**San Jose State University**

[**SJSU ScholarWorks**](http://scholarworks.sjsu.edu/?utm_source=scholarworks.sjsu.edu%2Fetd_projects%2F634&amp;utm_medium=PDF&amp;utm_campaign=PDFCoverPages)



[Master's Projects](http://scholarworks.sjsu.edu/etd_projects?utm_source=scholarworks.sjsu.edu%2Fetd_projects%2F634&amp;utm_medium=PDF&amp;utm_campaign=PDFCoverPages) [Master's Project and Graduate Research](http://scholarworks.sjsu.edu/etd?utm_source=scholarworks.sjsu.edu%2Fetd_projects%2F634&amp;utm_medium=PDF&amp;utm_campaign=PDFCoverPages)



Fall  2018

Detect Clickbait With Machine Learning Method and Deep Learning

Yee Zhian Liew

*San Jose State University*

Sowndhariya Nandarajkumar

*San Jose State University*

Saurabh Aggarwal

*San Jose State University*

Abhishek Prabhudesai

*San Jose State University*



Detect Clickbait with Machine Learning and Deep Learning

A Project Report Presented to Chandrasekar Vuppalapati

Department of Computer Science San Jose State University

In Partial Fulfillment

of the Requirements for the Class CMPE 256

By

Yee Zhian Liew- 013708240

Sowndhariya Nandarajkumar -013761410

Saurabh Aggarwal -013311649

Abhishek Prabhudesai-013736879

Dec. 2018

## ABSTRACT

As far as we know that bait means luring someone to do something. Click-baiting is to attracts readers attention to click it. Click-baiting is a growing phenomena on the internet, and it is defined as a method of exploiting cognitive biases to attract online viewership, that is, to attract “clicks.” In other words, these headlines contain text which leaves the reader curious about what the article contents might be, or they contain text about topics not really covered in the article itself. In order to avoid readers from clicking it, we attempt to detect the clickbait with our design model. In this paper, we are going to use 2 different dataset to understand the extent of clickbait practice, its impact and user engagement by using our own developed clickbait detection model. Moreover, we study the clickbait problem on YouTube by collecting metadata of youtube videos. To address it, we devise a deep learning model based on variational autoencoders that supports the diverse modalities of data that videos include. The proposed model relies on a limited amount of manually labeled data to classify a large corpus of unlabeled data.

# Contents

1. [INTRODUCTION](#_1t3h5sf) 6
   1. What is Clickbait?6
2. [BACKGROUND](#_2s8eyo1) 7

