Analysing the Upworthy Research Archive

Citadel European Datathon Team 3:
Benedict King, Michael Kerins,
Armando Schmid, Wilfried Bounsi

Contents

1 BACKGROUND AND MOTIVATION

As we enter into the second decade of the Social Media revolution, the term *clickbait* has become part of our common vocabulary. The term refers to a link to a web-page that combines often misleading headlines with enticing images to attract as many readers as possible. The idea is not a new one, and the concepts behind clickbait can be traced back as far as newspapers in the 1800s, up to the early world wide web of the late 1900s [1]. However, what was once just a slight irritation for an internet user is now an area of great debate, as in an always-connected society that thrives off of fast-access to information, the ability to influence what content people read offers a lot of power.

The news and entertainment website Upworthy was one of the first websites to be recognised as a large-scale clickbait producer, being the third most viewed publisher on Facebook by 2013 [2] despite smaller scale article production to competitors. Eventually, this emphasis on clickbait led to wide-spread criticism of Upworthy and other publishers like it, with many arguing [3] that the headlines are sensationalised to the point of being dishonest, and that at its extremes it can be seen as fraudulent due to the money the websites receive from advertisers for increased views. Therefore Upworthy changed tack in 2017, with a new vision to use its social reach to broadcast upbeat stories and drive social improvement campaigns to make a positive social impact [4].

With this in mind, Upworthy publicly released the Upworthy Research Archive, a data set of the results from 150,817 A/B tests conducted between 2013 and 2015, that can be used to investigate the factors that influence a viewer's decision to click on a hyperlink. The aim of this report is to look in detail at what makes a hyperlink compelling, whether headlines alone can be used to predict the success of a package, and whether we can use hyperlink click data to gain any insight about the psychology of the users that click on them.

2 PROBLEM STATEMENT

When it comes to deciding whether to click on a hyperlink the two main factors are the image used as the link's thumbnail and the content of the headline. Therefore, due to less availability of image data, we focus in this report on the content of the headlines, and aim to answer the following questions:

1. Can the headlines be used directly to predict the click-through-rate (CTR) of a package and would including the knowledge of the publish date improve our prediction ?
2. Which of the headlines' characteristics have the biggest part to play in a users decision to click on a hyperlink?
3. Can the time-dynamics of the picked headlines' sentiments give insight into how user's attitudes change over time?

Question 1 seeks to find a sentence embedding for each headline that can allow for separation between picked and not picked packages based on headline's words alone. This is fundamentally a natural language processing (NLP) problem, and so we use machine learning techniques related to text classification to approach it. There would be many uses for knowing what makes an alluring headline, for example for advertising companies to learn what attracts viewers, and for social science researchers to help understand the biases and behaviours of user groups. Question 2 then aims to extend this analysis into looking at specific characteristics of the headlines, such as sentiment, emotion, and name entities, and analyses whether these more fine-grain approaches can improve from the results of Question 1.

So far the questions proposed treat the data as being static, but there are a multitude of reasons why this may not be the case. Over the two year period spanned by the tests social behaviours and public opinion can vary dramatically, for example online media sentiment around politicians can vary greatly even within timescales of months [5]. Therefore Question 3 explores whether the dynamics of the inferred sentiments and click-through-rates of headlines shed light onto how user-group attitudes change with time.

3 EXECUTIVE SUMMARY

The purpose of this section is to be a high-level overview of the results of the analyses undertaken in this investigation. The following sections will then take a more technical approach and lay out in detail the steps taken to arrive at our conclusions.

In answer to out first problem statement, we attempted to find vector representations of the raw headline texts in either low or high dimensions that would allow a classification algorithm to separate between picked and not picked headlines. By looking at the absolute differences in click-through-rates between the successful and unsuccessful packages it was clear that there is not much between their CTRs (i.e. a picked headlines' CTR is similar to a non-picked ones' CTR, even for the same test), and therefore it is expected that there would not be a simple low-dimensional representation that can separate them. This suspicion was confirmed by trying to find 2-dimensional non-linear mappings that separate the classes using T-distributed Stochastic Neighbour Embedding (t-SNE), and finding that over 1000 iterations there was no solution that achieved this. Even when using higher dimension (e.g. 384 dimension) representations with the Bidirectional Encoder Representations from Transformers (BERT) algorithm, a representation could not be found that presented a suitable classification using raw headlines alone.

Including details about date of testing and image ids and using a feed-forward neural network in Section 7 showed a slight improvement of around 10% (on a held-out validation set of packages), but even including more information proved to not provide a satisfactory prediction accuracy. Therefore, we moved onto looking at more specific features of the headlines, and what insight they give about what components make a headline more likely to be clicked on. The analysis in Section 6 explored three avenues of increasing granularity: firstly sentiment of negative, neutral or positive; then emotion; and finally which named entities the headlines include.

For the analysis of sentiment, it was found that, in general, the picked headlines tended to have more negative sentiment than those that were not picked

(which tended to have more neutral sentiment). There was found to be statistical significance between the means of the sentiment probabilities (i.e. the probabilities of a headline being negative, neutral or positive) of the picked and non-picked headlines. However, as almost 90% of the headlines were classified as negative sentiment, there is not much spread over the sentiment classes, and therefore even this significance does not say a lot about why some headlines are more popular. This is partly fixed by looking at emotion, as headlines have far greater spread over emotions than sentiments. This analysis demonstrated that picked headlines tend to have a noticeably greater element of surprise and disgust than non-picked ones, and hints at the well-believed notion that the most successful clickbait headlines are commonly over-sensationalised (see Section 1).

These results are important for understanding why online publishers often create headlines to be exaggerated or negative, as from a basis of purely wanting to maximise hyperlink clicks, they are the most popular with the users. This could also be an interesting avenue for social science research, because whilst this analysis gives insight into which sentiments users tend to prefer, it does not elucidate *why* they prefer them.

The analysis on names entities did not explain much about why some headlines were picked over others, as the named entity contents of both the picked and not picked packages were quite similar throughout. However, this being said, it did reveal the common topics of headlines in general. For example, almost a quarter of the headlines included a person's name, and around 15% a name of an organisation. An explanation for this could be that the majority of people will tend to read content that is non-technical and easily readable, and articles on specific people and organisations tend to fit this criteria. For example, there will not be many headlines on topics containing a lot of jargon or technicalities, because these articles would not be accessible to a large proportion of the population.

A second crucial aspect of the data is incorporated into the second half of our analysis; Time.

Tests in the Upworthy research archive span a period of over 2 years and 6 months. The wide scope and unprecedented volume of testing may allow us to uncover dynamic structure to the data set, out of reach of a static analysis. We seek answers to two main questions in this section:

1. How has CTR changed over time?
2. How has sentiment changed over time?

To begin tackling the first question, the package CTR data was transformed such that it was grouped by week and aggregated by mean. This forms the time series $\{Y_t\}$ used for subsequent analysis. An ARIMA(p, d, q) model is then fit to the series. The ARIMA framework is a simple way to capture Auto-Regressive, Integrated, and Moving-average components in a time series.

An ARIMA(0, 1, 1) is found to offer the best fit. All but one modelling assumption is satisfied; the time series exhibits heteroskedasticity. That is, the volatility of the model is time contingent. This is an interesting discovery in the context of CTR. A nice visualisation of such behaviour is a plot of the squared model residuals (Figure 1). We clearly see transient bursts in the residuals, interspersed between periods of relative stability.

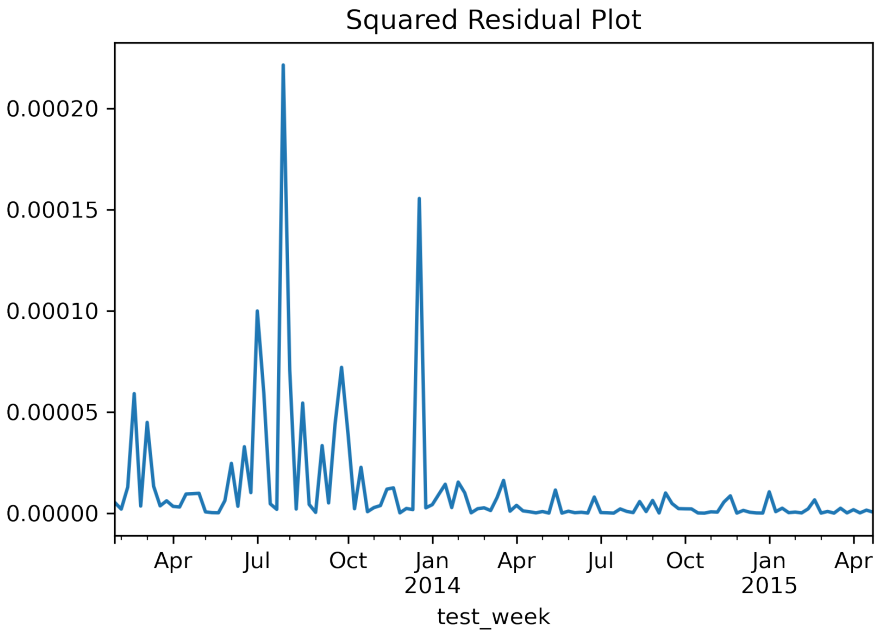

Figure 1: Volatility Clustering of Residuals

We often see this type of behaviour in financial time series, so it is very interesting that CTR also exhibits it. Although the cause of this 'volatility clustering' beyond the scope of this report, it is conjectured that viral stories may be a cause. Intuitively, this is consistent to the very sudden appearance of volatility, and it's short duration. We captured this behavior by redefining the model as $\{Y_t\} \sim$ ARIMA(0, 1, 1) + GARCH(p, q). The second component here (Generalised Auto-Regressive Conditional Heteroskedastic) models the conditional volatility in the residuals of the ARIMA(1,1). Simply put, this model outputs a volatility multiplied by a white noise at each time t, which is added to the ARIMA(0, 1, 1) model to get our full model. This volatility term is a function of two things: past values outputted, and past values of volatility. A GARCH(1,1) is found to be the best fit to the time series. Adding the GARCH component successfully removed residual heteroskedasticity, leaving a white noise. All remaining modelling assumptions are satisfied, and parameters are estimated. These are found to be highly significant, supporting our model choice. The full model is as follows:

$$Y_t = -0.000137 + Y_{t-1} + \epsilon_t + 0.486\epsilon_{t-1} + \eta_t$$

$$\eta_t = \delta_{t|t-1}\epsilon_t$$

$$\delta_{t|t-1} = 0.0000000171 + 0.13\eta^2_{t-1} + 0.866\delta^2_{t-1|t-2}$$

Forecasting using the model. We found that the times series is fitted by an integrated model of order one (d=1). What does this mean for the future of CTR? Non-stationary time series have autocovariances which depend on time. Because of this, it would be increasingly difficult to predict the average CTR the farther we forecast into the future. Of course, time series modelling is not typically powerful or even applicable on longer timescales, so this is not surprising. The moving average component (q=1) here essentially smooths this Integrated component (which by itself is a random walk). It offers further Information to a one step forecast, but none beyond that. If we observe high volatility at the current time step, we expect to see a drop back to normal levels in the near future. (Note this would not be the case if we had a higher order GARCH model).

