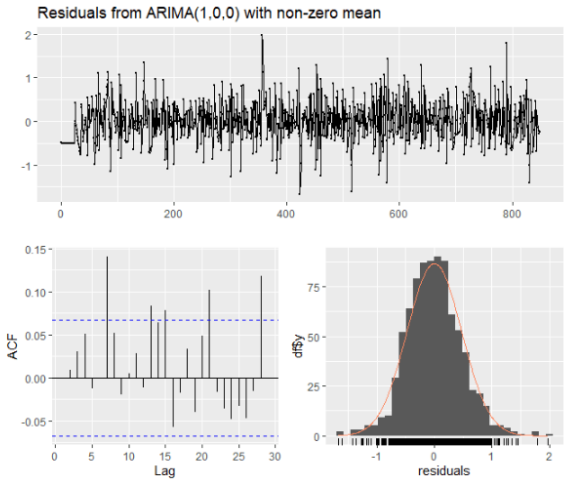
The residual analysis of the ARIMA model using the log-transformed and differenced users model showed:



The Ljung-Box test of the residuals from ARIMA(2,0,1) with zero mean for log-transformed and differenced users was Q* = 29.313, df = 7, p-value = 0.0001268, Model df: 3. Total lags used: 10.

Note that the Log transformed and differenced *Max_Click* Model is also autocorrelated.

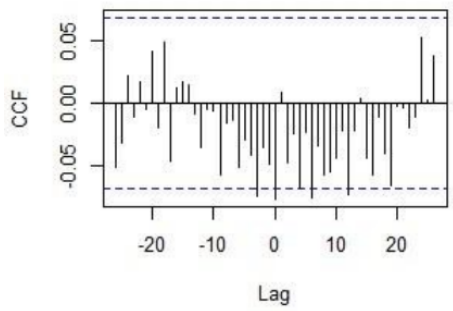
The residual analysis of the ARIMA model using the log-transformed and differenced *max_click* model showed:



The Ljung-Box test of the residuals from ARIMA(1,0,0) with non-zero mean for log-transformed and differenced *max_users* was Q* = 22.868, df = 8, p-value = 0.003538 Model df: 2. Total lags used: 10. As such using differencing did not seem to remove the autocorrelation of the residuals in these complex models.

3. *A simpler modelling of the complex relationship of users and max clicks*

A simpler autoregressive model that does estimation using conditional and exact maximum likelihoods or conditional least-squares can prewhiten the data and show meaningful correlations of users and *max_clicks*:



This graph shows that there is a spike in correlation just before 20 days between users and *max_clicks*. This shows that *max_clicks* precedes increases in user numbers by around 20 days. But there is also another interesting cyclical relationship which shows as the number of users increase, the max clicks are also increased 20 days after the increase in the total user numbers. This is a highly interesting circular relationship between max clicks and users, which leads to a highly multiplicative effect of the testing. Thus, the popularity of packages in the testing begins to show as an increase in popularity of the website (through new users) around just over 2 weeks after these popular packages are tested. Furthermore, we found that an increase in the users of the website is also correlated to a later increase in the popularity of the packages tested in general (examined through max clicks), which would indicate some sort of circular relationship between users and the popularity of articles in testing. However, overall we can see that success in testing can be used as a predictor to forecast an increase in the overall users of the website. Thus, analysis of headline category data using the A/B testing data does have implications for real-world popularity of packages and articles as the A/B testing data does have some correlation with real-world popularity of packages.

V. ROADBLOCKS ENCOUNTERED

A slight roadblock was the model used to classify the headlines. The accuracy was poor although grouping similar headlines together nonetheless. Instead more sophisticated training methods could've been used with more time in order to generate a more accurate representation of the topics of the headlines.

Some of the unsuccessful ideal were interpolating clicks and impressions data and finding their correlation using the two-sample Kolmogorov-Smirnov test. Also not being able to use tSNE to compare it with UMAP due to machine specification constraints.

VI. FURTHER INVESTIGATION

In the future we will investigate the relationship between properties of categories that belong to the same cluster in the UMAP plot, potentially providing reasons for why certain topics lie within the same cluster commonly. We will also investigate other datasets of articles like this, and potentially find more patterns between article topics and user and organisational behaviours.

It would be interesting to find relations between the topics and the excerpt and lede, as we had ignored this in our analysis for this paper as it was not used in any of the A/B tests. This would provide insight more about the articles themselves, the writing styles, and how these vary depending on the content of the article.

