Predicting User Engagement based only on Article Titles

1802525 - dy18708@essex.ac.uk

MA981-7-FY, MSc Dissertation

Abstract

News articles consumption is an integrated part of people's life. However, not many of these articles are equally engaging, and so the prediction of the article's popularity is an important technique for news organizations to evaluate which articles have to be streamlined. There are a variety of factors that may influence the impact of an article. The main objective of this paper is to investigate the assumption that the way article titles are formulated highly contributes to the level of user engagement. The given dataset contains 5000 legal article descriptions and article activities collected from Mondaq Website. We have adopted a two-stage approach. First, we focused on understanding data properties and derive insights of user engagement using exploratory data analysis. Second, we proposed a supervised three state classification which predicts the high, moderate, and low level of user engagement based on article titles. As part of exploratory data analysis, We also extracted text-based and natural language-based features from article titles and analysed to recommend data insights to news organizations. In terms of modeling, we have presented a comparative analysis of the success rates of state-of-the-art prediction algorithms, which are Naive Bayes, Random Forest, Support Vector Classification, Extreme Gradient Boosting algorithms, and Stacking models. Also, we constructed a neural network using the multilayer perceptron model to capture the non-linearity of features to obtain a better prediction score. These State of Art Machine learning models have been trained using Bag of words and Term Frequency–Inverse Document Frequency (TFIDF) vectorization techniques whereas Artificial Neural networks used word embeddings as a layer to convert text to vectors. Our model evaluation results highlight that the best method turned out to be the Naive Bayes modeling using the TFIDF vectorization technique.

Contents

List of Tables

List of Figures

1 Introduction

The service life of a news article is, by its existence, very short (following its publication). Predicting the early popularity of an article, rather than its long-term popularity, seems to be more interesting and useful. In this paper, therefore, the article's popularity is defined as the user engagement profiles of a high, medium, and low within the first two weeks following publication.

The reasons behind the news article's popularity are typically varied and may include contemporaneity, quality of writing, and other latent factors. By predicting which articles would become influential so that resources can be distributed effectively to support the better user experience. The predicted probability is used to determine which article impressions the user. The prediction accuracy not only determines the placement of the article but also the performance of the article.

Imagine a user visiting a website, and doing a news search. From the results shown, the user can click on some of the results he/she is curious in, and after reading the details of the news item, they click on the item to go to the article page. There are a variety of factors that may influence the impact of an article, including the context and availability of the news article in which it is published, the type of publication topic, its author(s), its subject, its length, and so on. The title is one of these factors which plays a crucial part in engaging a user. We aim to find out whether there is any significant predictive relationship between the article title and user's interest levels by studying 5000 articles published on Mondaq Website. Titles also play a key role in the branding of the article, as hundreds of articles are published annually in each subject and a manner competes with each other to be viewed. The article's title precedes the beginning and should give a strong indication of the subject and arouse interest. The title of an article precedes the beginning and should give a strong indication of the subject and cause interest. An appropriate title is very simple, insightful, and appealing. But as Kane (2000)[18] suggested these qualities are difficult to balance, and most titles appear to be appealing but not insightful, or insightful but not appealing. Each title should encourage the reader to read an article; provide the users with a brief outline of the content, an overview of the topics and findings discussed; introduce how users look at the listed things in (Ball 2009)[2] and attract the reader's attention to an article by reminding them of its content (Manten and Greenhalgh 1977)[25]. Most writers may not however select a correct title for their papers (Manten and Greenhalgh 1977)[25]. Hereby assuming that the way article titles are formed contributes to user engagement.

Traditional workflow to predict user engagement based on the article title will be of 3 stages. In, first stage Domain experts select a list of article features of interest that they want to analyze. During, the second stage they will search for evidence in the article literature that somehow is relevant to the article features of interest. Finally, the domain expert analyzes the evidence related to each of the features to evaluate them. The main objective is to predict user engagement using the given article title based on evidence from Article literature. The first stage is pretty simple as it is directly related to domain knowledge, the second

stage is a simple data web scraping or collection of historical data whereas, during the third stage, the domain expert has to spend a huge amount of time and efforts in analyzing the evidence to find user engagement.

This kind of business objective can be solved efficiently using data science techniques. Formulating to data science problems leads to unearthing reasons behind user clicks using data analytics and building a predictive model to predict user engagement using machine leaning classification approach.

2 Related Work

News article popularity prediction is a fairly novel challenge and very few studies addressed this challenge. However, an increasing number of studies on predicting the popularity of certain forms of online content has been carried out. The objective of these studies include estimating the number of page clicks for a news article[26], news article ranking[30], predicting popularity ranges for a news article[3], number of comments prediction for a news article [32, 30], current studies pose the challenge of predicting popularity as one of classification [32, 20], regression [23], or clustering [12]. These studies employ several content-based features to predict news article popularity[31]. These content-based features are typically created from news articles text such are text sentiments[4, 28], text emotions[5, 6], named entities[3] are all viewed to be strongly correlated contributors to articles virality. Similarly, this paper also derives natural language processing based features such as polarity, subjectivity, parts of speech tags, and finally named entities from the title text.

Similar to our study of investigating user engagement prediction using article titles, Hardt and Rambow (2017)[13] states the choice to click on an article will be based solely on the title – the article is only visible after the decision to click is made. According to them, the title of the article provides a lot of information that helps users to decide to view. (Hardt and Rambow, 2017)[13]. In other words, readers will predict an article's content from a headline in advance because of the linguistic and world information they carry to bear when analyzing the headline.

The titles of the article were examined from various perspectives including linguistic, science, and academic communication. Yitzhaki (1997)[36] compared the title substantive informativity with social sciences in between 1940 and 1960 and found substantial differences. Yitzhaki (1994, 2002)[35, 37] analyzed the length of the article and found a strong correlation between the length of the title, the number of writers, and the article's length. Lewison and Hartley (2005)[24] analyzed the number of words and the use of the colon in article titles and observed that they mostly increased from 1981 to 2001. (Sagi and Yechiam 2008; Jacques and Sebire 2009)[29, 15] determined possible citations using artitle titles. Almost 20 million research papers has been reviewed by Ball (2009)[2] and he observed a major rise in the number of publications with question mark titles over the 40 years from 50 percent to over 200 percent. He cited 'marketing factor' as one of the main factors behind the use of question mark titles in

articles. Hartley has studied the existence of colon in titles and its connection to citation rate (2007a, 2007b)[10, 14]. He demonstrated that there were differences in the use of the colon in article titles in various disciplines and stated that colon use had no impact on their corresponding citation rate. Jacques and Sebire (2009)[15] reviewed the characteristics of articles title and they found that title's length, colon presence is positively correlated with article citation rate. Similar to all these studies, this paper also derives text-based features such as title length, word count, word density, punctuation count, words ending with punctuation count, upper case word count, stop word count, questioning Word count, from the title text.

Chakraborty et. al[8] developed a clickbait classifier on a corpus of 15000 article headlines scraped from wikinews and various tabloid news websites. Their best-performing classifier used n-grams in a Support Vector Machine (SVM) model to achieve an accuracy of 0.93. Lagun and Lalmas (2016)[21] examine evaluation metrics for user engagement of news article reading, stating that a small set of confusion matrix metrics can predict whether a user will read the complete article.