To make a statement about the change in sentiment over time, we use the sentiment scores computed in Section 6 as a proxy. For each observation, in our

case each unique combination of image ID and test ID with at least two different headlines, no duplicate headlines and more than 0 clicks in the whole test, we get three columns of sentiment scores: negative, neutral and positive. All 3 columns together add up to 1. We also want to know how sentiment changed in the relevant packages, i.e., the picked packages. Therefore, for each sentiment, we create another column with a binary variable indicating whether the observed package was mainly assigned to the respective sentiment category and picked.

Similar to the question about the change in CTR over time, the packages are also grouped by week to perform further analysis of sentiment over time. Headlines with negative sentiment are picked more often than others, followed by headlines with neutral sentiment. All three time series are stationary, but the most interesting to model is the positive sentiment time series. It turns out that an ARMA(1,2) model describes its behaviour the best. In contrast to CTR over time, this time series is stationary with homoskedastic variance. Therefore, a simple ARMA model without differentiating or seasonal component is sufficient to model the true underlying data generating process. The assumptions needed for computing the optimal ARMA model using the Akaike Information Criterion applied in the Box-Jenkins method are satisfied as indicated by the statistical tests taken.

To conclude, it can be stated with a high certainty, that the sentiment of the most clicked packages over time remain constant. The negative and neutral time series both resemble stationary process without any trend or seasonality, but with some fluctuations over time, while the positive sentiment is best modelled with an ARMA(1,2) process. Given the structure of the A/B tests in the data set and the performed statistical tests, we can conclude that although negative sentiment in the headlines leads to higher CTR, the proportion of negative headlines that outperformed neutral or positive headlines in causal statistical tests, is independent of time.

4 EXPLORATORY ANALYSIS

As discussed in the above section, this report will focus in particular on the headlines used for Upworthy's data collection, and what makes a headline more likely to be chosen. We define a headline as being *picked* if it has the most clicks of all the packages in a particular test, when all other variables are held constant. This means that we neglect the headlines from tests where headlines are held constant and the image IDs or excerpts, etcetera, are varied, as in these cases we cannot be sure that it is the headlines that influence a user to click on a particular link, if other aspects of the link are also changing. The other important point to take into effect is that the absolute numbers of clicks on the links are not transferable between different tests. This means that we consistently normalise the number of clicks by the number of people who have seen the link, called the *impressions*, to negate any bias from some tests being shown to a larger audience than others. We will refer to this normalised term as the *click-through-rate* (CTR).

This section aims to provide a basic analysis of the headlines present in the provided data set, and how they relate to the CTR.

DATA CLEANING

In order to prepare the data for exploratory data analysis (EDA), NLP, as well as time series analysis, it must first be cleaned and transformed into the correct format. Of the given columns in the package dataset, only impressions and clicks are numeric, both represented as integer values. The columns created at and test week are transformed to datetime formats to correctly represent the logical value of the entries. The values of winner and first place are boolean values. The first step is to create a new column called click-through-rate (CTR) where clicks are divided by impressions to quantitatively compare different packages on a standardized basis. The first place column is not a one-to-one mapping of which package had the highest click-through-rate within a test.Therefore, the click-through rate, which is a quantitative metric, well-defined and measurable is considered for further analysis. Throughout the rest

of this report, any reference to *picked* packages refer to having the highest click-through rate. The feature winner is not considered appropriate for analysis because it is unclear what constitutes a winner and the number of winners, 7664, is significantly less than the number of tests.

In order to perform correct statistical tests for causality, it must be ensured that the desired properties can be measured and that a correct cause-effect relationship can be established. In particular, any confounding effects must be mitigated. Since no images are available for the dataset and the focus is on the NLP of the headlines, it must be ensured that the headlines considered for the analysis each refer to the same image. If the only difference between two packets is the image, no causal relationship can be inferred with respect to the headline and thus these packets must be excluded from the analysis. In addition, tests with 0 clicks across all packets are also not considered, as these would not provide any insights and would lead to a bias in the analysis. It is important to note that the different packages must be tested within the same week, otherwise no causal conclusion can be drawn and a time series analysis is not possible. All cases that differ only in terms of slug, excerpt, lede, share text, and share image must also be discarded, as these were not displayed before clicking on the article and are therefore not relevant for the click decision. The few test ids that contained duplicate packages were also deleted. This guarantees that for every test id, image id combination there are at least two distinct headlines with a positive number of clicks. Only after this data cleaning is it possible to conclude causal behavior and not just correlation. 72135 packages and 16519 unique test ids are left in the pre-processed data set.

IT'S A THIN LINE BETWEEN PICKED OR NOT

It is easy to think that if a particular package is picked then it must have had a lot more attention than the others in the test case and thus a much higher click-through-rate. However, it is clear from Figure 2 that there is not much of a difference in CTRs between the picked and not picked classes, with many of the non-picked packages having CTRs higher than the picked counterparts (this is possible because picking is done within each test, so the picked package from

one test case can have a smaller CTR than a not-picked package from another, for example, if the image of the first test case was very unappealing).

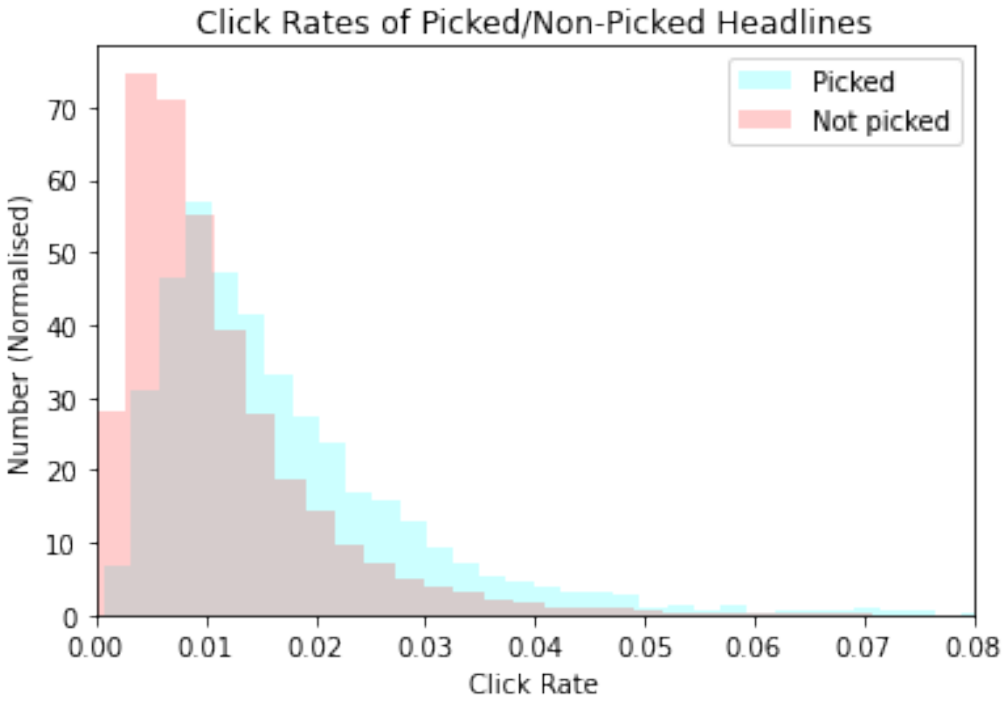

Figure 2: A histogram of the normalised number of headlines in each click-rate category for the picked and not picked classes, showing the small differences in click rate that lead to a package being picked or not.

As a result of this similarity in CTR between the classes, we expect that the words in the headlines that were picked will not be all that different to the words in those that were not, and therefore that any sentence embedding that can separate the classes will be subtle, so we should anticipate having to use a highly non-linear embedding and classification algorithm. This is discussed in greater detail in Section 5.

5 SENTENCE EMBEDDINGS FOR SEPARATING CLASSES

In order to gauge whether a headline being picked can be predicted purely from its contents, we explored how the high-dimensional feature vector representations of the headlines can be non-linearly separated. If, in any dimension $\mathcal{D} \leq \mathcal{K}$ (where $\mathcal{K}$ is the dimension of the sparse feature vectors), there exists a linear or non-linear mapping between the headline feature vectors and whether each headline is picked, then we can be sure that it is possible to accurately predict the effectiveness of the headline based on the headline text alone.

Figure 3: A plot of the 2 dimensional t-SNE representing a non-linear mapping from the higher dimensional feature vectors to a 2D representation. Even with a perplexity of 500 and 1000 iterations it is clear that t-SNE did not find a way to separate the picked from the non-picked packages. The sparse feature vectors were passed into Truncated Singular Value Decomposition prior to the t-SNE algorithm to initially reduce dimensions for computational speed.