2.1. Text Sentiment Analysis 7

2.2. Dataset10

3. DATA PREPROCESSING12

3.1 Natural Language Processing ToolKit13

3.2 Tokenization13

3.3 Count and TFIDF vectorizer14

4. FEATURE REDUCTION METHOD14

4.1 Feature Selection 15

4.2 Feature Extraction 15

5. CLASSIFICATION APPROACHES16

5.1 [Machine learning approach](#_2jxsxqh) 16

5.2 Deep Learning approach17

6. GRAPH AND VISUALIZATION18

6.1 [Data Modeling 1](#_32hioqz)8

6.2 [Data Visualization and Presentation 1](#_1hmsyys)9

6.3 Coding 26

7. RELATED WORK 41

8. [CONCLUSION](#_2u6wntf) 43

9. [FUTURE WORKS](#_19c6y18) 43

10. [BIBLIOGRAPHY](#_3tbugp1) 44

**TABLE OF FIGURES:**

Figure 1: KDD process 8

Figure 2: Sentiment Analysis Process Flow 8

Figure 3: Snapshot of Non Clickbait Data 8

Figure 4: Snapshot of Clickbait Data 9

Figure 5: Snapshot of Clickbait and Non Clickbait Youtube Data 10

Figure 6: Snapshot of Non Clickbait Image Data 11

Figure 7: Snapshot of Clickbait Image Data 11

Figure 8: Snapshot of Website Data 12

Figure 9: Deep learning with text 12

Figure 10: LSTM Module 13

Figure 11: ANN Module 13

Figure 12: Exploratory Analysis using Random Forest 18

Figure 13: Exploratory Analysis using Decision Tree 19

Figure 14: Exploratory Analysis using KNN 20

Figure 15: Exploratory Analysis using Naive Bayes 21

Figure 16: Exploratory Analysis using SVM 21

Figure 17: Exploratory Analysis using ANN 22

Figure 18: Exploratory Analysis using LSTM 22

Figure 19: Comparison of 5 ML Algorithm 22

Figure 20: Result from all baseline and Neural Network model 22

Figure 21: dislike distribution23

Figure 22: view distribution23

Figure 23: comment distribution24

Figure 24: like distribution24

Figure 25: top 30 tokens in clickbait25

Figure 26: SVM result on youtube dataset25

Figure 27: Deep learning accuracy on Image dataset 26

## INTRODUCTION

## What is Clickbait?

The basic trick of how the publisher gets reader’s attention is creating a very interesting headlines. Without curiosity, readers will not border to see the content of the web. However, this bait often has a very bad content and leaves the readers disappointed. According to Cambridge Dictionary, clickbait is define as [articles](https://dictionary.cambridge.org/us/dictionary/english/article), [photographs](https://dictionary.cambridge.org/us/dictionary/english/photograph) which is on the [internet](https://dictionary.cambridge.org/us/dictionary/english/internet) that are [intended](https://dictionary.cambridge.org/us/dictionary/english/intended) to [attract](https://dictionary.cambridge.org/us/dictionary/english/attract) readers’ [attention](https://dictionary.cambridge.org/us/dictionary/english/attention) and [encourage](https://dictionary.cambridge.org/us/dictionary/english/encourage) readers to [click](https://dictionary.cambridge.org/us/dictionary/english/click) on [links](https://dictionary.cambridge.org/us/dictionary/english/links) to [particular](https://dictionary.cambridge.org/us/dictionary/english/particular) [websites](https://dictionary.cambridge.org/us/dictionary/english/website). Example of the common clickbait title will be “Top 10 tricks will change your life”, “5 things that will change your life!!” and “She said she had HIV: What happens next will blow your mind!”. In order to detect click-bait, a supervising learning model can be created. Before creating a model, we need to have a decent amount of dataset.

We collected extensive dataset of clickbait and non clickbait from GitHub which contains 16,000 headlines.Beside, social media such as Youtube will contains clickbait headlines. Chances for a reader to click on clickbait header without noticing it is high. The main question is, why are some of the Youtube trying so hard to post interesting headlines which is not related to their content? The answer is simple: each of them trying to get reader’s attention which is views, likes, and shares. We refer this eye-catching thumbnails technique as clickbait. Detecting clickbait according to content seems like not good enough for traditional machine learning. We are scraping website features such as number of images to improve our model on predicting skills. So, we are including deep learning with LSTM model into it to get a better model accuracy.

## BACKGROUND

**2.1. Text Sentiment Analysis**

Sentiment analysis or opinion mining is one of the famous research areas in computer sciences. The application of sentiment analysis are very broad and powerful especially in text mining. It can also be an essential part of your market research and customer service approach. For example, Amazon used sentiment analysis to analyse the comment in terms of text. It is very powerful as it can detect even emojis whether it is positive or negative. Based on sentiment analysis, sales not only can see what people think of your own products or services, you can see what they think about your competitors. Sentiment analysis can be classified into three diverse as sentence level, document level, and entity-aspect level. In a sentence level, a supposition of specific sentence is considered as a priority for sentiment prediction. Whereas, document level is a more generalized feeling which considers the whole document for sentiment prediction. And if the focus is straightforwardly on the opinion itself then it can be termed as an entity-aspect level sentiment analysis. As databases is getting larger and more vast, without machine learning, it is very hard to extract the interesting or important information. Machine learning algorithm such as Naïve Bayes, Logic Regression, and Support vector machine for predicting class for sentiment problem. Model is train from the train dataset given and passing various of algorithm and calculating the model accuracy.

Classification accuracy is the major issue. This gives a motivation for acquiring a good classification precision picking great feature determination, preprocessing along with order procedures.Basic process of sentiment analysis is shown as table below. At first, we will gather dataset from various reliable website, web warehouse. Since, component might be unstructured and messy, we need to use data cleaning method to remove the noise. Choose or remove the features by using feature extraction and selection strategy. After done with extracting, we need to vectorize our dataset into 1 and 0 because machine learning only can understand integer type. At this time, we can apply our vectorized data into different model and choose the highest accuracy model choice.



*Figure 1: KDD process*



*Figure 2: Sentiment Analysis Process Flow*

**2.2** **Dataset**

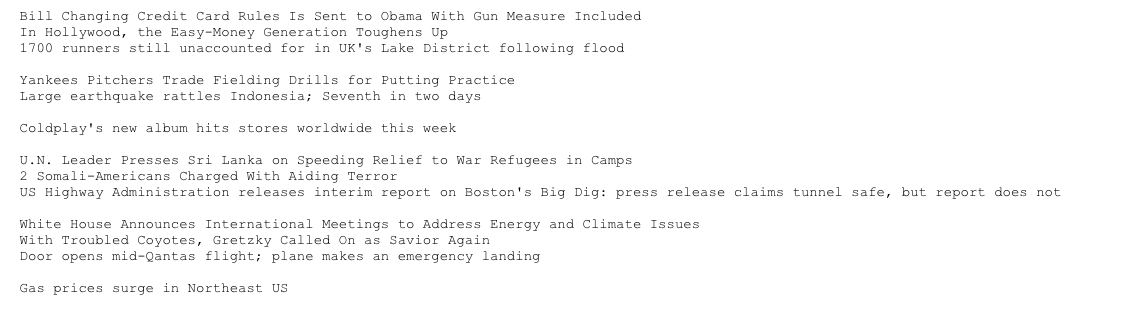
For our project, we have collected clickbait and non clickbait dataset from github which has 32000 rows. Each row contains clickbait and non clickbait headline title. Also, we have another module on youtube dataset. For youtube dataset, it contains 10,000 rows with 8 features. The 8 features such as number of likes, number of dislikes, number of comments, number of share.

Beside that, we have also included image classification module for our project. Color and pixel are the main features that used for detecting clickbait purposes.