In recent years, the prediction of Click Through Rate has been of great concern to academic computer advertising communities. Chakrabarti et al[7] predicted the logistic regression-based Click Through Rate and employed cumulative factorization to model interaction effects for several regions. Regelson and Fain [27] predicted the term degree of Click Through Rate and adopted hierarchical clusters for lower frequencies or new words. The Click Through Rate was predicted by Dembczynski et al. [9] in the light of decision rules. Xiong et al. [34] developed a model focused on continuous conditional random fields and considered both ad characteristics and surrounding similarities. Wang and Chen [33] predicted Click Through Rate by training several machine learning models Support Vector Machine(SVM), decision tree, and Artificial Neural Networks(ANN). They have used user searching and clicking logs as data.

3 Methodology

The technology stack used for the overall project is Python, Jupyter notebook, and PowerBI. Python is a general-purpose high-level programming language which contains readily available packages for artificial intelligence through Scikit-learn, Tensorflow, and Keras. Jupyter notebook is an open-source web application that helps to perform a wide range of workflows in data science.Finally, PowerBI is a business intelligence tool served by Microsoft which aims in providing interactive visualizations.

3.1 Data Description

Data has been collected from Mondaq website. There website is based on the MVC .Net Core framework, which is used for tracking events on the website, which are then uploaded in real-time into the Microsoft SQL database. A sub-

set of data has been considered to perform analysis. Data comprises 5000 legal article descriptions and article activity which is associated with meta information including total clicks in the first 2 weeks since articles publishing date. A detailed description of features are shown in table 1 and table 2.

Feature	Description
Article id	Unique article identifier
Title	Articles title
Topic desc	Articles topic description
Article publish date	Date when article was published on mondaq.com
Country desc	Articles country description/name (i.e. which country does the article cover)

Table 1: Article Description

Feature	Description
Article id	Unique article identifier
Title	Articles title
Session tracking id	Unique event identifier
Individual session id	Unique session identifier (i.e. there can be multiple unique events in a single session)
Click event date	Date when article was interacted with (i.e. clicked on)

Table 2: Article Activity

Table 1 contains variables used to describe articles such as title, publish date, associated country and topic. Similarly Table 2 contains activities related to articles such as click event date, session tracking identifier and individual tracking identifier. session Tracking identifier is a unique event identifier which will be generated whenever user logins the Mondaq website.On the other hand individual session identifier is an unique session identifier which will be generated whenever logged in user made any kind of activity such as article click.As a result, data contains multiple unique events in a single session. Both the tables contain article identifier which is an unique id used to join descriptions and activities.

3.2 Data Exploration

The main objective of data exploration is to understand data properties, exploring challenges in the data, and listing solutions to make data cleaner.

Data has been loaded to Jupyter notebook using pandas modules. Typical data points in data description and data Activity looks as shown in Table 3 and Table 4.

Article id	Article publish date	Title	topic desc	Country desc
715494	02/07/2018	Ontario Welcomes New Cabinet - Expect Quick Action	Government, Public Sector	Canada
716654	06/07/2018	Residential Focus – 4 July 2018: Part 2	RealEstate and Construction	Australia
718260	11/07/2018	Copyright Quarterly: Issue 1	Intellectual Property	Canada

Table 3: Data Points in Article Descriptions

Article id	Session tracking id	Individual session id	Click event date
726634	747820650	704312710	11/08/2018
728406	749792604	706344088	16/08/2018
726884	748993528	705447000	14/08/2018

Table 4: Data Points in Article Activities

Table 3 contains data points in article descriptions. Upon exploration of data and each variable, it is found that data contains 5000 entries with five features of integer and object datatypes. article publish date contains 57 unique dates ranging from 02 July 2018 to 28 December 2019. Title Feature contains 4918 unique titles among 5000 entries. The unique number of topics in data is 10 and finally, data distributed among Canada and Australia.

Data points in article activities are tabulated in Table 4. Upon exploration of Data and each variable, it is found that data contains 1,92,957 entries with four features of integer and object datatypes. Click event date contains 558 unique dates ranging from 02 July 2018 to 10 January 2020.

3.3 Data Preparation

As part of data preparation, article activity data and article description data are merged based on article id. The primary challenge of object data types has been solved by converting article publish date and click event date using 'to datetime' object in pandas. Secondly, the target variable has been created by generating new features named 'day' and 'number of clicks' from 'article publish date' and 'click event date'. The difference between 'article publish date' and 'click event date' are represented as 'day' and finally, 'number of clicks' has been generated by grouping 'click event date' and using count as aggregate function.Typical data points looks as shown in Table 5.

Table 5 contains data points from prepared data with associated variables, it

article id	title	article publish date	click event date	topic desc	country desc	Day	no of clicks
399550	Circumstantial Evidence And Insider Trading	30/08/2018	05/09/2018	Corporate/ Commercial Law	Canada	6	1
713470	Will An Abused Horse Named Justice Find Justice In Court?	19/07/2018	19/07/2018	Litigation, Mediation and Arbitration	Canada	0	1

Table 5: Data Points after Data Preparation

is found that data contains 37,442 entries with 8 features of integer and object data types. Created feature 'Day' ranges from 0 to 14. In other words, the 'click event date' is extracted up to 15 days from the 'article publish date'. The target variable 'no of clicks' is associated with each 'title' based on the counting of events and range from 1 click to 909 clicks.

3.4 Exploratory Data Analysis

The overall objective of this session is to perform tasks of bivarient analysis, engineering, and analyzation of text-based, Natural Language Processing(NLP) based features to derive insights from the data which in turn helps news organisations to take decisions related to data.It is essential to make clear that all findings we presented in this section relate to 5000 Mondaq Articles with a published duration of 2 weeks in the countries of Australia and Canada and findings may not be correct for other countries and other businesses.

3.4.1 Bivariant Analysis

Bivariant Analysis is one of the simplest ways of exploratory data analysis which considers the analysis of two variables, dependent and single independent features as shown in Figure 1.

On observation of 'number of clicks' and 'day', we can infer that first day of article publishment has the highest number of clicks and it gradually decreasing with an increase in the number of days. Users prefer the latest (or) current news which tends click rate may be high.

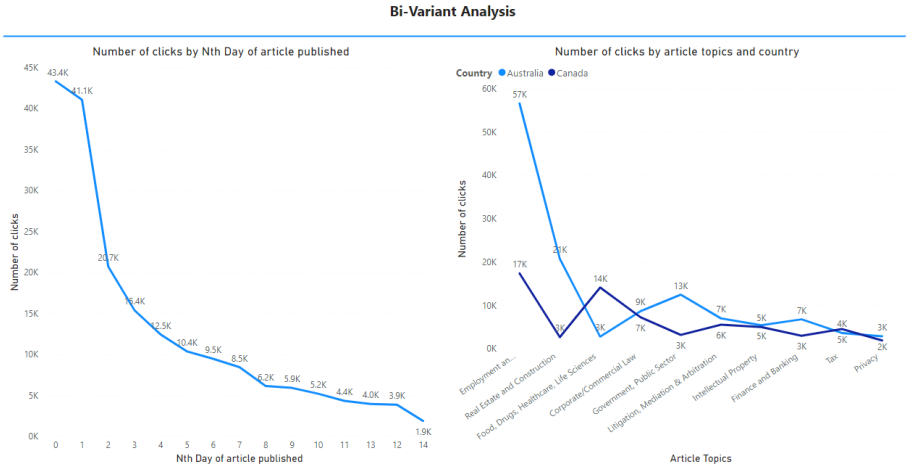

Figure 1: Bivariant Analysis

From the analysis of 'topic description' and 'number of clicks' we came to know that the 'Employment and HR topic' has the highest number of clicks of 74,000. Users are more interested in knowing about human resource management and job analytics. The secondary User interest topic is 'Real Estate and Construction' with 24,000.A similar analysis has been performed by considering country descriptions as legend given that Australia has a high number of clicks than Canada.