As discussed in Section 4, the difference in CTRs between picked and non-picked headlines is a small one, and so we expect that if a mapping exists, it will not produce clearly separated clusters. This is further shown in Figure 3, where the T-distributed Stochastic Neighbour Embedding (t-SNE) algorithm as described in [6] has been used to try to separate headline feature vectors into the two classes in 2 dimensions. Despite a high perplexity to account for the small number of classes, and iterating over 1000 steps in an attempt to reach a stable

result, t-SNE fails to pull-apart the picked and non-picked classes, demonstrating that even highly non-linear mappings are likely insufficient to separate between picked and not in 2 dimensions.

Therefore if there does exist a sentence embedding that allows for distinguishability between the two classes then it would have to belong in a higher dimensional space. In order to search for this higher dimension representation, 384 dimension sentence embedding vectors were extracted from the output layer of a state-of-the-art pre-trained language classification model, called a BERT (Bidirectional Encoder Representations from Transformations) model (see Section 6 for further details on this model).

Using these high-dimension feature vectors, a non-linear Support Vector Machine (SVM) was trained to try to separate vectors based on whether they were picked. In order to avoid availability biases, the number of packages for training was reduced so that the number that were picked is equal to the number not picked (i.e. equal *a priori* probabilities between the classes). The training set then used 75% of the reduced packages, and 25% was held out for testing. Again, if the SVM is able to accurately distinguish between the cases then it would show that there exists an accurate non-linear mapping in 384 dimensional space, and therefore that the success of packages can be directly classified from their headlines. However, this was not the case, and the test-accuracy was 50.2%, showing little improvement over a coin-toss like decision (i.e. not based on any knowledge of the data).

Therefore, in answer to Question 1 of our Problem Statement, it seems there is not a simple method to map between the raw headline text and the package having the highest CTR of it's test batch, as was expected by looking at the small discrepancies between CTRs in picked and non-picked packages in Section 4. However, this is not to say that the headlines do not provide insight into click-through-rates, only that we have to ask more specific questions about the headlines in order to see how they effect the likeliness of being picked. This is the basis of the following section (Section 6), which attempts to give an answer to the second question of our problem statement - how sentiments can be inferred from the headlines, and how they effect the packages CTRs.

6 THE INFLUENCE OF HEADLINE SENTIMENT

From visual inspection of the headlines there seems to be a trend of the headlines being picked having negative sentiment. However, even if this is true, it could also be a consequence of there being more negative headlines in general, and so to gauge whether there is any significance we analysed the sentiments of all the distinct headlines and compared between the picked and non-picked classes. In order to do this we required a means of judging sentiment from raw text using a language classification model. Due to their state-of-the-art, we decided to run the headline texts through a pre-trained Bidirectional Encoder Representations from Transformers (BERT) model [7] trained on the TASS corpus of over 850 million bodies of text, taken largely from Twitter tweets between 2012 and 2019 with sentiment and emotion labels. This therefore introduces a set of assumptions on the form of the headlines if we are to treat the classifications from the model as accurate:

1. We assume that the vocabulary used in the headlines is included in the vocabulary of the training corpus. Due to the size of the training corpus being a lot larger than our headline data (over 10,000 times larger), this is a reasonable assumption.

2. In addition, we assume that the vocabulary of our headlines is used in a similar context to the vocabulary in the training corpus. As the training corpus is comprised of tweets, it is unlikely that any significant part of the training data will have a different context to hyperlink texts, i.e. they are both mainly in colloquial English, and unlikely to contain any significant amount of jargon or words used in unusual contexts. Therefore, we feel this is also a safe assumption to take.

With these assumptions we used the BERT model to create word embeddings that then are passed to a simple logistic regression model to assign three probabilities to each headline, representing the probabilities of the headline being negative, neutral and positive. The sentiment of that headline was then simply taken as the the maximum of the three sentiment probabilities.

Sentiment	Headline Picked	Not Picked
Negative	44.0%	39.8%
Neutral	39.8%	42.8%
Positive	16.2%	17.5%

Table 1: The percentages of sentiments for headlines in picked and not picked classes. In the picked headlines, the most common assigned sentiment was negative, in contrast to the non-picked class, where the most common assignment was a neutral sentiment. In general, there were far more negative and neutral sentiments than positive ones.

Table 1 shows the percentage sentiment compositions of the picked and non-picked classes. Whilst there looks to be a trend in the results, it is difficult to see if there is significance when just using the percentages of headlines with each sentiment. Therefore, using the raw probability scores associated with negative, neutral, and positive sentiment for each of the picked and non-picked headlines, we were able to compare the means of the picked and non-picked classes for each sentiment, and use a Student T-test to assess significance. As the entered values are probabilities (and therefore between 0 and 1), we first mapped the probabilities onto the entire real axis using the arcsin equation below (inspired by [8]), in order to justify Gaussian distributed samples. With this method, the differences were found to be significant for all cases with a 5% significance threshold, and hence we conclude that there is a difference in sentiment between picked and non-picked classes.

$$y = \arcsin\left(2x - 1\right)$$

INCLUDING EMOTIONS

The results of sentiment analysis suggest that a picked headline is more likely to have a negative sentiment than their non-picked counterparts, which tend to have more neutral sentiments. However, it is also clear that the majority (83.9%) of the headlines are of negative sentiment, so this result does not give

particular insight into what factors make a headline more likely to be picked. Therefore our next test was to further decompose the headlines into emotion categories, again using a state-of-the-art BERT model trained on the TASS corpus, but now with emotion labels. The results of this analysis are shown in Figure 4.

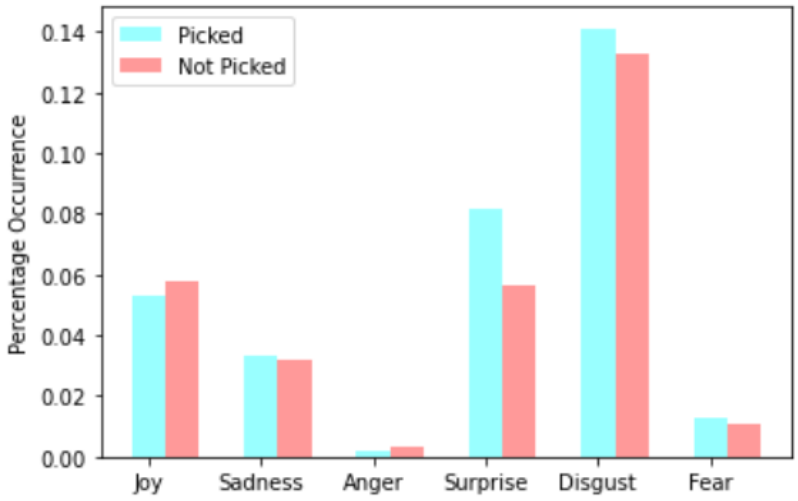

Figure 4: Bar chart showing what percentage of picked and non-picked headlines can be associated with certain emotions. The percentages are normalised to the number of packages in each picked/non-picked category in order to make the results comparable. Probabilities do sum to 1, except the 'other' category was omitted here. Emotions provide more insight into why some headlines were picked, as the headlines have far more spread over emotions than sentiment.

These results show that there is greater discrepancy between the inferred emotions of picked and non-picked headlines than between their sentiments alone. In particular, a picked headline is 44% more likely to be classed as representing surprise, suggesting an element of sensationalism in their wording. This is inline with the criticisms that Upworthy received during the tests' time-period (see Section 1), which claimed that the headlines were often exaggerated almost to the point of being dishonest. However, it also works to justify Upworthy's decision to post sensationalised headlines as simply a consequence of the users' preferences: the CTR data shows that including words that portray surprise and other strong emotions such as disgust make the headlines more likely to be clicked, and therefore, from a mentality of purely wanting to maximise user-clicks, the decision to produce more headlines of this type seems sensible.

Whilst sentiment and emotion form a large part of a headline, there are still other factors that we would expect to have significant influence on its effectiveness. For example, users may be more likely to click on a hyperlink if the headline contains content which they are familiar with, such as the name of a place or person, etcetera. Therefore, we took this analysis further and analysed the success of a headline as a consequence of the named entities it contained.

INCLUDING NAMED ENTITIES

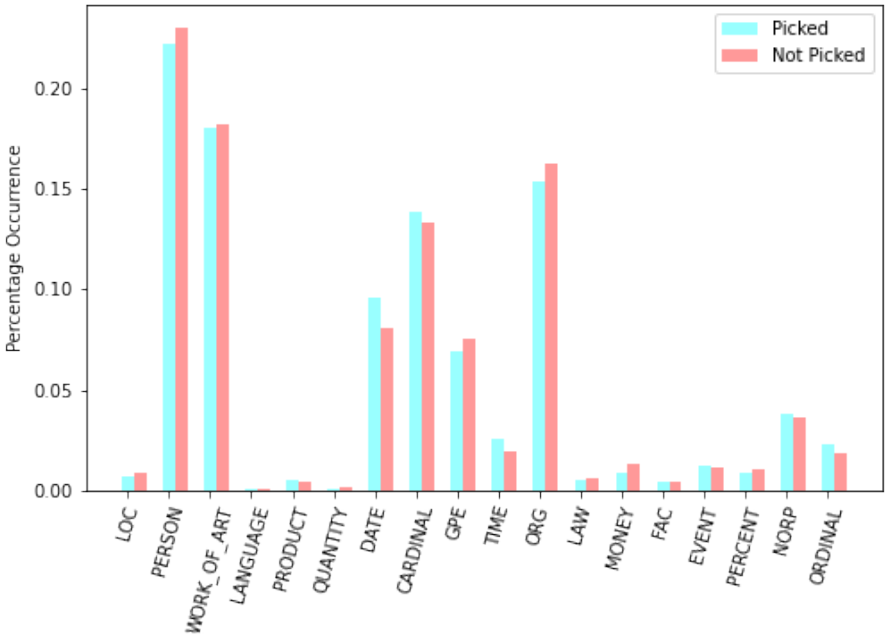

Figure 5: Bar chart showing what percentage of picked and non-picked headlines mentioned certain named entities. The percentages are normalised to the number of packages in each picked/non-picked category in order to make the results comparable.