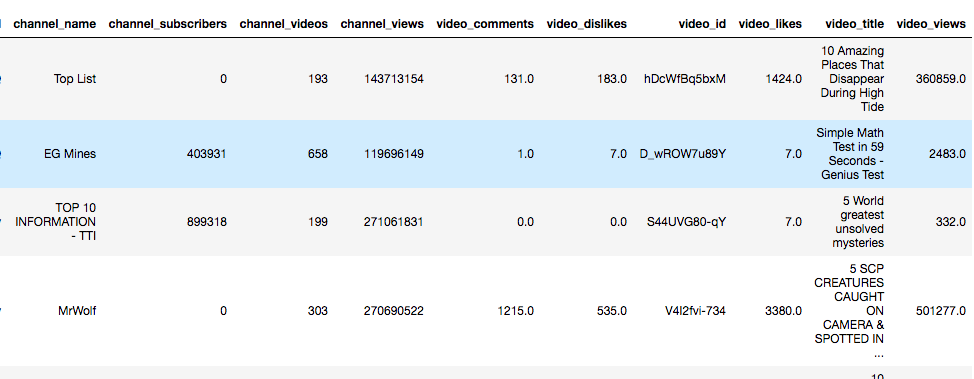
We have collected 2000 images from clickbait and non clickbait website to build our model. For last module, we are doing clickbait detection on formal website:buzzfeed and informal website: CNN news. We are using web scraping to scrap additional information from the website by using HTML scraper. We collected 3000 rows for website dataset and contains 8 features. Features such as number of H1, number of H2, H1 length, H2 length, number of image, number of tags, numbers of buttons. We split them into 2 different csv file: clickbait and non clickbait and start our evaluation.



*Figure 3: Snapshot of clickbait headlines data*



*Figure 4: Snapshot of Non clickbait headlines data*

**

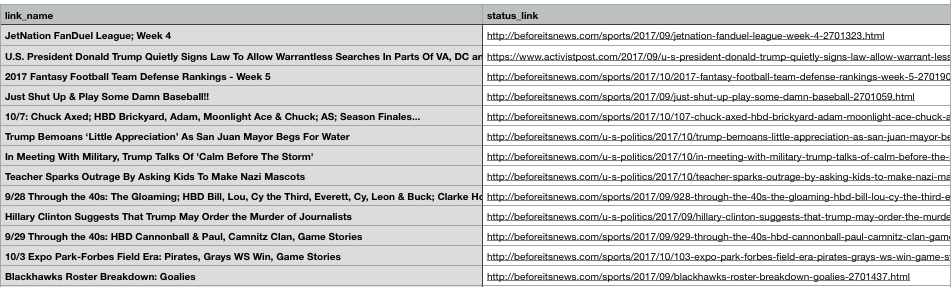
*Figure 5: Snapshot of Clickbait and Non Clickbait youtube data*

**

*Figure 6: Snapshot of Non Clickbait Image data*

**

*Figure 7: Snapshot of Clickbait Image data*



*Figure 8: Snapshot of Website Data*

**3. DATA PREPROCESSING**

**3.1 Natural Language Processing ToolKit**

Preprocessing is a the first and very important stage for text sentiment analysis. The very first cleaning method would be converting upper case word into lower case word. After that, non english word and stop word are basically cleaning steps. Natural language processing toolkit (NLTK) has contains a well-prepared document of stopwords and non english words to use for text cleaning. It is the most used tools for text mining problem. Then, unwanted punctuations or special character removal is likewise executed as a piece of preprocessing strategy. From this step, most of the unwanted words is done with cleaning, the next step would be stemming. Basically, stemming is removing word with suffixes, which ends with -ed, -ing, -ness. For example, ‘created” will convert into “create”. Word stem will reduce the number of unique word and thus reduce the dimensionality of data.

**3.2 Tokenization**

After cleaning up all the unwanted words, the last step of is tokenization. Tokenization is separates given text into tokens. For example: “I have a dream” will converting into “I”, “have”, “a”, “dream”. Each word in a text will serve as a single token. Natural language processing toolkit (NLTK) is utilized as a part of numerous existing papers using python to preprocess the dataset. NLTK is a python library for solving various text analytics and natural language processing tasks.

## 

## 

## 3.3 Count and TF-IDF Vectorizer

The next step is CountVectorizer. As the name implies, count vectorizer’s job is to count the word frequencies. After CountVectorizer apply, the next step is TF-IDF. Term frequency-inverse document frequency(TF-IDF) is to evaluate how important the word in a document. TF-IDF method is to convert text representation of information into a Vector Space Model (VSM). The first step in modeling the document into a vector space is to create a dictionary of terms present in documents. To do that, we can simple select all words from the document and convert it to a dimension in the vector space, but we know that there are some kind of words (stop words) that are present in almost all documents, and what we’re doing is extracting important features from documents, features do identify them among other similar documents.

## 4 Feature Reduction Methods

The two most basic techniques to reduce data dimensionality are:

**4.1 Feature Selection:** This method is also known as attribute and variable selection. Some of the famous method will be Wrapper method and Embedded Method. For dataset which is collected from World Wide Web, we can get variety of features. It is depending on your problem and to consider which features to utilize. Thousands of features is computed and we need to verify relationship between each variables. Excessive numbers of features might causing the algorithm do not function well.

There are numbers of existing features selection algorithms which used to convert a bigger dimension dataset into manageable size of dataset. Big advantages of feature selection is it can reduce the model complexity and evaluation steps will be easier. Moreover, it can used to improve model accuracy and reduces overfitting/underfitting problems. Without a great feature selection from data scientist, there is no big different from the result. Feature selection is one of the most important stage as it will directly affects our output.