3.4.2 Text based Analysis

The text-based analysis is a way to understand detailed properties of text which,in this case, are article titles. As part of text-based analysis, we have created features such as title length, word count, word density, punctuation count, words ending with punctuation count, upper case word count, stop word count, questioning word count.

Table 6 contains engineered text and NLP features. 'title length' feature has been created by applying a function which was created to count number of characters (including spaces) in a given articles title. This function takes text as input and returns an integer value i.e., length of the title. As we have seen a title is a list of words, a feature named 'word count' has been created by counting the occurrence of the word. 'word density' has been featured using 'title length' and 'word count' which refers to the number of times words appeared in a given title. As punctuations also play a key role in posing titles. Hereby creating features 'punctuation count' which counts the number of punctuations in title and feature 'words ending with punctuation' which implies the status of punctuation in the last word. Upper case words and stop words associated with titles are counted and stored in features named 'upper case word count' and 'stop

Feature	Description
Title Length	total number of characters in project title including spaces
Word count	total number of words in the complete title text
Word Density	average length of the words used in the essay
Punctuation count	total number of punctuation marks in the essay
Words Ending with Punctuation count	total number of words ending with punctuation in the essay
Upper case word count	total number of upper count words in the essay
Stop word count	total number of stopwords in the essay
Questioning Word count	total number of questioning words in the essay
Polarity	polarity of a given text
Subjectivity	expression of opinions
Noun Count	total number of nouns in the text
Verb Count	total number of verbs in the text
Adjective Count	total number of adjectives in the text
Adverb Count	total number of adverbs in the text
Pronoun Count	total number of pronouns in the text
NER Label	NER label in the text
Entities	Entities in the text
Entity count	total number of Entities in the text

Table 6: Text and NLP based Feature Engineering

word count' respectively which helps to derive some detailed insights about the title. Finally 'questioning Word count' has been created which contains several questioning words such as 'how', 'why', 'what', 'who', 'will', 'when', 'which' in the respective title. Typical data points shown in table 7.

A similar type of bivariant Analysis has been performed which considers each text-based feature as an independent variable and 'number of clicks' as dependent features as shown in Figure 2.

By observing 'length' and 'number of clicks', we can infer that, until a certain point,title length increases with respect to length of 42 has the highest number of clicks and it gradually decreases with an increase in title length. It seems user's are preferring moderate title length.

From an analysis of 'word count' and 'number of clicks' we came to know that title containing word count of 9 has the highest number clicks of 45k.

When investigating the relationship between 'punctuation count' and 'number of clicks'we can see that the highest number of clicks happened for a punctuation count of one and then zero. Also as the punctuation count is increasing click rate has been decreasing.

By looking into the 'upper case count' and 'number of Clicks' we can say that a high number of clicks have happened for the titles having zero upper cases.It

Title	Length	Word count	Word density	Punctuation count	Upper case Word count	Stop word count	Ques Words	End Puct
Circumstantial Evidence And Insider Trading	43	5	7.16666	0	0	1	0	0
Will An Abused Horse Named Justice Find Justice In Court?	57	10	5.18181	1	0	3	1	1

Table 7: Data Points in Text based features

is evident that 77 percent of clicks are registered for the titles with lower cases.

The pattern from 'stopwords' and 'no of clicks' infers titles without stopwords have a higher click rate. Users may be not interested in looking at common words. The more the stop words less the click rate. They may be looking into content words.

Observing 'questioning words' and 'number of clicks', we can say that 91 percent of the clicks are registered with titles without question words. Henceforth interrogative titles may not catch the user's attention.

By concluding the text-based analysis it is recommended that article titles with moderate length, the word count of nine, at least one punctuation, lower casing, minimum stop words, and descriptive or declarative title can cause the high contribution to user engagement.

3.4.3 NLP based Analysis

Natural Language Processing(NLP) based analysis is a way to understand detailed linguistics text properties which is a title in this case and as part of NLP based analysis, we have created sentiment features such as polarity and subjectivity and part of speech tags features such as noun count, verb count, adjective count, adverb count, pronoun count. Finally named entities features such as entities, NER label, and entity count has been created.

Table 6 contains NLP based features. Textblob package has been used to get sentiment features and parts of speech tags features. The process of determining the emotion of the text or title is referred to as sentiment analysis. Any title can be distinguished as either positive or negative or neutral. Textblob sentiment function returns polarity and subjectivity properties which are used to distinguish the emotion or attitude of the title. Polarity is a float within the

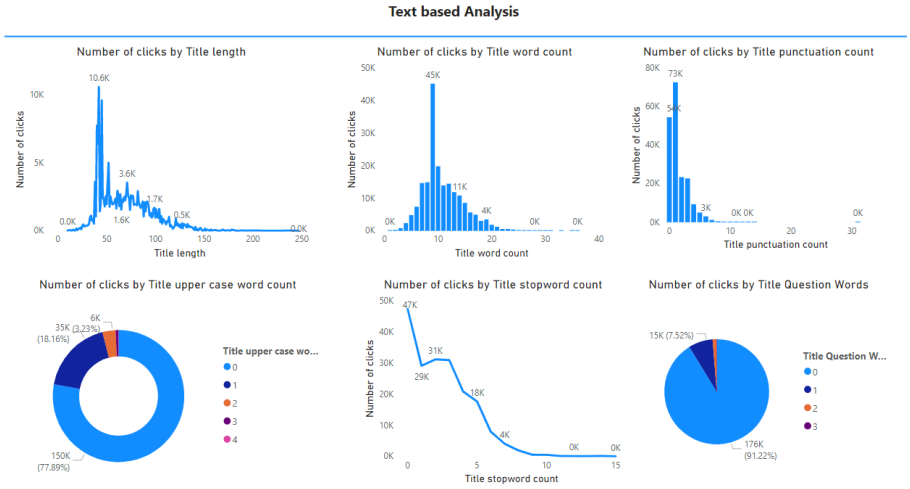

Figure 2: Text Based Analysis

-1 to 1 range where 1 is positive and -1 negative. In general, subjective sentences refer to personal opinions,judgments or emotions while objective refers to quantitative evidence. Subjectivity is likewise a float within the range of 0 and 1.

Parts of speech(POS) tagging is a very important step, to understand the meaning of any sentence or to extract relationships and build a knowledge graph. It is a process of marking a word in a text data, in the corresponding part of a speech tag, depending on its meaning and description. This function is not easy, as a specific word may have another part of speech depending on the context in which the word is used. Also, the basic natural language processing(NLP) models like bag-of-words(bow) fails to identify these relationships between the words. For that, we use POS tagging to mark a word to its POS tag based on its context in the data. POS is also used to extract the relationship between the words[38,39].