Figure 5 shows percentage occurrence of various named entities in the different classes of packages Picked and Not-Picked. We can see that histograms of both classes are very similar, thus it's not obvious which types of named entities augment the chances a package has to being picked. This similarity can be explained by the fact that the headlines where mainly about the same underlying articles, therefore, there was probably little room for the usage of different named entities across different packages in the same test, since they have to remain consistent with the topic of the article.

That being said, they might not be blatant, but we can still observe some patterns emerging patterns in the figures. For example, picked headlines do tend to have more mention of a numbers (cardinals and ordinals) than non-picked ones, this aligns with results of the studies mentioned in blog articles [**carere**] & [9]. Additionally, and perhaps more surprisingly, picked headlines are less likely to mention persons or organizations.

7 CLICK THROUGH RATE PREDICTION

In this section, we are going to discuss the predictability of the Click Through Rate (CTR) of a package, given a minimal set of features or predictors. We first take an optimistic approach, restricting ourselves to a single predictor, the BERT-based headline embeddings. In the light of the limits of this approach, we will extend our space of predictors, via an analysis of the learnability of the ctr and other features such as the publication week, and fine-grained headline properties such as length, sentiment, mentions of named entities (e.g. numbers, organizations, locations, persons etc..). This will lead us to select among the latter candidate predictors, those having the most influence over the CTR. The resulting trained model can serve to evaluate packages before deployment, which, of course, can turn out to be very interesting from a commercial point of view.

MODEL ARCHITECTURE AND EXPERIMENTAL SETTINGS

We have designed a MLP with two hidden layers of size 64 and 32 respectively, and opted for $ReLu$ as our activation function. As this is a regression problem, and we have no prior on the sparsity of the weights, we have decided to choose the Mean Squared Error as our loss function, and to go with the Adam Optimizer, as it shows best results in practice [10]. Additionally, we have performed a standard normalization of the CTR in order to have nicely scaled loss values. Finally, We have split our full dataset into 80% of training data and 20% of validation data. The training is done on 30 Epochs.

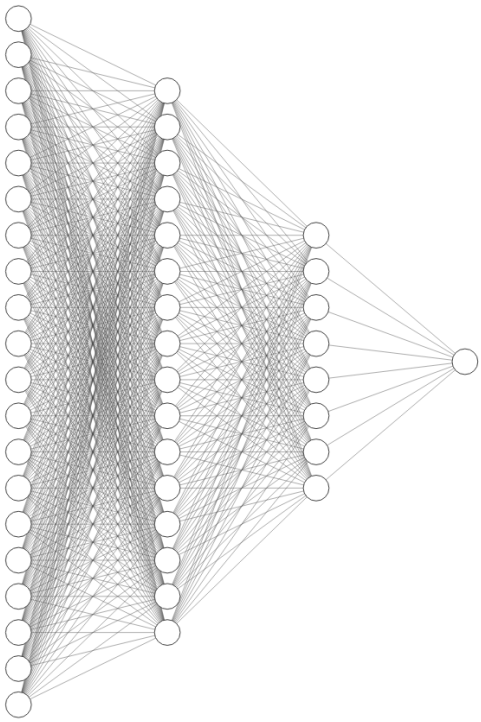

Figure 6: MLP For CTR Prediction

CTR PREDICTION (TAKE I)

Figure 7 shows the learning curves of our model when we pass it only the BERT encodings of the headlines. We note that although it is decreasing, the loss on the validation data remains quite significant compared to that on the training data, with it's best value being 0.63, which suggests that for this simple model, the headline embedding alone is not sufficient to predict the CTR of a package.

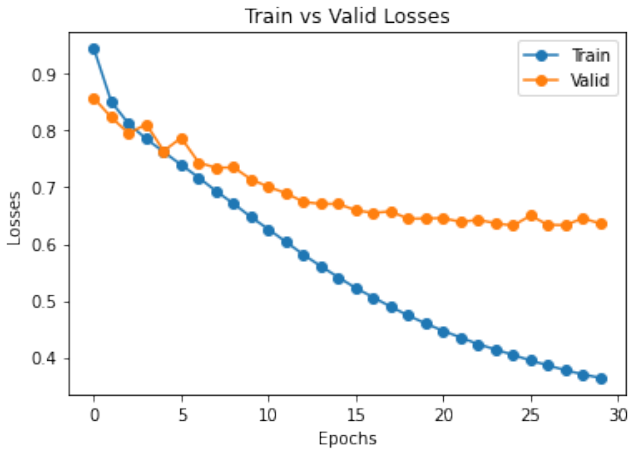

Figure 7: Learning Curves CTR prediction (Take I)

INFLUENCE OF OTHER FEATURES

As we will see in section 9, time has a determinant influence over CTR, so we already know it's a good CTR predictor candidate. Now the question that remains is what other features can help us improve our CTR prediction? Since the only other data we can use is the headline (as previously argued), we've decided to extract as much information from as we can, and pass them explicitly to our neural network. At first glance, one might think that this is redundant because we are already passing the headline embedding to the neural network in 7. However, since an embedding is a projection in a small dimensional space, in other words, a kind of compression, it is reasonable that it entails some loss of information, however small it may be. Therefore the explicit addition of headline extracted information can potentially improve the results, although not necessarily by much, as we will see.

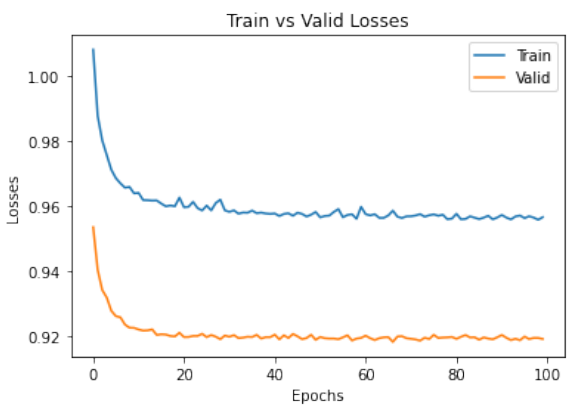

Figure 8: Time & Headline Length

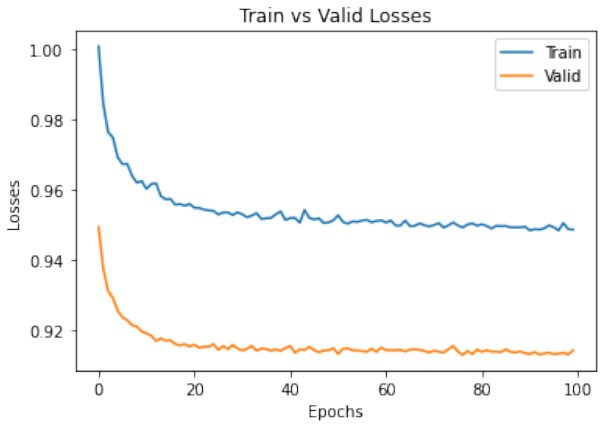

Figure 9: Time & Sentiment Scores

Figure 8, 9 show that the headline's length and sentiment scores are not that useful when it comes to determining the CTR. As we can see, the training plateau is reached relatively early on, and the loss remains quite high on the training set. On the other hand, 10 confirms the relevance of named entities occurrences as predictor for of the CTR.

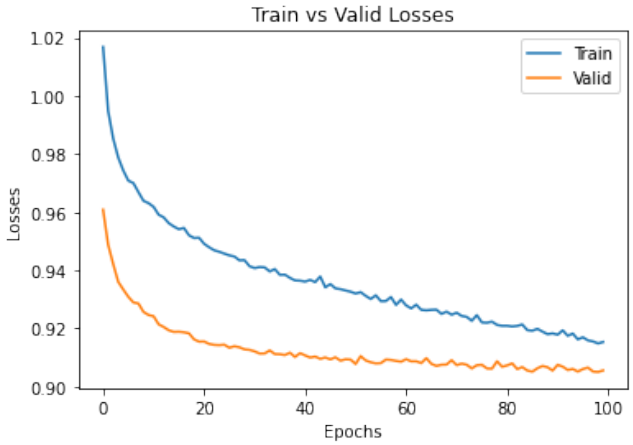

Figure 10: Time & Named Entities Occurences

CTR PREDICTION (TAKE II)

Figure 11 shows the learning curves of our model when we additionally pass it publication time (as the month number in the year and the week number in the month), the headline's length, inferred sentiment scores, and types of named entities mentioned if any. We note that our learning curves are significantly smoother, and although the learning is clearly still sub-optimal, the best loss value on the validation set is now 0.56, that is a about a 10% improvement over the previous one.

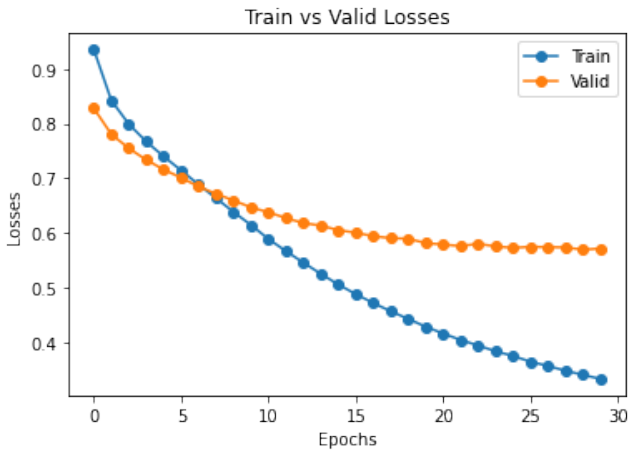

Figure 11: Learning Curves CTR prediction (Take II)

8 WINNING PREDICTION

In this section we are interested in the automatic testing of packages. That is, we want to answer the following question: given a set of packages, can we automatically predict the one that will be picked?

Note that this is slightly different problem than before, Firstly, because it is adversarial in nature, our input here is a set of packages (an entire test), instead of just one, as it was the case in section 7. Secondly, Our output here is a vector representing the probabilities that each package wins. Similarly as before, we first take an optimistic approach, restricting ourselves to a single predictor, the BERT-based headline embeddings. In the light of the limits of this approach, we extend our space of predictors, by leveraging the headline properties introduced above.