**4.2 Feature Extraction:** While feature selection is maintaining the original features, feature extraction is using original features to create a brand new features. Data can be in different form of type such as ordinal, binary, continuous and categorical. We often use this “raw” data and convert it into something interesting.

As more new features is created, the data dimension will increase gradually. This is what we call as “curse of dimensionality”. The basic idea is to reduce the dimension of data in order for algorithm to be applied smoothly. It is a hard task for algorithm to process such high dimensional space and most likely will affects the model accuracy. The main approach for this problem will be Principal Component Analysis(PCA). PCA is one of the common strategy that creates linear combination to reduce data dimensional. Its point is to take in a discriminative transformation matrix to lessen underlying component space making it to a lower dimensional, remembering the true objective to diminish the many-sided quality of the grouping errand with no exchange off in terms of the accuracy of the model. Eigenvectors and Eigenvalues are the mathematical reason behind the transformation of the dimensions.

## 5. Classification approaches

**5.1** **Machine learning approach**

Machine learning and recurrent neural network deep learning are two the types of classification approaches. There are mainly 2 types of ML algorithm: Supervised learning and Unsupervised learning. Supervised learning means that the data is given a classify ‘class’ for it to refer and predict the output for other unseen data. Some of the useful supervised learning algorithm will be Support Vector Machine and Naive Bayes classifier. For Unsupervised Learning, the dataset does not contains any class label for the model to refer. In short, the algorithm is trying to find the interesting pattern itself according to what and numbers of features are given. Usually, unsupervised learning is related to clustering and association problem. Since we are dealing with classification problem, we are using algorithm such as random forest, logistic regression, k nearest neighbor to fit and test our dataset.

## 5.2 Deep Learning approach

Deep Learning approach or Deep analysis learning is a every high level machine learning to solve either supervised, semi-supervised or unsupervised problem. Recurrent Neural Network (RNN) is a class of AI which connects a bunch of neurons in a sequence. The basic idea is feeding it some input data, it will start processing using expensive algorithm to find out its predicted output. What RNN is different from other traditional machine learning is that, it can detect the hidden layer in the dataset. This is increase the accuracy of model effectively since more data is feeding in.

RNN is very useful in finding solution for sequences dataset such as video frame or image data. A very famous use cases would be word prediction. It is referring to the first input layer, then second and hidden layer and so on to predict what should be the next word. In every neurons layer, it contains input and its own specific weight value, which we call it as parameter. Weight is constantly updating to be larger or smaller when passing by gradient back-propagation phase. When the weight matrix is smaller than 1.0, it will causing the gradient signal become weak and learning model will either slow down or stop. Conversely, with larger weight matrix will cause high gradient signal and model learning path will be diverge. Thus, vanishing gradient problem and exploding gradient problem are the 2 main problems for RNN.

This is where Long Short Term Memory(LSTM) is introducing into it. LSTM is a unit of Recurrent Neural Network which can be added into the model to deal with the 2 main difficulty. Behind LSTM model it has a memory cell structure. Basically, this cell contains 4 very crucial components: input gate, a neuron with a self-recurrent connection, forget gate and output gate. Self-recurrent connection neuron is to maintain the weight matrix to be 1 as this point. Every gates will be having interaction with the cell. For input gate, it will choose or block the input signal into the respective neurons in each state. Forget gate will have the power to adjust the cell to remember or forget the previous state. At last, the output gate can allow or block the input signal to have effect on the neuron in the next state.



*Figure 9: Deep Learning with text*

****

*Figure 10: LSTM module*



*Figure 11: ANN module*

**6. GRAPH AND VISUALIZATION**

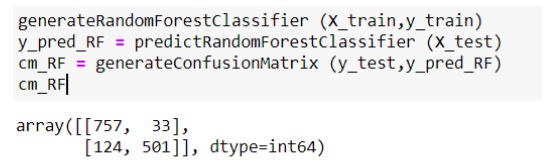
**6.1** [**Data Modeling**](#_32hioqz)

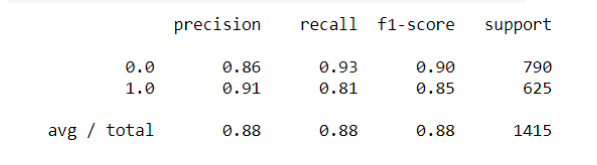
Machine learning algorithms: The project implements a Naïve Bayes machine learning algorithm. Naïve Bayes is a simple probabilistic classifier that works best when the features are independent of each other. It is a very popular method of text classification and uses word frequency as its features. In the current set of problem, for determining a sentiment of clickbait or non clickbait sentiment, the Naïve Bayes is considered as one of the best algorithm that can be used for a binomial classification problem.

**6.2** [**Data Visualization and Presentation**](#_1hmsyys)

In the current project, a confusion matrix are utilized to compare the results. A confusion metrics is a visualization tool that shows performances of the classification model.