Named Entity Recognition (NER) is one of the key tasks of information extraction. It attempts to recognize and classify references to named entities in unstructured text into predefined categories such as people names,organisations, places, amounts etc. Typically this is split into two main phases: detection and typing of entities (also known as classification) (Grishman Sundheim, 1996)[7]; It is a critical task since it identifies which snippets are references to entities in a text very real world.[38,39] Spacy(https://spacy.io/) with loaded Core English small version Library has been used to extract entities from the title. It is a Python open-source library that parses and "comprises" large volumes of text and is designed to effectively handle NLP tasks with the most efficient implementation of common algorithms. typical data points of NLP based features are shown in table 8.

title	polarity	subjectivity	noun count	verb count	adj count	adv count	pron count
Circumstantial Evidence And Insider Trading	0	0	3	0	1	0	0
The New Construction Act: Summary And Timelines Of Major Changes	0.099431	0.477272	5	0	2	0	0

Table 8: Data Points in NLP based features

A similar type of bivariant analysis has been performed which considers each NLP based feature as an independent variable and 'number of clicks' as dependent features as shown in Figure 3 and Figure 4

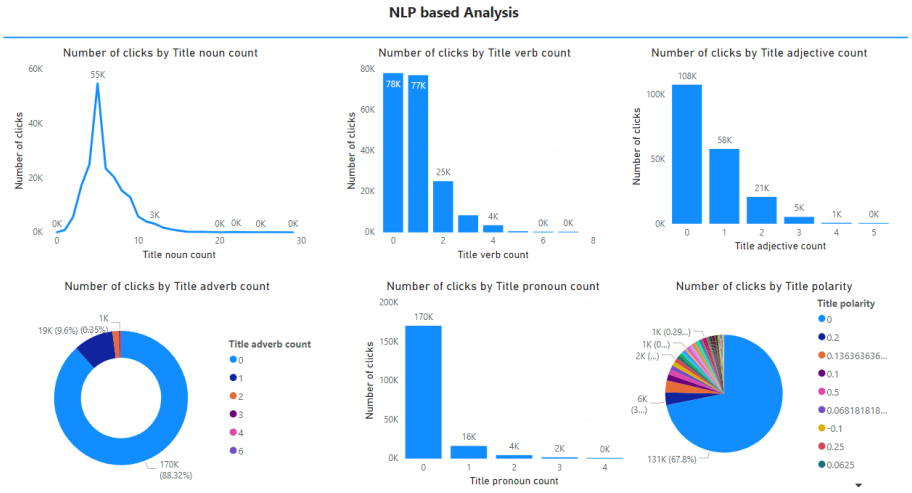

Figure 3: NLP Based Analysis - POS

On observation of 'noun count' and 'number of clicks', we can infer that clicks have been increasing up to a certain point. noun count of five has the highest number of clicks and it gradually decreases with an increase in title length.

From the analysis of 'verb count' and 'number of clicks' we came to know that title containing verb count of zero and one have the highest number clicks of 78,000 and 77,000 respectively.

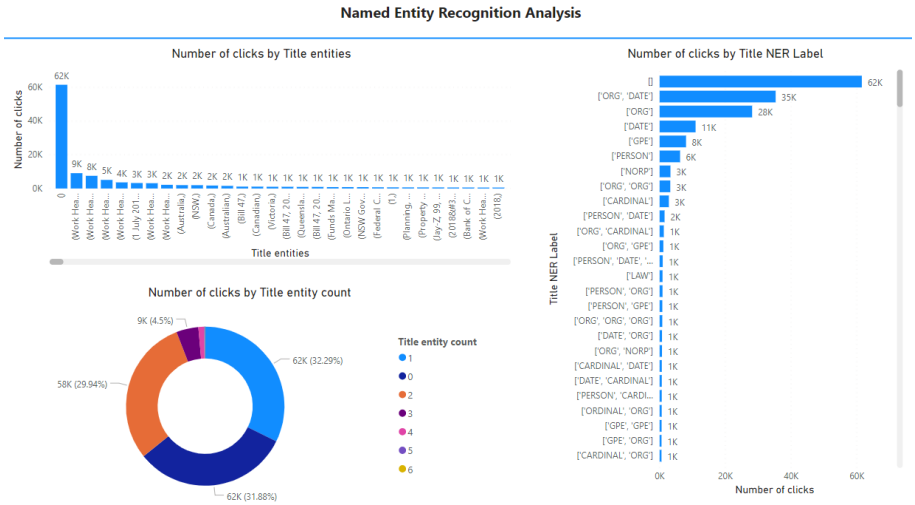

Figure 4: NLPBasedAnalysis - NER

Observation of 'adjective count' and 'number of clicks' shows us the highest number of clicks happened for an adjective count of zero. Also as the adjective count is increasing click rate has been decreasing.

By looking into 'adverb count' and 'number of clicks' we can say that a high number of clicks have happened for the titles having zero adverbs. It is evident that 88 percent of clicks are registered for the titles with zero adverbs.

The pattern from 'pronoun count' and 'number of clicks' infers titles without Pronouns have a higher click rate. The more the Pronouns less the click rate.

Observing sentiments of text 'polarity' and 'number of Clicks', we can say that 67 percent of the clicks are registered with titles with zero polarity. Users may be not interested in looking sentimental words. The neutral title may catch the user's attention. Similarly looking into 'subjectivity' and 'number of clicks', the highest number of clicks are registered for zero subjectivity which is 57 percent.

From an analysis of 'named entities' and 'number of clicks', we can infer that titles with zero and one Entity labels have the highest number of clicks which is 32 percent each.

By concluding the NLP based analysis it is recommended that article titles with moderate nouns, avoiding adjectives, adverbs and pronouns, minimum usage of verbs and entities, and finally posing neutral sentiment titles can cause the high contribution to user engagement.

3.5 Data Preparation for modelling

The data used for analysis in earlier stages contains day-wise clicks. Grouping data with respect to the title and sum of no of clicks gives us final data for

modelling and this data has been converted to three user engagement classes of a low, medium, and high. As the user's engagement with an article likely to vary according to topic popularity, dissection of clicks into classes has been performed based on clicks quantiles by considering topic description.

Upon exploration of the data,it is found that data contains categorical variables which needs to be encoded and title is a text data which needs preprocessing and featurization.

3.5.1 Encoding Categorical Variables

Encoding is a technique of converting categorical variables into numbers. As machine learning models involve a lot of mathematical calculations, direct feeding of categorical variables to machine maybe results in misinterpretation. Categorical features are widely used in both classification and regression problems. However, most of the machine learning algorithms accept only numeric values as input. To use these categorical data for machine learning purposes, the data needs to be encoded into numeric values such that each categorical variable is represented with a number. Data contains two categorical variables which are country description and topic description. One hot encoding has been used to convert these categories to numbers[38,39].

Both country description and topic description are nominal categorical variables where there are no intrinsic ordering to its categories. Upon exploration, it is found that country description variables contain two categories and the topic description variable contains ten categories.

One Hot Coding is the coding scheme for nominal categorical variables that is commonly used. It compares a fixed reference level for each level of the categorical variable. This technique converts a single variable to d binary variables with n observations which indicates the presence (1) or absence (0) of the binary categorical variable and finally d distinct values.

This encoding methodology for country description and topic description involves the Usage of Pandas get dummies method which takes categorical variables as input and returns the dummy variables with numbers. It has been implemented by dropping one of the columns to avoid the dummy variable trap. Furthermore, these dummy variables are concatenated back to the original dataset.