MODEL ARCHITECTURE AND EXPERIMENTAL SETTINGS

For this task we've designed an MLP similar to the previous one, the differences being that the input vector is obtained by stacking vector representation of each package in the test, with possibly padding. We've designed our model to handle up to 17 packages in a test, which corresponds to what we've observed in our dataset. The task is therefore modeled as a classification problem with 17 classes. We naturally choose the Cross-Entropy Loss as our loss function, and once again go with the Adam Optimizer, with the same motivation as before.

We split our full dataset into 80% of training data and 20% of validation data. The training is done on 15 Epochs.

WINNING PREDICTION RESULTS

Figures 13, 14 show the learning curves of our model for both loss and accuracy when we pass it the embeddings of the headline, and other headline derived features aforementioned in section 7. We note that the accuracy remains quite

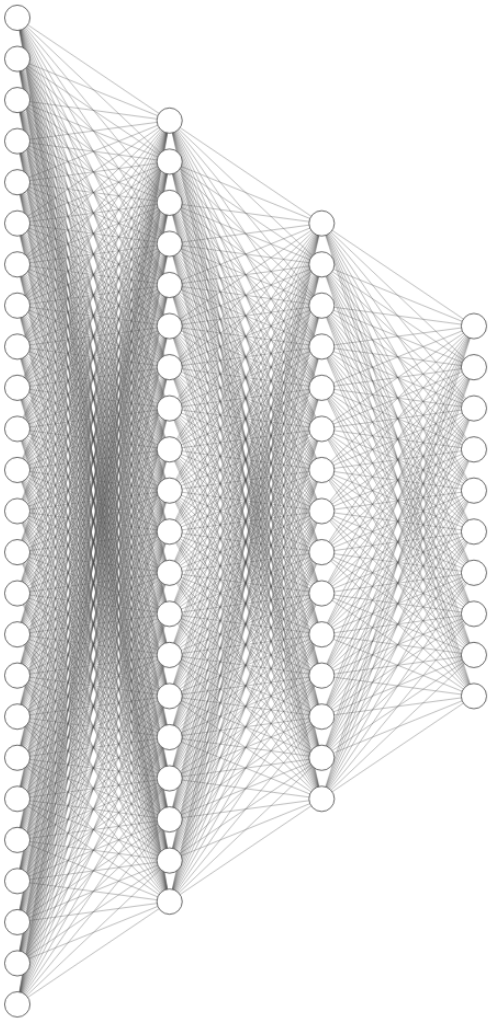

Figure 12: MLP For Winning Prediction

low, with a maximum at 30%, suggesting that for this simple model, the knowledge of the headline alone is not sufficient to predict the Winner of a test.

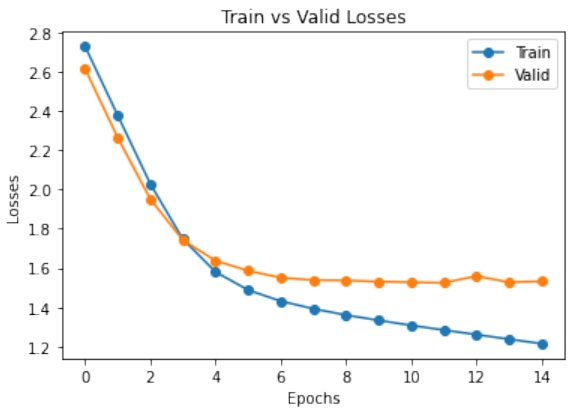

Figure 13: Winning Prediction: Loss

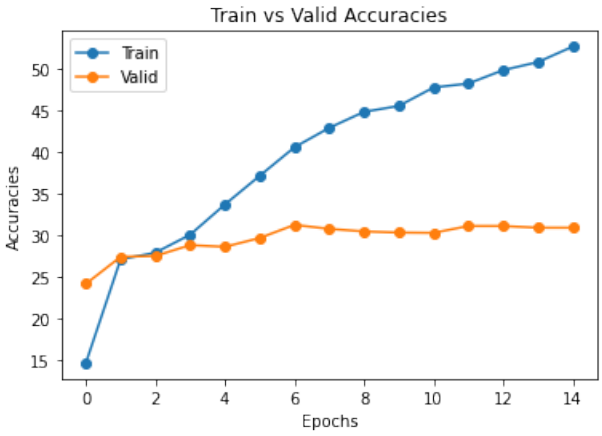

Figure 14: Winning Prediction: Accuracy

9 THE TIME DYNAMICS OF CLICK THROUGH RATE AND SENTIMENT

Dynamical aspects of the Upworthy data-set are now investigated. The research aims to answer two questions:

1. How has click through rate changed over time?
2. How has sentiment changed over time?

QUESTION 3.1 HOW HAS CTR CHANGED OVER TIME?

The CTR of all cleaned packages (see section on data cleaning) were grouped by week and aggregated by mean to produce a time series of average CTR spanning the duration of the Upworthy tests (2 years, 3 months, and 6 days). Figure 15 displays this time series. A striking feature of this plot is the reduction in CTR over time. This immediately raises the question whether the times series can be suitably modelled using a non-stationary ARIMA model. A time series $\{Y_t\}$ is generated from an ARIMA(p,d,q) model if and only if the the following holds:

$$\phi(B)(1-B)^d(Y_t) = \theta(B)\epsilon_t$$

Where ϵ_t is a white noise, B is the backshift operator and: $\phi(B) = (1-\varphi_1 B)...(1-\varphi_p B)$ where $|\varphi| < 1$, and $\theta = 1 - \theta_1 B - ... - \theta_q B^q$ has no unit roots.

The first step in model identification is to use the sample auto-correlation plot (ACF plot) of the time series to determine if we have stationarity or not (Figure 16).

The Sample Auto-correlation decays linearly as lag increases, implying that the time series is non-stationary. This was formally tested using an Augmented Dickey-Fuller test. The null-hypothesis in this test is that the model has a unit root (d > 0 in the model statement). The alternative hypothesis is that the model

has no unit roots (d = 0). This is by definition equivalent to testing the null hypothesis that the model is not stationary. We chose to conduct the test at a 5% significance level. We obtained a p-value of 0.386 for the test. Therefore, we fail to reject the null hypothesis and conclude that it is highly likely that the series is non-stationary.

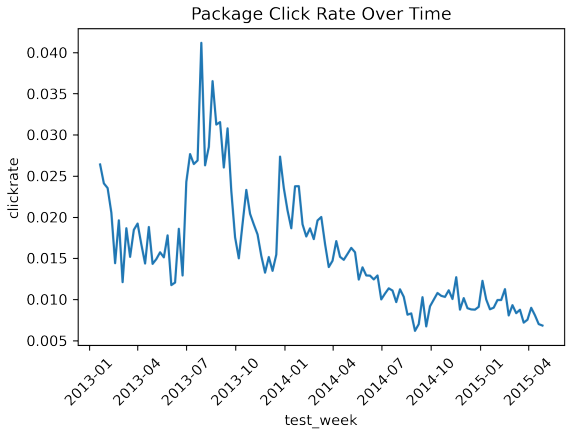

Figure 15: Time Series of Package CTR

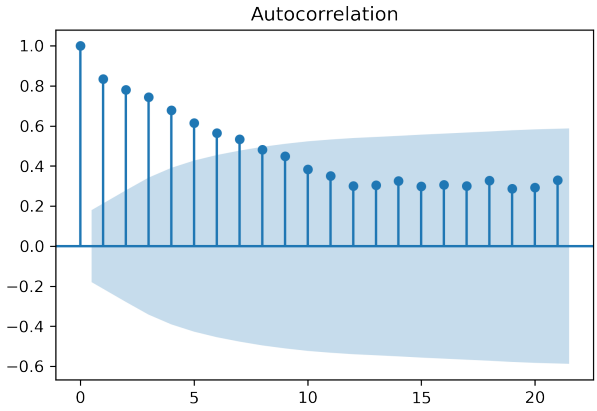

Figure 16: sample ACF for Package CTR

With non-stationarity established, we must obtain a stationary series so that we can identify the other model parameters. This is done by differencing the series, and checking if the resulting series is stationary (if not, we repeat the process). Figures 17-19 show the differenced time series, along with its ACF and PACF plots.

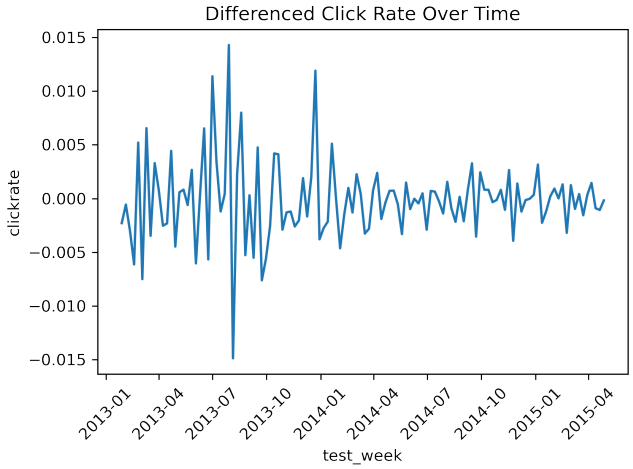

Figure 17: Differenced TS

We check for stationarity as before, by running an Augmented Dickey-Fuller test. This time the p-value is 2.8e-19 which is highly significant. We reject