## Analysis of the Random Forest results:

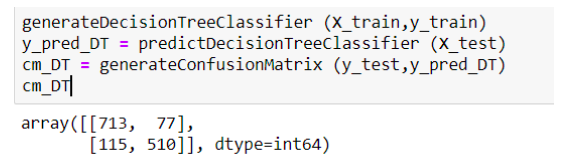


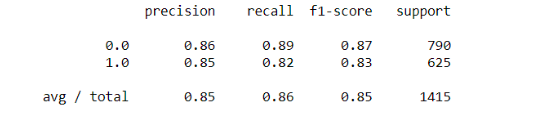


*Figure 12: Exploratory Analysis using Random Forest*

According to the Random Forest confusion matrix as shown above, the model predicted 757 correctly as True positive whereas predicted 33 wrong as False positive. Also, the model predicted 501 as True negative but it predicted 124 wrong as False negative.

## Analysis of the Decision Tree results:





*Figure 13: Exploratory Analysis using Decision Tree*

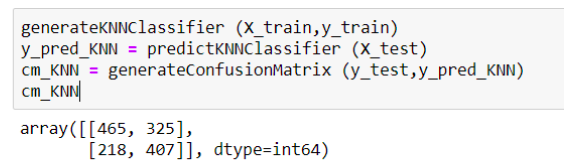
According to the Decision Tree confusion matrix as shown above, the model predicted 713 correctly as True positive whereas predicted 77 wrong as False positive. Also, the model predicted 510 as True negative but it predicted 115 wrong as False negative.

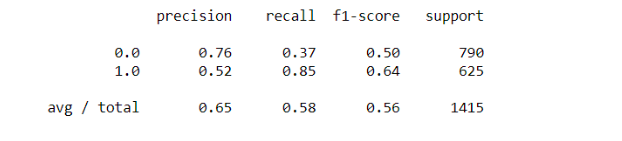
## 

## 

## 

## Analysis of the KNN results:



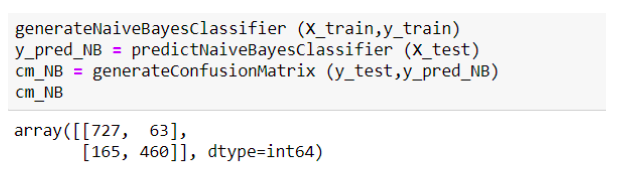


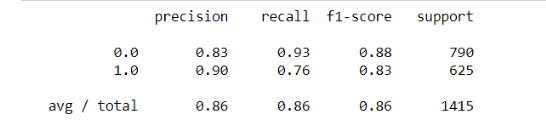
*Figure 14: Exploratory Analysis using KNN*

According to the KNN confusion matrix as shown above, the model predicted 462 correctly as True positive whereas predicted 325 wrong as False positive. Also, the model predicted 407 as True negative but it predicted 218 wrong as False negative.

## 

## Analysis of the Naive Bayes results:

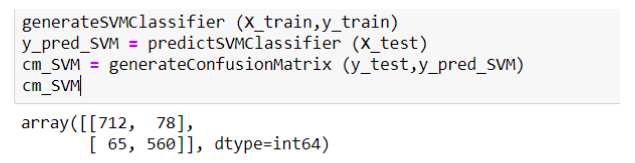


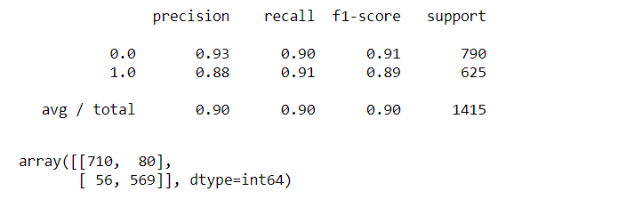


*Figure 15: Exploratory Analysis using Naive Bayes*

According to the Naive Bayes confusion matrix as shown above, the model predicted 727 correctly as True positive whereas predicted 63 wrong as False positive. Also, the model predicted 460 as True negative but it predicted 165 wrong as False negative.

## Analysis of the SVM results:

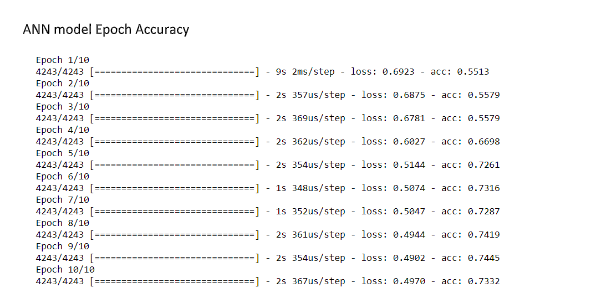


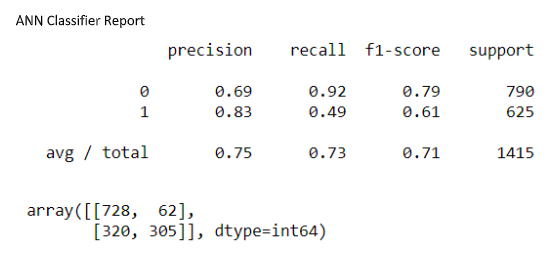


*Figure 16: Exploratory Analysis using SVM*

According to the SVM confusion matrix as shown above, the model predicted 712 correctly as True positive whereas predicted 78 wrong as False positive. Also, the model predicted 560 as True negative but it predicted 65 wrong as False negative.