3.5.2 Text Preprocessing

Since machine won't understand any language's sentences, we need to clean the dataset by removing stopwords, punctuation, and many more irrelevant things inside the data, and we need to make it up to that point where we can feed those data into our computer or deep learning algorithms from which we can get some performance. This process is referred to as Text preprocessing.[38,39] The potential of the natural language toolkit library has been used for this purpose. A text preprocessor has been created as a function where it will take each line of title as input and returns cleaned text. Initially, the stage of pipeline

contains a substitution of all the characters with space apart from alphabets 'a-z and A-Z' which results in the elimination of extra characters, punctuations, numbers. We remove punctuations because we don't have a different form of the same word. It follows text lowercase to reduce the size of the vocabulary. The next stage of pre-processor is the removal of stopwords. Stopwords are words that don't add to the sentence's meaning. Therefore they can be removed safely without causing any alteration in a sentence's context. The Natural Language Toolkit(NLTK) library has the collection of stopwords and we can use them to delete stopwords and return a list of word tokens from our text. The final stage of text pre-processor is Stemming. Stemming is a process of getting the root form of a word. Root or Stem is the part to which inflectional affixes(like -ed, -ize, etc) are added. We would create the stem words by removing the prefix or suffix of a word. As our sentences are not in tokens, we need to convert them into tokens. After we converted strings of text into tokens, then we can convert those word tokens into their root form using porter stemmer. These lists of root words are joined and appended to a new list which is the output of text pre-processor.

3.5.3 Text Featurization using Bag Of Words

The next challenge in the dataset is to transform the title into corresponding numerical vectors, which is termed as featurization. The whole potential of linear algebra can be leveraged by converting text to numerical vectors. The choice of the featurization technique is crucial because of the reason that model accuracy is directly related to the technique used.

Bag of words (BOW) is one of the simplest techniques to convert textual data to numerical vectors.A text is described in this BOWas an unordered set of its words, disregarding word ordering and grammar, and A word in a document is assigned a weight in the document according to its frequency and the frequency between the various documents[38,39].

In terms of implementation, 'count vectorizer' has been imported from the sklearn feature extraction module into a variable. The cleaned text has been passed as input to this model which returns the BOW matrix with corresponding numerical vectors. Furthermore, a new dataset has been created by concatenating these BOW vectors with original dataset.

3.5.4 Text Featurization using Term frequency-Inverse document frequency

TFIDF (Term Frequency Inverse Document Frequency), a commonly used weighting technique for information retrieval and information exploration.It is a statistical method mostly used to evaluate the importance of a word in respective text. The importance of the word increases in proportion to the number of times it appears in the file, but at the same time decreases inversely with the frequency of its appearance in the corpus.Term frequency TF refers to number of times a given word appears in the text. This number is usually normalized.

As the numerator is generally smaller than the denominator, to prevent it from favoring long documents, because whether the term is important or not, it is likely to appear more often in long documents than in paragraph documents, hecnce forth introduced ab another term of Inverse document frequency which is a measure of the general importance of a word. The main idea is that if there are fewer documents containing the entry , it means that the entry has a good ability to distinguish categories. The IDF of a specific word can be calculated by dividing the total number of files by the number of files containing the word, and then taking the log of the obtained quotient.By multiplying TF with IDF the greater or higher occurrence of a word in documents will give higher term frequency and the less occurrence of a word in documents will yield higher importance (IDF)[38,39].

In terms of implementation, 'Tdidf vectorizer' has been imported from the sklearn feature extraction module into a variable. The cleaned text has been passed as input to this model which returns the TFIDF matrix with corresponding numerical vectors. Furthermore, a new dataset has been created by concatenating these TFIDF vectors with the original dataset.

3.5.5 Text Featurization using Word Embeddings

Word embeddings are the high dimensional distributed vector representations of text. In Word embedding, each word is represented as a real-valued vector[38,39].

Word embedding has been implemented using Keras embedding layers. In Keras, the Embedding layer requires two parameters, one is the number of words in the token, and the other is the embedded dimension. It has been integrated as an input layer during the training of Neural Network.

The outcome of the featurization section gives two datasets named 'data BOW' and 'data TFIDF'.The main difference between them is feature representation of title. 'data BOW' uses Bag of words representation and 'data TFIDF' uses Term frequency-Inverse document representation.

3.6 Modelling

This section comprises data normalization, generating training and test datasets and finally Model Training. All the steps have been performed on both 'data BOW' and 'data TFIDF'.

3.6.1 Data Normalization

Data Normalization is a preprocessing technique that transforms each variable to a range between 1 and 0 usually employed before machine learning modelling.

It has been implemented using MinMaxScalar extracted from the sklearn preprocessing module. The raw training data has been fed as input to this model which returns normalized training data as output. The minimum value

of the variable will be transformed as 0, the maximum value will be transformed as 1 and other values in between 0 and 1.

3.6.2 Training and Test datasets

The data set has been split into training and testing data. The training dataset is used to make a model understand the patterns, whereas the test dataset has been used to evaluate the performance of the model.

It has been implemented using the 'train test split' model extracted from the sklearn preprocessing module. Normalized data has been fed as input to this model which returns train and test dataset with test data set of size 33 percent.

3.6.3 Model Training using Naïve bayes

The main process of modeling involves training suitable machine learning and deep learning models using a training dataset in order to predict output labels (or) classes. The choice of models plays a crucial role in achievement objectives. Henceforth several models who can work well with high dimensional data have been chosen,training, and evaluated.

Naïve Bayes is a supervised classification technique based on the fundamentals of probabilities.Henceforth it is often called as probabilty based classifier. The reason for choosing Naïve Bayes is that the popularity of this model in text classification applications which are high dimensional data. In fact, it is the first technique used to build a spam filter. The first fundamental idea in Naïve based is the idea of conditional probability. Conditional Probability is often written as

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}, \tag{1}$$

Given A and B are random variables and the probability of B not equal to zero probability of Event A conditioned on the fact of B already happened or given is equal to the probability of both A and B happened divided by probability of B happened.

Second Fundamental idea in Naïve Bayes is Bayes theorem which is a simple, elegant and most useful theorem.

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{2}$$

Where A and B are events and Probability of B is not equal to zero

• P(A given B) is referred to as Posterior which is a conditional probability defines the likelihood of event A happening given that B is true

• P(B given A) is referred to as likelihood which is a conditional probability defines the likelihood of event B happening given that A is true

• P(A) and P(B) are the probabilities of observing A and B referred to as Prior and evidence respectively

By using these two fundamentals of conditional independence assumption on Bayes theorem, the Naïve Bayes algorithm became an unsophisticated or simplistic algorithm. It works differently from other classification algorithms of the fact that it does not first try to learn how to classify the points. It directly uses the label to identify the two separate classes and then it predicts the class to which the new point shall belong.

During the training phase, the Naïve base calculates all the likelihoods based on training data for all the features as a first step. The next step is to compute the probabilities of each class or response variable which are called priors. These likelihoods and priors are tabulated or stored as part of the learning phase. During the testing phase, whenever a new instance has occurred, the stored priors and likelihoods in the training phase are used with the Maximum a Posterior rule to classify test points to the right instance.

In the case of python implementation, necessary packages are imported form modules. GaussianNB has been imported from the sklearn naïve Bayes module into the model variable. The model has been trained by fitting normalized training data with the optimum hyperparameters tuned from gridsearchcv. This phase model tries to stores likelihoods and priors from training data. Upon successful training, the same model has been used to predict the output classes for the test dataset.