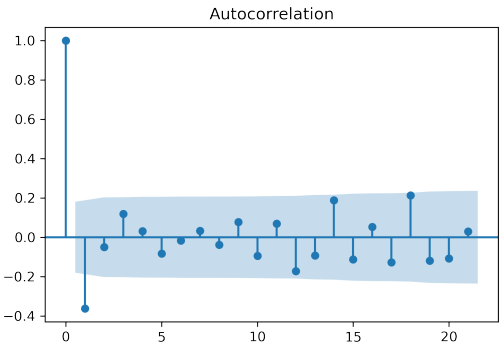

Figure 18: Sample ACF of Differenced TS

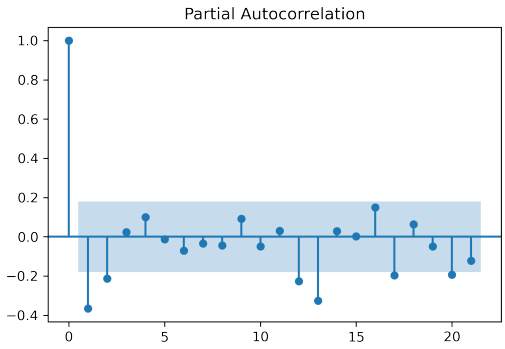

Figure 19: Sample PACF of Differenced TS

the null hypothesis, in favour of the alternative; the differenced time series is stationary. the ACF plot confirms this stationarity (no linear-decay). Therefore the original time series is integrated of order one. All auto-correlations are null after lag one in the ACF plot. The partial auto-correlations decay exponentially to zero. The combination of these observations implies that the differenced time series ΔY_t may be modelled as an moving average of order 1 (MA(1)) model. The model is stated as follows: $\Delta Y_t = \epsilon_t - \theta_1 \epsilon_{t-1}$

In this case, the full model would be an ARIMA(0,1,1) where we include a drift parameter μ which can be interpreted as the average CTR:

$$Y_t = \mu + Y_{t-1} + \epsilon_t - \theta \epsilon_{t-1}$$

However, before fitting the model above it is noted by inspection of the plot of the differenced time series above that the time series may exhibit heteroskedasticity. This is a violation of the ARIMA model above, which assumes homoskedasticity. Once the model is fitted, we can formally test if heteroskedasticity is present using the residuals.

We now fit the model to the observed package CTR time series. This will allow us to estimate the parameters μ and θ. The goal of this section is to investigate the change in CTR over time rather than predict future changes, therefore we fit the model on the entire time series, rather then create a train test split. Table 2 and Figures 20 through 23 summarise the fitted model.

Paremeter	Coeficient	Std err	$P > \|z\|$
$\hat{\mu}$	-0.4146	0.061	0
$\hat{\theta}$	1.411e-05	1.03e-06	0

Table 2: Summary of Fitted ARIMA(0,1,1) Model

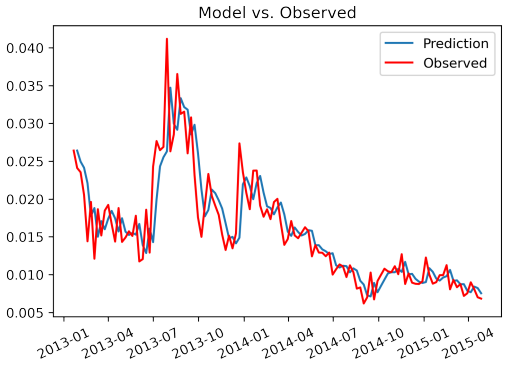

Figure 20: Fitted ARIMA(0,1,1) Model

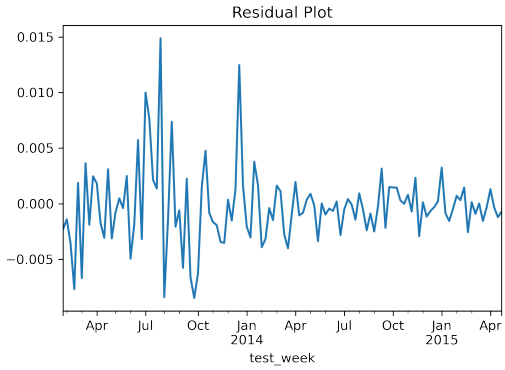

Figure 21: Model Residuals

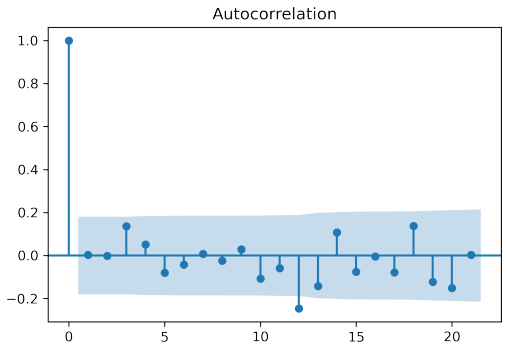

Figure 22: Residual Sample ACF

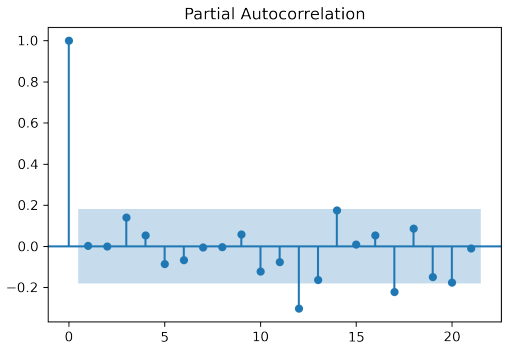

Figure 23: Residual Sample PACF

The model coefficients are highly significant. In addition, the ACF and PACF plots of the model residuals show no significant correlation at any lag. (with the exception of lag 13 of the PACF. But this value is not worrying, since for at a 5% confidence level there is a $1 - 0.05^2 0 = 0.64$ probability of seeing one significant value when there are 20 sampled lags). The correlation of residuals is formally tested using the Ljung-Box test. The Null Hypothesis for this test is that the residuals are uncorrelated. As usual, we choose to test at a 5% significant level. The obtained test statistic 26.09 has an associated p-value of 0.202928. Therefore, we fail to reject the null hypothesis and conclude that the model residuals are uncorrelated.

The homoskedastic assumption for the model mentioned earlier is now inves-

tigated. Examining figures 24 and 25 are useful before conducting a formal hypothesis test. The squared series does seem to contain volatility clusters, indicating heteroskedasticity. This often occurs when model variance is conditional on previous values of the series and/or previous variance values. The squared residuals ACF plot provides further evidence of heteroskedasticity. When we have heteroskedasticity, we expect that the squared residuals will be correlated at low lags. this is the case here, as we see in figure 25. Lags one, three and four are marginally significant, while all other lags are null (except for lag 21, which is likely an outlier). To formally test the Null hypothesis that we have homoskedasticity, a McLeod-Li test is conducted. This is essentially a Ljung-Box test, where squared residuals replace residuals. The test is conducted at 5% significance. The test statistic is 42.5 coinciding with a p-value of 0.003602. This is less than the significance level 0.05, therefore we reject the null hypothesis in favour of the alternative. It is concluded that heteroskedasticity is present in the model.

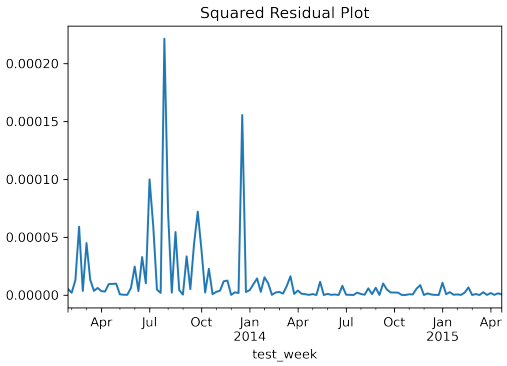

Figure 24: Squared Residuals

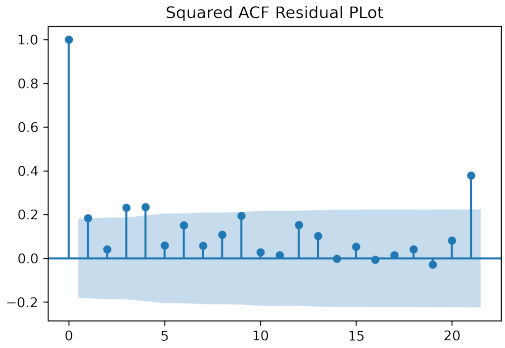

Figure 25: Squared Residual ACF

In light of this discovery, our model must be augmented. We model the residuals of the ARIMA(0, 1, 1) model using a GARCH(p,q) model. The Generalized Auto-Regressive Conditional Heteroskedastic (GARCH(p,q)) component models the conditional variance discovered above. It is stated as follows:

$$\eta_t = \delta_{t|t-1}\epsilon_t$$

$$\delta_{t|t-1} = \omega_0 + \alpha_1\eta_{t-1}^2 + ... + \alpha_q\eta_{t-q}^2 + \beta_1\delta_{t-1|t-2}^2 + ... + \beta_p\delta_{t-p|t-p-1}^2$$

Where $\delta_{t|t-1}$ is the volatility which is conditional on previous time points. The

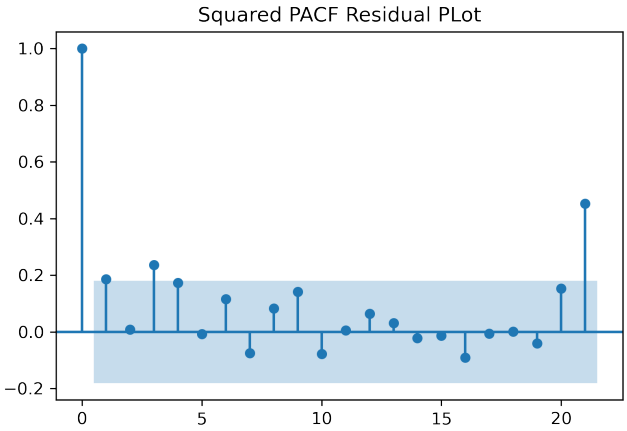

Figure 26: PACF of Squared Residuals

α_i and β_j are non-negative real numbers, and ϵ_t is a white noise. The component p can be identified from the ACF and PACF plots of the squared residuals of the ARIMA(0,1,1) fitted earlier (figures 14, 15). It can be shown that if η GARCH(p,q) then η^2 ARMA(max(p,q), p). The plots are typical of a MA(1) time series. These indicate GARCH(1,q) may be the best fit. Here q is estimated by repeatedly incrementing and testing each set of ω and α_q for significance. This is summarised in table 3.