## Analysis of the ANN results:





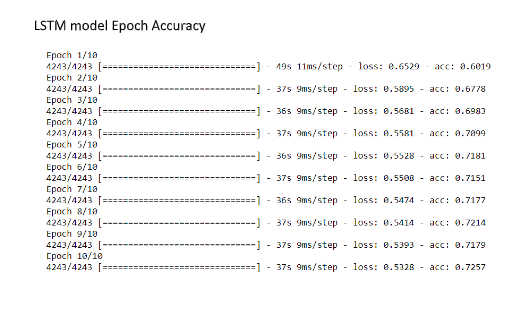
## 

## *Figure 17: Exploratory Analysis using ANN*

According to the ANN confusion matrix as shown above, the model predicted 728 correctly as True positive whereas predicted 62 wrong as False positive. Also, the model predicted 305 as True negative but it predicted 320 wrong as False negative

## 

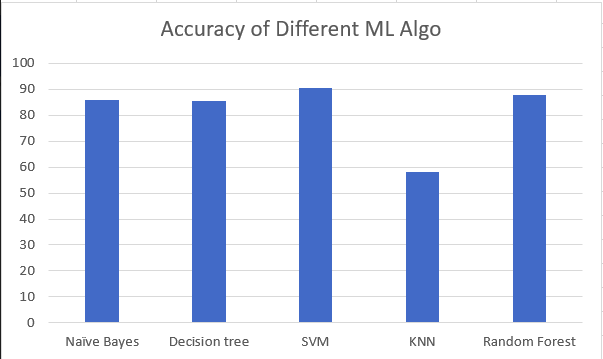
## Analysis of the LSTM results:



## 

## *Figure 18: Exploratory Analysis using LSTM*

According to the LSTM confusion matrix as shown above, the model predicted 692 correctly as True positive whereas predicted 98 wrong as False positive. Also, the model predicted 315 as True negative but it predicted 310 wrong as False negative.

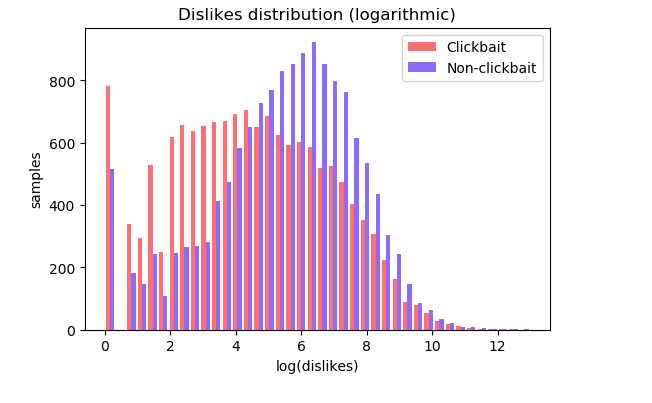


## *Figure 19: Comparison of 5 ML algorithm*

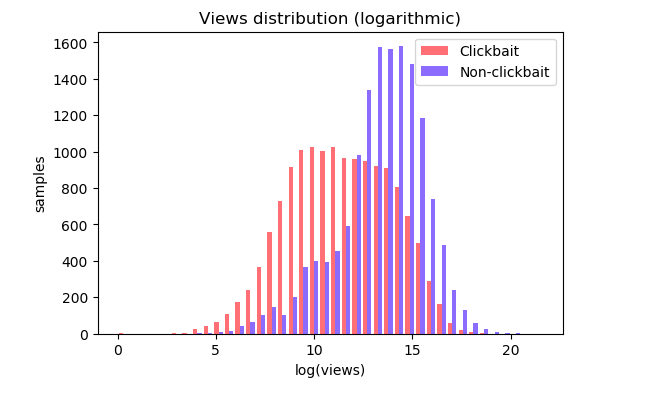
## 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-score |
| Random Forest | 0.88 | 0.88 | 0.88 | 0.88 |
| Decision Tree | 0.85 | 0.85 | 0.86 | 0.85 |
| KNN | 0.65 | 0.65 | 0.58 | 0.56 |
| Naive Bayes | 0.86 | 0.86 | 0.86 | 0.86 |
| SVM | 0.90 | 0.90 | 0.90 | 0.90 |
| LSTM | 0.72 | 0.72 | 0.71 | 0.70 |
| ANN | 0.75 | 0.73 | 0.71 | 0.71 |