3.6.4 Model Training using Support Vector Machine

Support Vector Machine also referred to as SVM is a very popular supervised technique that can be used for both classification and regression problems which became extremely popular in the late 1990s.It is one of the simple and elegant algorithms which tries to separate the data using hyperplanes and then makes predictions. SVM is a non-probabilistic classifier which directly says to which group the datapoint belongs to without using any probability calculation.

For any classification problem of linear separable positive and negative points, multiple hyperplanes can be used to separate positive points and negative points. A hyperplane is an n-1 dimensional plane that optimally divides the data of n dimensions. Choosing the right hyperplane will always a challenge because points close to hyperplane may result in misclassification based on a slight change in the position of the hyperplane. The key idea of SVM is to find a hyperplane that separates positive points from negative points as widely as possible. Such a place is referred to as margin maximizing hyperplane.

The essentials elements in an SVM are support vectors which are points through which positive hyperplane or negative hyperplane passes. The algorithm creates the optimum classification line by maximizing its distance from the two support vectors.

When the data is not linearly separable, then to create a hyperplane to separate data into different groups, the SVM algorithm needs to perform computations in a higher-dimensional space. But the introduction of new dimensions makes the computations for the SVMs more intensive, which impacts the algorithm performance. To rectify this, mathematicians came up with the ap-

proach of Kernel methods. Kernel methods use kernel functions available in mathematics. The unique feature of a kernel function is to compute in a higher-dimensional space without calculating the new coordinates in that higher dimension. It implicitly uses predefined mathematical functions to do operations on the existing points which mimic the computation in a higher-dimensional space without adding to the computation cost as they are not calculating the coordinates in the higher dimension thereby avoiding the computation of calculating distances from the newly computed points. This is called the kernel trick[38,39].

Every machine learning model contains a certain number of hyperparameters which contributes directly to the model predictive performance. The values of hypermeters are always determined by specific leaning patterns in the training dataset. The process of finding the right value for the hyperparameter on a given training dataset is known as hyperparameter tuning.

One of the methods to tune the model's hyperparameters is GridSearchCV. Different values of hyperparameters are passed as grid search parameters. It then generates all combinations of parameters passed and returns the best models hyperparameters using training data and cross-validation score.

In the case of python implementation, necessary packages are imported form modules. SVC has been imported from the sklearn SVM module into the model variable. The model has been trained using normalized training data with the optimum hyperparameters tuned from gridsearchcv. In this phase, the model tries to learn patterns in data. Upon successful training, the same model has been used to predict the output classes for the test dataset.

3.6.5 Model Training using Random Forest

Random Forest is a supervised ensembling technique that can be used for both Classification and Regression tasks. Random Forest uses Decision Trees as base learners and then applying bagging and column sampling on top of it. Decision trees are such models having low bias and high variance which tend to overfit the data. Bagging is one of the technique which reduces the variance in decision trees. However, bagging results in correlation among the sampled dataset. Henceforth, introducing some randomness in the sampled data through column sampling leads to the Random Forest. The advantage of Random Forest is that Random Forest makes a tweak as column sampling in the working principle of bagging to reduce the correlation effect[38,39].

The workflow of the Random Forest can be explained as follows: Different training dataset samples are collected using bootstrapping and column sampling

Train the tree models using each sample up to high depths Upon formulation of trees, the Random Forest makes the predictions by aggregating all the model outputs. The choice of aggregation functions depends upon problem type. That will be a mean aggregate function in case of regression or a mode aggregation function in case of the Classification task. This process is referred to as majority voting[38,39].

The hyperparameters in the Random Forest are a combination of hyperparameters of both Bagging model and Decision Tree which are 'n estimators', a criterion which measures the quality of split, 'max depth' defines maximum depth of the tree, 'min samplesleaf' defines minimum samples count required at a leaf node, 'min samples split' refers to a minimum number of samples required to split an internal node and 'max features' defines a number of features to consider when looking for the best split. In the case of python implementation, necessary packages are imported form modules. RandomForestClassifier has been imported from the sklearn module into a model variable. The model has been trained using normalized training data with the optimum hyperparameters tuned from gridsearchcv. In this phase, the model creates multiple sample datasets using bagging and feature sampling. Trees try to learn patterns in each sample data. Upon successful training, the same model has been used to predict the output classes for test datasets using aggregate functions.

3.6.6 Model Training using XGBoost

XGBoost also referred to as extreme gradient boosting is a supervised technique that can be used for both Classification and Regression tasks. It is an implementation gradient boosted Trees, row sampling, and column sampling. As a result data regularisation will be better than gradient boosted Trees. It was developed by Tianqi Chen from the University of Washington. XGBoost is a tree boosting system that is used extensively by AI Engineers and data scientists to achieve state-of-the-art results on many data science challenges[38,39].

The building blocks of XGBoost are gradient boosted Trees which uses use decision trees as estimators. The main theme of boosting techniques is to enhance the performance of week learners by starting from a weaker decision and keeps on building the models such that the final prediction is the weighted sum of all the weaker decision-makers. The weights are assigned based on the performance of an individual tree. Unlike the other tree-building algorithms, XGBoost doesn't use entropy or Gini indices. Instead, it utilizes gradient as the error term and hessian for creating the trees. The hyperparameters involved in XGBoost are 'learning rate','max depth', and 'n estimators'[38,39].

In the case of python implementation, XGBoost is an open-source package that has been installed and imported as xgb into a model variable. The model has been trained using normalized training data with the optimum hyperparameters tuned from gridsearchcv. In this phase, the model tries to learn patterns in data. Upon successful training, the same model has been used to predict the output classes for the test dataset.

3.6.7 Model Training using Stacking Techniques

Stacking techniques is a supervised technique that can be used for both Classification and Regression tasks. It is a type of ensemble technique which combines the predictions of two or more base models and use the combination as the input for a new model which is a meta-model trained on the predictions of the base

models[38,39].

we are training a classification problem with several base models like Random Forest, Naïve Bayes, Support Vector Machine. The main theme is to use a few models such as Naïve Bayes and Support vector machines as base models and finding predictions from these models. Furthermore, these predictions are used as an input for a meta-model which is a Random Forest, in this case, to train and give predictions. It results in an improvement inaccuracy.

In the case of python implementation, necessary packages are imported form modules. StackingClassifier has been imported from the mlxtend module into a model variable. The model has been trained using Naive bayes and Support vector classifier as base models and Random Forest classifier as a meta classifier by normalized training data with the optimum hyperparameters tuned from gridsearchcv. In this phase, the model tries to learn patterns in data. Upon successful training, the same model has been used to predict the output classes for the test dataset.

3.6.8 Model Training using Artificial Neural Networks

Artificial Neural Networks also referred to as ANN is a part of deep learning which can be used for both classification and regression tasks. The deep refers to the depth of the network. The structure of the cerebral cortex is the inspiration of Neural Networks. The representation of biological neurons is perceptron. Similar to the cerebral cortex, there are multiple interconnected perceptrons layers. These are the input layer which is the first layer responsible for collecting input data, Second layer in the middle or hidden layer which will be activated by the input layers. Final layers are output layers that are responsible to generate models output[38,39].