Model	Coefficient	Estimation	$P > \|z\|$
GARCH(1,0)	ω	7.0649e-06	0
	α_1	0.2000	0.274
GARCH(1,1)	ω	2.8260e-07	0
	α_1	0.1	3.157e-02
	β_1	0.88	0
GARCH(1,2)	ω	2.8260e-07	0
	α_1	0.2	9.657e-03
	β_1	0.39	8.673e-03
	β_2	0.39	2.319e-02

Table 3: Summary of Fitted Models

A GARCH(1,3) model was also fit to the residuals, however two of its five coefficients were not significant, indicating it was overfit. We had Akaike's Information Criteria (AIC) and Bayesian Information Criteria (BIC) scores of -

1012.89 and -1004.57 for the GARCH(1,1) model, and scores of -1012.41 and -1001.32 for the GARCH(1,2) model. The marginal superiority of GARCH(1,1) over GARCH(1,2) in BIC may indicate slight over-fitting since this criterion penalises additional parameters more than the AIC for models with large sample sizes. Our chosen model is accordingly GARCH(1,1).

Finally, we examine the residuals of the fitted GARCH(1,1) model. Since we have heteroskedasticity, we must standardize residuals. Figures 28,29 show the sample ACF and PACF of the standardized residuals. Both are indicative of a white noise. A Ljung Box test is conducted to test the null hypothesis of uncorrelated residuals. We fail to reject the null hypothesis at 5% significance (p-value: 0.715727). A McLeod-Li test is performed to test the null hypothesis that model residuals have constant variance. We fail to reject the null hypothesis at 5% (p-value: 0.562206). This is the desired result since it implies our GARCH modelling has successfully accounted for the conditional relationship of the residuals. Finally, a Shapiro-Wilk is again performed, and once again the null-hypothesis is rejected (p-value: 3.494e-06). The residuals are not normally distributed. Inspection of figure 27 shows that there is again divergence from theoretical quantiles in the tails. It is hypothesised that a log-normal distribution could be applied to the residuals for predictive purposes, this is discussed further in the non-technical executive summary. Parameter estimation for the full model is now conducted. ARIMA parameters must be re-calculated simultaneously with the GARCH parameters, since it can be shown that if we estimate the ARIMA and GARCH parameters separately our results will be inconsistent. There is no python package with direct functionality, so this estimation was conducted using r. The estimated parameters are summarised in table 4 below.

	μ	θ	ω	α_1	β_1
Estimate	-1.37e-04	-4.86e-01	1.71e-08	1.30e-01	8.66e-01

Table 4: Summary of Fitted Models

Our final model, ARIMA(0, 1, 1) + GARCH(1,1), for the CTR time series is given by the following set of equations:

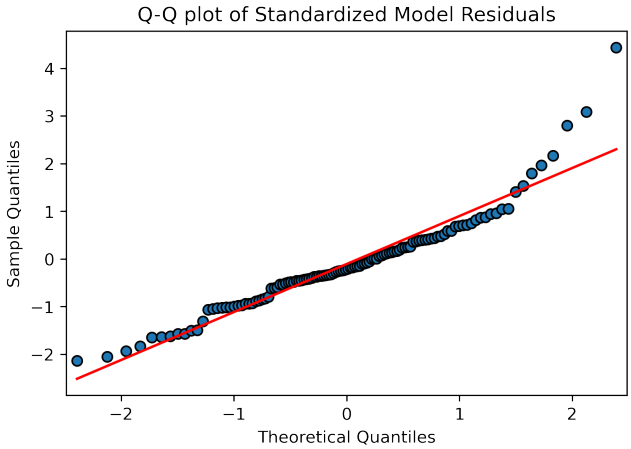

Figure 27: GARCH Residuals PACF

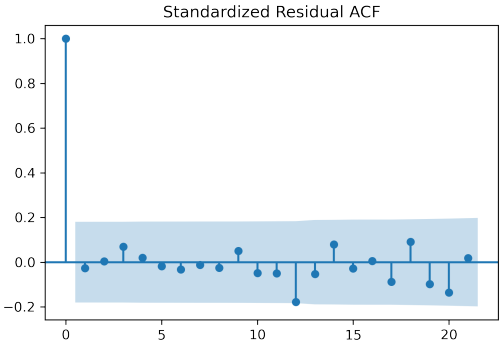

Figure 28: GARCH Residuals ACF

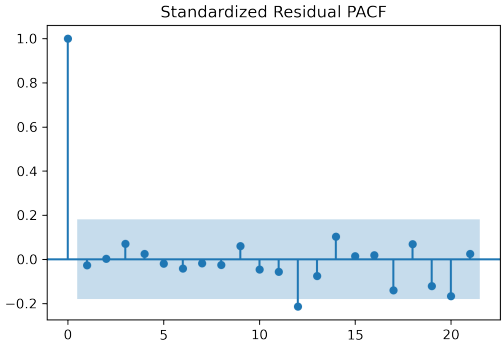

Figure 29: GARCH Residuals PACF

$$Y_t = \mu + Y_{t-1} + \epsilon_t - \theta\epsilon_{t-1} + \eta_t$$

$$\eta_t = \delta_{t|t-1}\epsilon_t$$

$$\delta_{t|t-1} = \omega_0 + \alpha_1\eta_{t-1}^2 + \beta_1\delta_{t-1|t-2}^2$$

Where $_t$ is the average CTR for packages in a given week t, ϵ_t is a random white noise term in week t, ν_t is the heteroskedastic component, and $\delta_{t|t-1}$ is the conditional volatility at time point t given time point t-1. The parameters μ, θ, ω, α_1, β_1 are parameters, given by the estimates -1.37e-04, -4.86e-01, 1.71e-08, 1.30e-01 and 8.66e-01 respectively.

QUESTION 3.2 HOW HAS SENTIMENT CHANGED OVER TIME?

To make a statement about the change in sentiment over time, we use the sentiment scores computed in section 6 as a proxy. For each observation, in our case each unique combination of image ID and test ID with at least two different headlines, no duplicate headlines and more than 0 clicks in the whole test, we get three columns of sentiment scores: negative, neutral and positive. All 3 columns together add up to 1. Note that in a statistical setting, these per-category values obtained from the output of a deep neural network cannot be interpreted as actual probabilities or as a good measure of uncertainty, i.e., as confidence scores [11]. Therefore, for each of the three sentiments, we create a binary variable indicating whether the particular sentiment was the strongest for that observation. Since the prediction power is very high compared to the estimated uncertainty of the prediction, the risk of false conclusions is mitigated. We also want to know how sentiment changed in the relevant packages, i.e., the picked packages. Therefore, for each sentiment, we create another column with a binary variable indicating whether the observed package was mainly assigned to the respective sentiment category and picked.

Similar to the question about the change in CTR over time, the packages are also grouped by week to perform further analysis of sentiment over time. We aggregate over the newly created binary columns, summing the number of true values. From then on, we obtain distributions over the three sentiments per week for all tested packages and for all picked packages. To obtain comparable results and account for the different number of impressions and clicks, we normalize by creating new columns containing the percentage values of the picked and tested packages for each sentiment category for the week. Looking at the three time series in Figure 30, one for each percentage value of picked packages belonging to a particular sentiment category, it is clear that the distribution is quite stable over time. Headlines with negative sentiment are picked more often than others, followed by headlines with neutral sentiment.
Before we fit a model to the 3 given distributions, we can already draw some initial conclusions by visually inspecting the plot. There is a strong relationship between all 3 processes as they all add up to 1. It is also clear that no process

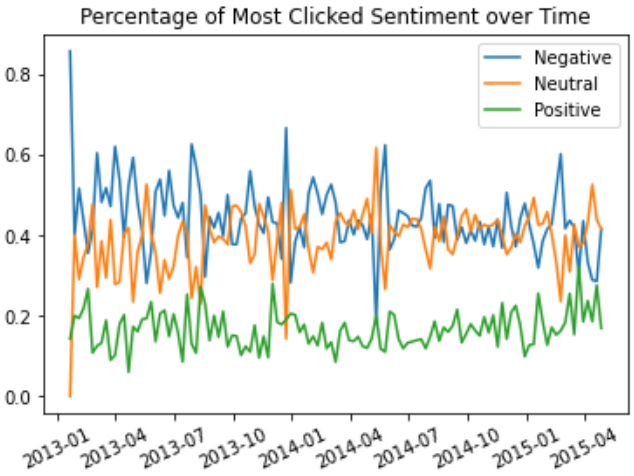

Figure 30: Time Series of Picked Package Sentiment

has a strong trend. To gain further insight into the true but unknown underlying data generation process, we will apply a series of statistical tests and inference methods to the given data. The main focus will be on the interpretability of the results and inferring causal relationships between sentiment over time and the previous analysis of CTR over time.