## *Figure 20: Result from all baseline and Neural Network model*



*Figure 20: Dislike distribution*

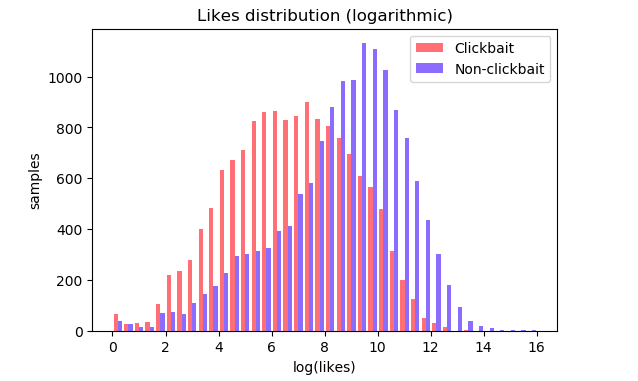
**

*Figure 21: View distribution*

## 

## 

*Figure 22: Comment distribution*

**

*Figure 23: Comment distribution*

## 

## 

*Figure 24: Top 30 tokens in clickbait*

## 

## 

## 

*Figure 25: SVM result on youtube set*

## 

## 

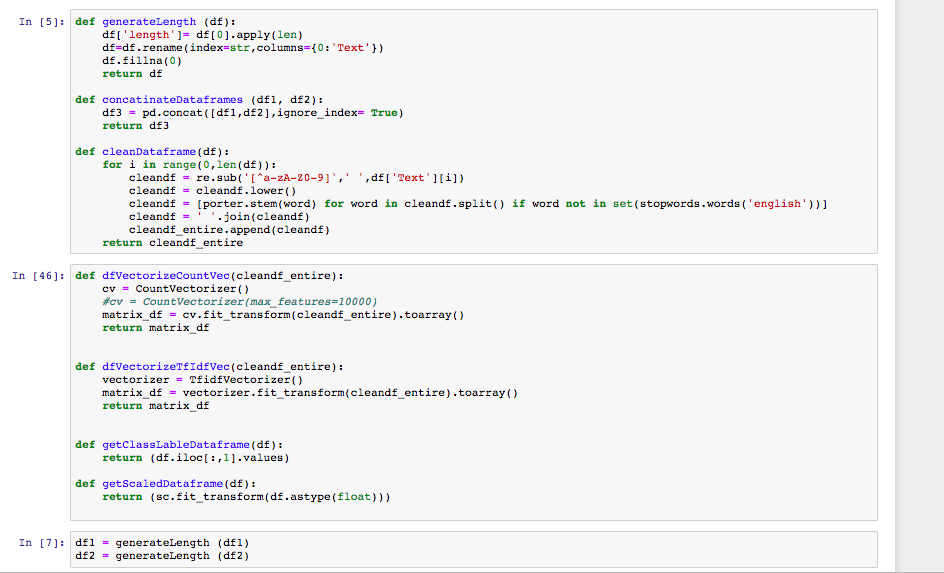
## 

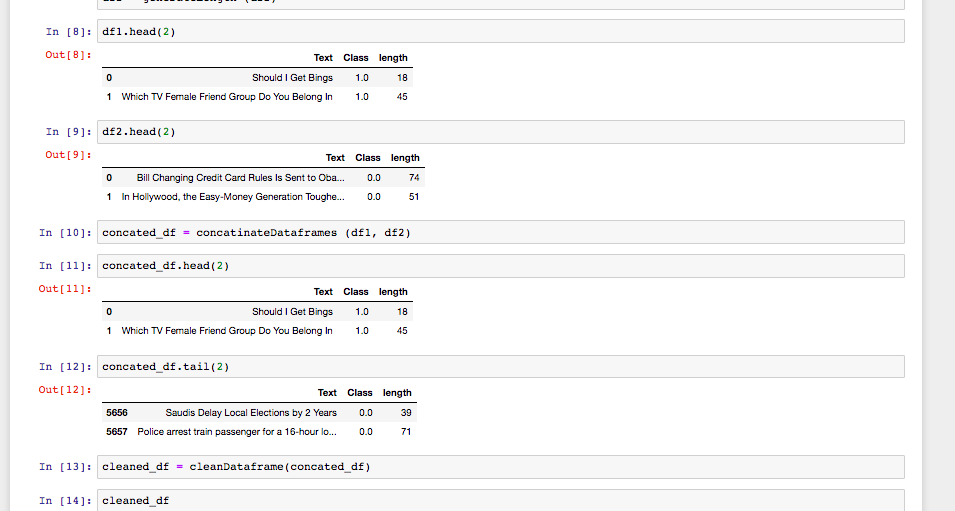
*Figure 26: Image dataset deep learning accuracy*

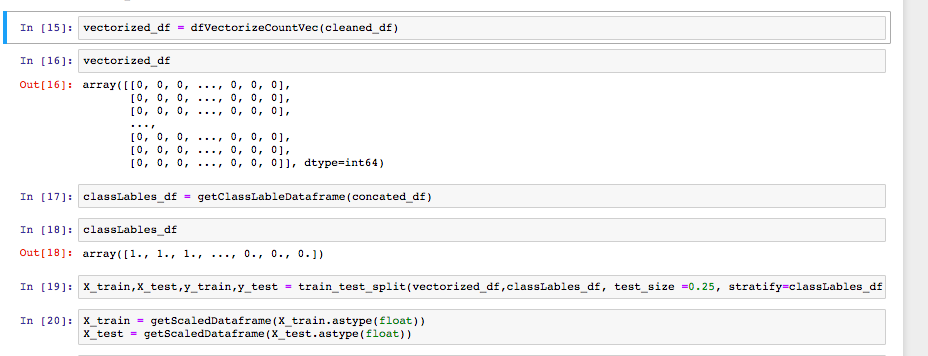
## 6.3 CODING

Module 1(Machine learning Algorithm):