The main components of an ANN are weights and activation functions. Weights are learnable parameters that explain the influence of outputs on inputs. The overall objective of ANN is to find these optimal weights in the training process. The second component of an ANN is Activation function which is responsible for the operation of activation and deactivation of neurons. Some of the widely used activation functions are Sigmoid, Tanh, ReLU, and LeakyRelu.

The operation of ANN can be explained as a two-stage process which is forward propagation and backward propagation. The first stage of forward propagation contains a prediction of output based on given inputs and estimated weights. The second stage of backward propagation plays a crucial role as the networks try to learn patterns associated with training data. At the beginning of backward propagation, errors are calculated from the actual output and forward propagation predicted output using loss functions. The main aim of the backward propagation is to reduce this error by adjusting assigned network weight parameters using optimization functions such as Gradient descent. This function tries for effective calculation of the gradient of the cost function with respect to its network weights. One complete cycle of the forward pass and a backward pass is termed as an epoch. The process continues until the network converges[38,39].

It is important to be aware that gradient descent is a descending algorithm, it is liable to be caught in local minima with respect to starting values. Therefore, it is worthwhile training several networks using a range of starting values for the weights, so that you have a better chance of discovering a globally-competitive solution[38,39].

One useful performance enhancement for the learning algorithm is the addition of momentum to the weight updates. This is just a coefficient on the previous weight update that increases the correlation between the current weight and the weight after the next update. This is particularly useful for complex models, where falling into local minima is an issue; adding momentum will give some weight to the previous direction, making the resulting weights essentially a weighted average of the two directions. Adding momentum, along with a smaller learning rate, usually results in a more stable algorithm with quicker convergence[38,39].

In the case of python implementation, Deep learning can be accessible by many open source projects. Sequential and dense has been imported from Keras models and layers classifier variable.Word embeddings has been incorporated as a internal layer which is responsible for converting text data to vectors. The model has been trained using normalized training data. In this phase, the model tries to learn patterns in data. Upon successful training, the same model has been used to predict the output classes for the test dataset.

4 Evaluation and Results

The credibility of the model will be known by evaluating it. There are various metrics to measure the performance of the model. One of the well-known methods is using a confusion matrix to find out how accurately the model predicted true negatives and true positives.

Confusion matrix contains true positives which are referred as a result that was predicted as positive by the model and also it is positive, true negative which is referred as a result that was predicted as negative by the classification model and also it is negative, false positive which is referred as a result that was predicted as positive by the classification model but it is negative and false Negative which is referred as a result that was predicted as negative by the classification model but it is positive.The different performance metrics derived from Confusion matrix are

• Accuracy is defined as the total number of correctly classified points divided by the total number of classifications

• Precision is a measure of amongst all the positively predicted points, how many of them were positive points

• The recall is a measure from the total number of positive results how many positive points were correctly predicted by the classification model. The relevance of the model is, in terms of positive results only

• F1 Score considers both precision and recall which is defined as the harmonic mean of Precision and Recall.

Model	Accuracy	Precision	Recall	F-Score
Naïve bayes BOW	44.84	45.05	44.79	44.81
Naïve bayes TfIdf	45.75	45.69	45.68	45.66
SVM BOW	39.08	44.92	38.82	31.07
SVM TfIdf	44.24	47.38	44.21	43.89
Random Forest BOW	39.99	47.96	39.68	30.12
Random Forest TfIdf	45.33	47.26	45.26	44.39
XG boost BOW	45.15	45.69	45.07	44.92
XG boost TfIdf	45.45	45.91	45.39	45.15
StackingClassifier BOW	43.45	44.98	43.49	43.22
StackingClassifier TfIdf	44.90	45.73	44.92	44.90
Artificial Neural Network Word Embeddings	45.33	47.26	45.26	44.39

Table 9: Performance Metrics

Table 9 shows performances of models trained on 'data BOW', 'data TFIDF' and 'word embeddings'.On observation of model performance. Our model evaluation results highlight that the best method turned out to be the Naïve Bayes modeling using the TfIDF vectorization technique achieving an accuracy of 45.75 percentage.

Naive Bayes TfIDF	Precision	Recall	F1score
Class 0	0.47	0.50	0.49
Class 1	0.39	0.38	0.39
Class 2	0.51	0.48	0.50

Table 10: Precison and Recall Interpretation

The precision of the model Naive Bayes can be inferred as of all the points where model predicted as class 0, 47 percent are actually class 0. Similarly recall of model Naive Bayes can be inferred as of all the points that originally belong to class 2, 48 percent were classified as class 2 shown in Table 10.

5 Discussions

Our paper presents a two-stage approach to analyze and predict user engagement based on article titles. Primarily, exploratory data analysis to understand data patterns and user insights. The findings in this phase showed that the 'Employment and HR topic' has the highest number of clicks of 74,000. Users are more interested in knowing about human resource management and job analytics. The secondary User interest topic is real estate and construction with 24,000 number of clicks. The data also showed that titles containing stopwords unable to engage users. Users may be not interested in looking at common

words. Henceforth it is recommended to maintain some content words. Observing sentiments of text 'polarity' and 'number of Clicks', it is found that 67 percent of the clicks are registered with titles having zero polarity. Users may be not interested in looking sentimental words. The neutral title may catch the user's attention. Unlike the study by Ball (2009)[2] that articles with question-type titles are viewed and downloaded more, our study showed that 91 percent of the clicks are registered with titles without Question words. Henceforth interrogative titles may not catch the user's attention.Declarative or descriptive titles are preferred and this might be expected as declarative titles are normally those with clear findings and in that regard more attractive. Our findings lend partial support to those of Yitzhaki (1994, 2002)[35, 37] in that they show articles with longer titles are more likely to be attractive. On observation of 'length' vs 'number of clicks', we can infer that till a certain point title length increases with respect to length. The highest number of clicks are registered at a length of 42 and it gradually decreases with an increase in title length. It seems User's are preferring moderate title length.

Secondly, We have framed prediction as a Classification approach with three user engagement levels of high, moderate, and low. In terms of modeling, we have presented a comparative analysis of the success rates of state-of-the-art prediction algorithms, which are Naïve Bayes, Random Forest, Support Vector Classification, eXtreme Gradient Boosting algorithms, stacking models, and finally Artificial Neural Networks. These State of Art Machine learning models have been trained using Bag of words and TFIDF vectorization techniques whereas artificial neural networks used word embeddings as a layer to convert Text to vectors. The study reveals significant differences between the different methods. The best two methods turned out to be the Naïve Bayes modeling using the TFIDF vectorization technique and the Extreme Gradient Boosting algorithm with the TFIDF vectorization technique. In addition to model comparisons, we have tested different vectorization methods and have shown that they have different impacts on performance.

It is essential to make clear that all findings we presented in this paper relate to 5000 Mondaq Articles with a published duration of 2 weeks in the countries of Australia and Canada and findings may not be correct for other countries and other businesses. Moreover, we can also further improve the accuracy of the models by incorporating advanced Natural language processing techniques such as Attentive based models, Recurrent Neural Networks, RNN with Long short-term memory, and finally State of art Bert vectorization techniques. These are exactly our future work.