VII. NOTES

All authors worked equally towards the making of this study. Authors are not listed in any order in particular. This report was made over the course of a week - the duration of the competition. We have aimed to keep this document as true to the final submission as possible, we apologise for any formatting mistakes in this document.

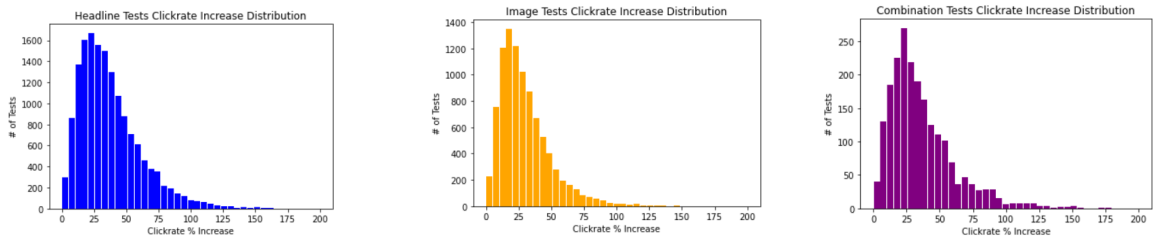
VIII. ACKNOWLEDGEMENTS

We would like to thank Citadel and Citadel Securities for organising the European Regional Datathon, and for awarding this work the 1^{st} place prize of the competition.

[1] G. Andrew and J. Gao, in *Proceedings of the 24th international conference on Machine learning* (2007) pp. 33–40.
[2] N. Reimers and I. Gurevych, arXiv preprint arXiv:1908.10084 (2019).
[3] R. Misra, 10.13140/RG.2.2.20331.18729 (2018).

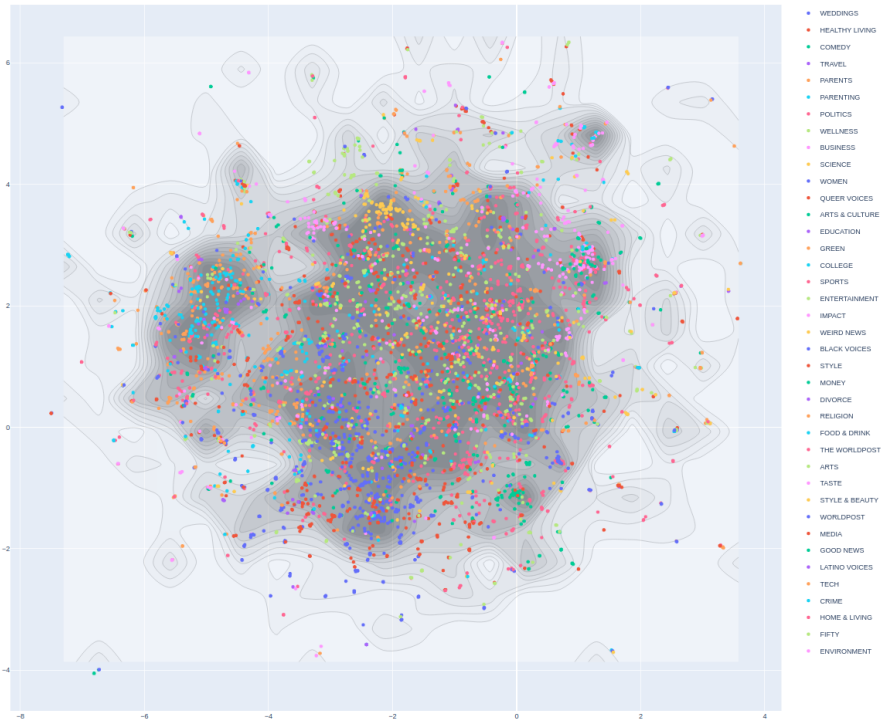
IX. APPENDIX

Histograms showing Impact of Headline vs Image Tests:

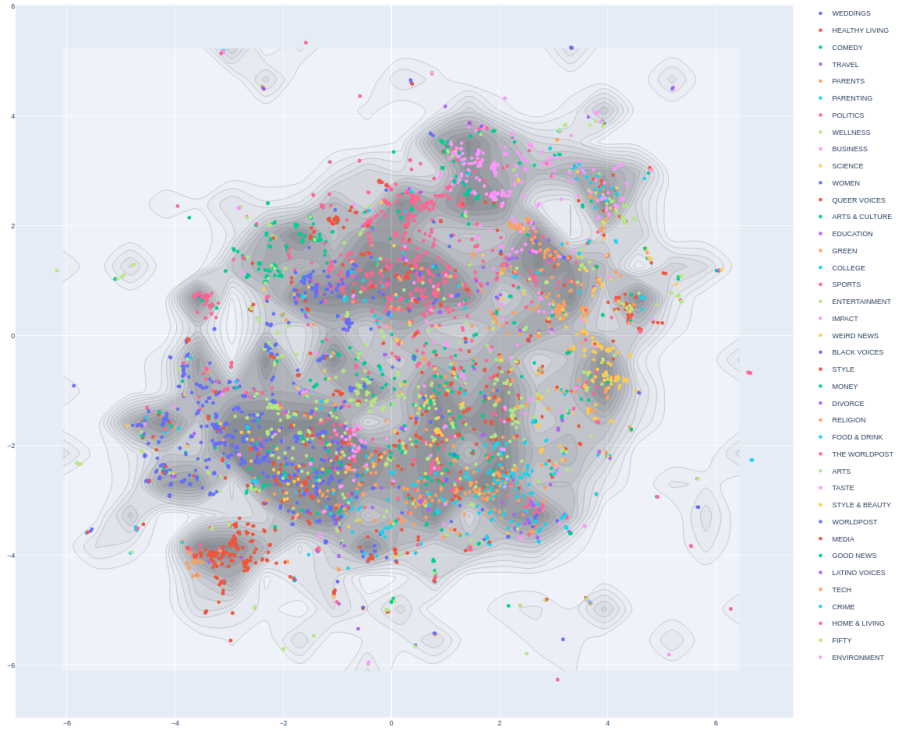


Here we have a bigger representation of the UMAP plots computed with the listed pretrained models:

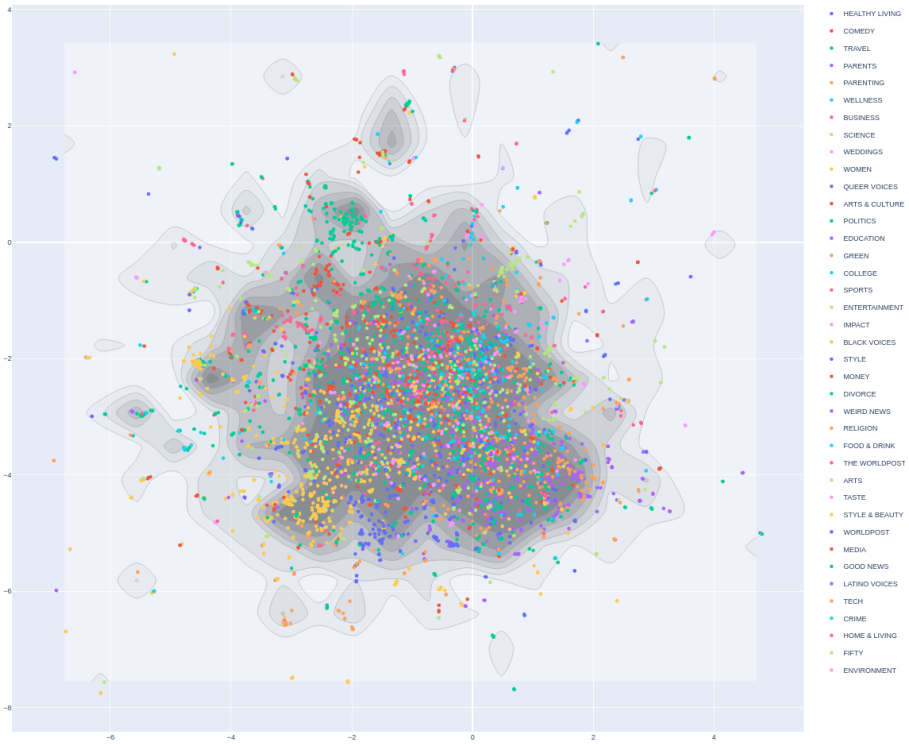
- **stsb-roberta-large**



- **all-mpnet-base-v2**



- **msmarco-distilroberta-base-v2**



- **paraphrase-distilroberta-base-v1**