A first step in time series modelling, is to check for stationarity. As discussed before in section 7.1, we perform an Augmented Dickey-Fuller test on the data. Remember, the null hypothesis is that a unit root is present, which is by definition equivalent to a non-stationary time series. Intuitively, this means we have a non-mean-reverting process that is more difficult to model. As visually inspected, all 3 processes are stationary and achieve p-values in the ADF test close to 0. This means we reject the null hypothesis in favour of the alternative. It turns out, that modelling the positive sentiment over time is more interesting than the negative or neutral sentiment, as those are best modelled by a constant. As a comparison to CTR over time, the ACF and PACF plots of the positive sentiment over time depicted in figure 31 and figure 32 show a very rapid drop in significance, implying stationarity and no significant seasonality. In fact, the signal appears to be weak and the noise strong.
In order to correctly fit the time series with a model from the ARMA family, we must first make sure that this is a valid approach, and the necessary conditions are fulfilled. For an ARMA model in general, stationarity must be given, most other conditions, apart from the definitions, are related to the estimation

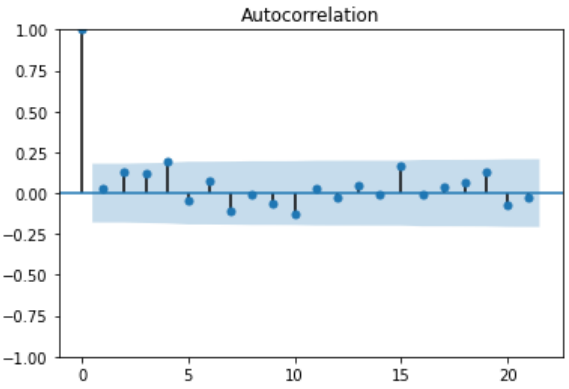

Figure 31: Sample ACF for Positive Sentiment Rate

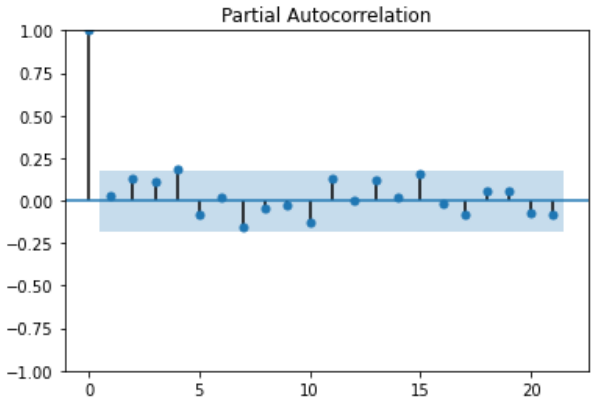

Figure 32: Sample PACF for Positive Sentiment Rate

method. One of the most commonly used methods is the Box-Jenkins method that uses a 3-stage approach: model identification, parameter estimation and statistical model checking. The most important assumptions for using the model are normally distributed errors, which for a linear time series is the same as having normally distributed observations and no exogenous variables. Models with exogenous variables are called ARIMAX or variants thereof and the normality assumption can be relaxed as they are mainly used to estimate likelihoods and apply the AIC for feature selection [12].

In the model identification stage, the Box-Jenkins method tests for stationarity and removes it by differentiating if present. Furthermore, it tests for seasonality and accounts for that by adding a seasonal component if necessary. This can often also be assessed from a ACF plot. In the parameter estimation step, maximum likelihood estimation is used to determine the best fit of parameter p and q for the ARMA(p,q) model. For inference we prefer simpler over more complicated models, to account for that the Akaike Information Criterion (AIC) defined as

$$\text{AIC} = 2k - 2\log \tilde{L}$$

is applied during the parameter estimation step. $\tilde{L}$ stands for the maximum value of the likelihood function, often a Gaussian likelihood is taken for the sake of simplicity, k represents the number of parameters in the model that we are penalising to favour a simple model. It must be noted, that many implementations of the Box-Jenkins algorithm allow for the usage of the Bayesian Information Criterion (BIC) instead of AIC, often this does not influence the results as

in this examined case.

SARIMAX Results

Dep. Variable:	y	No. Observations:	119
Model:	SARIMAX(1, 0, 2)	Log Likelihood	198.112
Date:	Sat, 06 Nov 2021	AIC	-386.223
Time:	19:56:35	BIC	-372.327
Sample:	0	HQIC	-380.581
	- 119		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	0.0439	0.038	1.153	0.249	-0.031	0.118
ar.L1	0.7367	0.229	3.214	0.001	0.287	1.186
ma.L1	-0.7684	0.228	-3.364	0.001	-1.216	-0.321
ma.L2	0.1916	0.092	2.082	0.037	0.011	0.372
sigma2	0.0021	0.000	7.308	0.000	0.002	0.003

Ljung-Box (L1) (Q):	0.04	Jarque-Bera (JB):	5.98
Prob(Q):	0.83	Prob(JB):	0.05
Heteroskedasticity (H):	0.78	Skew:	0.54
Prob(H) (two-sided):	0.43	Kurtosis:	3.18

Figure 33: Summary Output of the Positive Sentiment Model Fitting Using Box-Jenkins

The last step after finding the model is to perform statistical model checking, that involves tests for heteroskedasticity as described in the chapter before, as well as the Ljung-Box and the Jarque-Bera tests. When applied to the positive sentiment over time series we get the results in figure 33 It shows that the preferred estimated model is an ARIMA(1,0,2) which is equivalent to an ARMA(1,2) model, because the series is stationary even without differentiating. The variables are significant on the 5% level, but the intercept is not. A test not mentioned so far is the Jarque-Bera test that measures how close the time series is to a normal distribution. The lower the test value, the closer to a normal distribution the tested series is. A more intuitive way of checking the major assumptions for a model is to get common diagnostic plots. In figure 34 we can see the standardised residuals plot that shows the residuals divided by its standard deviations. The histogram of our models residuals and an overlaying standard normal density as well as an estimated density for the residuals of the model can be seen in plot 35. The QQ-plot to compare the sample and theoretical quantiles and visually check for normality and the Correlogram

that plots the autocorrelation against a lag h can be observed in figures 36 and 37, respectively. Normality assumptions are clearly observable in the plots, no other violations of assumptions seem present.

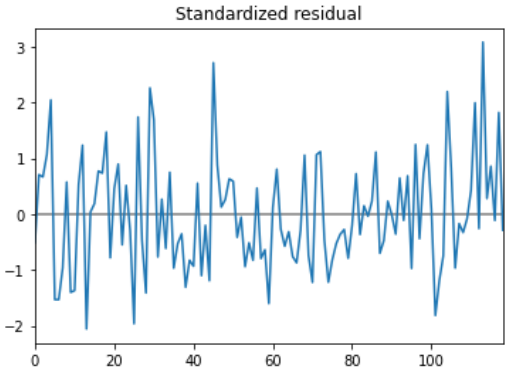

Figure 34: Standardized Residuals of the ARMA(1,2) Model

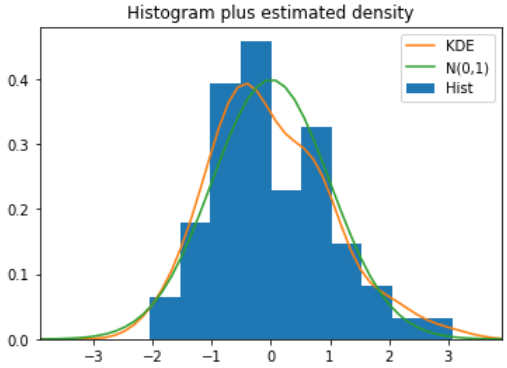

Figure 35: Histogram of the Residuals

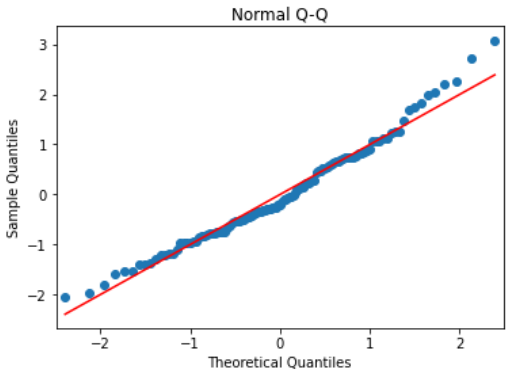

Figure 36: QQ-Plot of the ARMA(1,2) Model Residuals

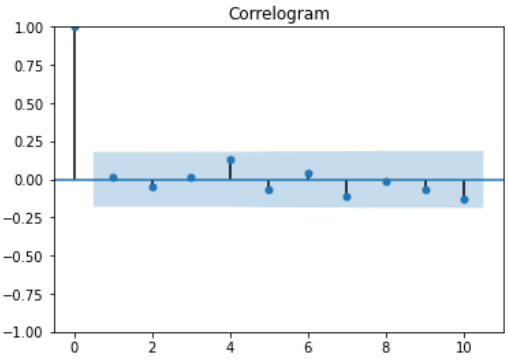

Figure 37: Correlogram of the ARMA(1,2) Model

To conclude, it can be stated with a high certainty, that the sentiment of the most clicked packages over time remain constant. The negative and neutral time series both resemble stationary process without any trend or seasonality, but with some fluctuations over time, while the positive sentiment is best modelled with an ARMA(1,2) process. Given the structure of the A/B tests in the data set and the performed statistical tests, we can conclude that although negative sentiment in the headlines lead to higher CTR, the proportion of negative headlines that outperformed neutral or positive headlines in causal statistical tests, is independent of time.

References

[1] Matthem Ingram. *The internet didn't invent viral content or clickbait journalism — there's just more of it now, and it happens faster*. 2014. URL: https://gigaom.com/2014/04/01/the-internet-didnt-invent-viral-content-or-clickbait-journalism-theres-just-more-of-it-now-and-it-happens-faster/ (visited on 04/01/2014).

[2] Liam Corcoran. *Top Social Publishers 2013*. 2014. URL: blog.newswhip.com (visited on 10/23/2014).

[3] James Ball. *Read this to find out how Upworthy's awful headlines changed the web*. 2014. URL: https://www.theguardian.com/media/2014/mar/16/upworthy-website-generation-y-awful-headlines (visited on 07/06/2015).

[4] *About Us*. 2021. URL: https://goodinc.com/ (visited on 11/04/2021).

[5] John Gramlich. 2020. URL: https://www.pewresearch.org/fact-tank/2020/03/23/use-our-global-indicators-database-to-analyze-international-public-opinion/ (visited on 03/03/2020).

[6] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

[7] Juan Manuel Pérez, Juan Carlos Giudici, and Franco Luque. *pysentimiento: A Python Toolkit for Sentiment Analysis and SocialNLP tasks*. 2021. arXiv: 2106.09462 [cs.CL].

[8] J Deacon. URL: http://archive.bio.ed.ac.uk/jdeacon/statistics/tress4.html (visited on 11/07/2021).

[9] Carere Christian. *10 Amazing Headline Strategies To Improve CTR - Digital Ducats Inc*. URL: https://digitalducats.com/headline-strategies-improve-ctr/.

[10] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. 2017. arXiv: 1609.04747 [cs.LG].

[11] Yarin Gal and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *international conference on machine learning*. PMLR. 2016, pp. 1050–1059.

[12] Mark E Lehr and Keh-Shin Lii. "Maximum Likelihood Estimates of Non-Gaussian ARMA Models". In: *1998 Symposium on Nonlinear Time Series Models*. 1998.