6 Conclusion

This paper aims to explore the assumption that the formulation of article titles contributes significantly to the level of user interaction. We have adopted a two-stage method, beginning with an exploratory data analysis that focuses on data properties and user engagement insights and a supervised three-state

classification that predicts user engagement levels based on article titles.Our text based analysis shows that article titles with moderate length, the word count of nine, at least one punctuation, lower casing, minimum stop words, and descriptive or declarative title can cause a high contribution to user engagement. Similarly, NLP based analysis shows that article titles with moderate nouns, avoiding adjectives, adverbs and pronouns, minimum usage of verbs and Entities, and finally posing neutral sentiment titles can engage users. On the other hand, Our model evaluation results highlight that the best method turned out to be the Naïve Bayes modeling using the TfIDF vectorization technique achieving an accuracy of 45.75 percentage. This research could aid News syndication platforms in understanding how article titles affect user engagement. Exploratory data analysis contribute to making better decisions during article title creation.

References

[1] Kholoud Khalil Aldous, Jisun An, and Bernard J Jansen. View, like, comment, post: Analyzing user engagement by topic at 4 levels across 5 social media platforms for 53 news organizations. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 47–57, 2019.

[2] Rafael Ball. Scholarly communication in transition: The use of question marks in the titles of scientific articles in medicine, life sciences and physics 1966–2005. *Scientometrics*, 79(3):667–679, 2009.

[3] Roja Bandari, Sitaram Asur, and Bernardo A Huberman. The pulse of news in social media: Forecasting popularity. *arXiv preprint arXiv:1202.0332*, 2012.

[4] Jonah Berger. Arousal increases social transmission of information. *Psychological science*, 22(7):891–893, 2011.

[5] Jonah Berger and Katherine L Milkman. What makes online content viral? *Journal of marketing research*, 49(2):192–205, 2012.

[6] Jonah Berger and Katy Milkman. Social transmission, emotion, and the virality of online content. *Wharton research paper*, 106:1–52, 2010.

[7] Deepayan Chakrabarti, Deepak Agarwal, and Vanja Josifovski. Contextual advertising by combining relevance with click feedback. In *Proceedings of the 17th international conference on World Wide Web*, pages 417–426, 2008.

[8] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 ieee/acm international conference on advances in social networks analysis and mining (asonam)*, pages 9–16. IEEE, 2016.

[9] Krzysztof Dembczynski, Wojciech Kotlowski, and Dawid Weiss. Predicting ads clickthrough rate with decision rules. In *Workshop on targeting and ranking in online advertising*, volume 2008, 2008.

[10] Mohammad Reza Falahati Qadimi Fumani, Marzieh Goltaji, Pardis Parto, et al. The impact of title length and punctuation marks on article citations. *Annals of Library and Information Studies (ALIS)*, 62(3):126–132, 2015.

[11] Zhipeng Fang, Kun Yue, Jixian Zhang, Dehai Zhang, and Weiyi Liu. Predicting click-through rates of new advertisements based on the bayesian network. *Mathematical problems in engineering*, 2014, 2014.

[12] Gonca Gürsun, Mark Crovella, and Ibrahim Matta. Describing and forecasting video access patterns. In *2011 proceedings IEEE infocom*, pages 16–20. IEEE, 2011.

[13] Daniel Hardt and Owen Rambow. Predicting user views in online news. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 7–12, 2017.

[14] James Hartley. Planning that title: Practices and preferences for titles with colons in academic articles. *Library & Information Science Research*, 29(4):553–568, 2007.

[15] Thomas S Jacques and Neil J Sebire. The impact of article titles on citation hits: an analysis of general and specialist medical journals. *JRSM short reports*, 1(1):1–5, 2010.

[16] Kokil Jaidka, Tanya Goyal, and Niyati Chhaya. Predicting email and article clickthroughs with domain-adaptive language models. In *Proceedings of the 10th ACM Conference on Web Science*, pages 177–184, 2018.

[17] Hamid R Jamali and Mahsa Nikzad. Article title type and its relation with the number of downloads and citations. *Scientometrics*, 88(2):653–661, 2011.

[18] S Kane Thomas. The oxford essential guide to writing, 2000.

[19] Yaser Keneshloo, Shuguang Wang, Eui-Hong Han, and Naren Ramakrishnan. Predicting the popularity of news articles. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 441–449. SIAM, 2016.

[20] Shoubin Kong, F Ye, and L Feng. Predicting future retweet counts in a microblog. *Journal of Computational Information Systems*, 10(4):1393–1404, 2014.

[21] Dmitry Lagun and Mounia Lalmas. Understanding user attention and engagement in online news reading. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 113–122, 2016.

[22] Sotiris Lamprinidis, Daniel Hardt, and Dirk Hovy. Predicting news headline popularity with syntactic and semantic knowledge using multi-task learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 659–664, 2018.

[23] Jong Gun Lee, Sue Moon, and Kavé Salamatian. Modeling and predicting the popularity of online contents with cox proportional hazard regression model. *Neurocomputing*, 76(1):134–145, 2012.

[24] Grant Lewison and James Hartley. What's in a title? numbers of words and the presence of colons. *Scientometrics*, 63(2):341–356, 2005.

[25] AA Manten and JFD Greenhalgh. s of scientific papers, 1977.

[26] Luís Marujo, Miguel Bugalho, João Paulo da Silva Neto, Anatole Gershman, and Jaime Carbonell. Hourly traffic prediction of news stories. *arXiv preprint arXiv:1306.4608*, 2013.

[27] Moira Regelson and D Fain. Predicting click-through rate using keyword clusters. In *Proceedings of the Second Workshop on Sponsored Search Auctions*, volume 9623, pages 1–6. Citeseer, 2006.

[28] Julio Reis, Fabrıcio Benevenuto, Pedro OS de Melo, Raquel Prates, Haewoon Kwak, and Jisun An. Breaking the news: First impressions matter on online news. *arXiv preprint arXiv:1503.07921*, 2015.

[29] Itay Sagi and Eldad Yechiam. Amusing titles in scientific journals and article citation. *Journal of Information Science*, 34(5):680–687, 2008.

[30] Alexandru Tatar, Panayotis Antoniadis, Marcelo Dias De Amorim, and Serge Fdida. Ranking news articles based on popularity prediction. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 106–110. IEEE, 2012.

[31] Alexandru Tatar, Marcelo Dias De Amorim, Serge Fdida, and Panayotis Antoniadis. A survey on predicting the popularity of web content. *Journal of Internet Services and Applications*, 5(1):8, 2014.

[32] Manos Tsagkias, Wouter Weerkamp, and Maarten De Rijke. Predicting the volume of comments on online news stories. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1765–1768, 2009.

[33] Chieh-Jen Wang and Hsin-Hsi Chen. Learning user behaviors for advertisements click prediction. In *ACM SIGIR*, volume 11, pages 1–6, 2011.

[34] Chenyan Xiong, Taifeng Wang, Wenkui Ding, Yidong Shen, and Tie-Yan Liu. Relational click prediction for sponsored search. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 493–502, 2012.

[35] Moshe Yitzhaki. Relation of title length of journal articles to number of authors. *Scientometrics*, 30(1):321–332, 1994.

[36] Moshe Yitzhaki. Variation in informativity of titles of research papers in selected humanities journals: A comparative study. *Scientometrics*, 38(2):219–229, 1997.

[37] Moshe Yitzhaki. Relation of the title length of a journal article to the length of the article. *Scientometrics*, 54(3):435–447, 2002.

[38] Machine Learning and Deep Learning with Deployment from iNeuron.

[39] Applied Machine Learning Course from AppliedAI