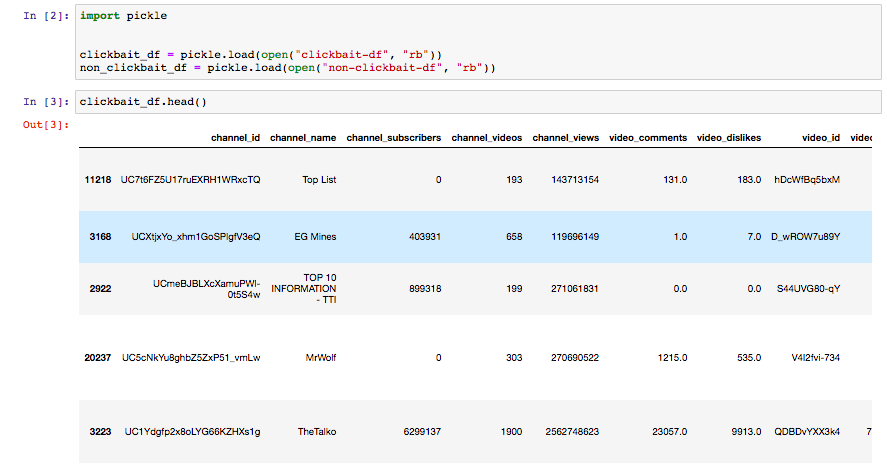


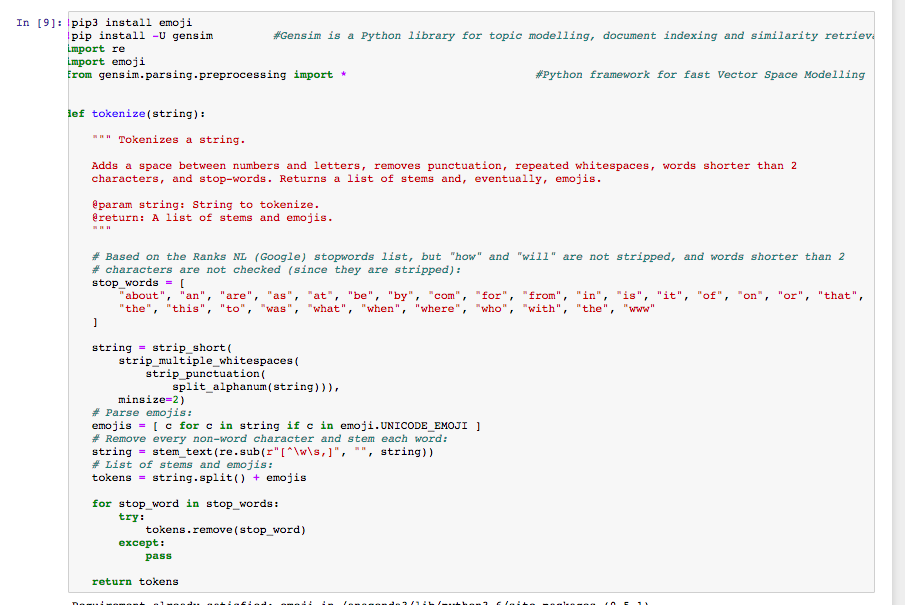




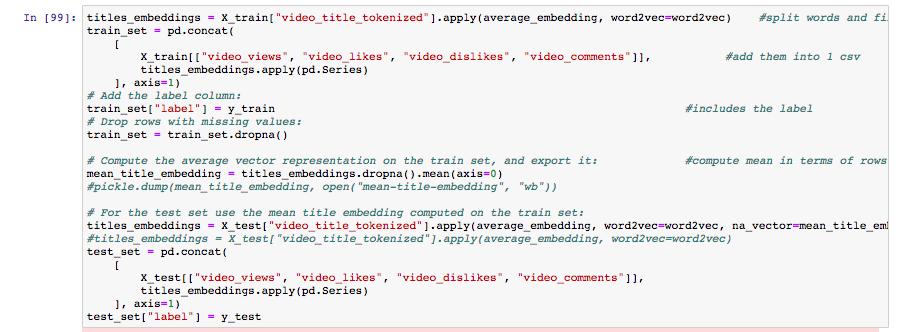


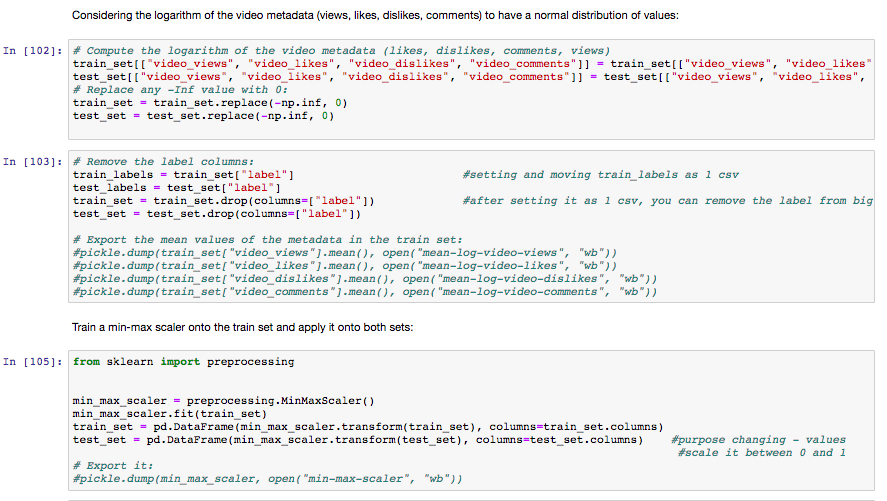
Module 2(Youtube):

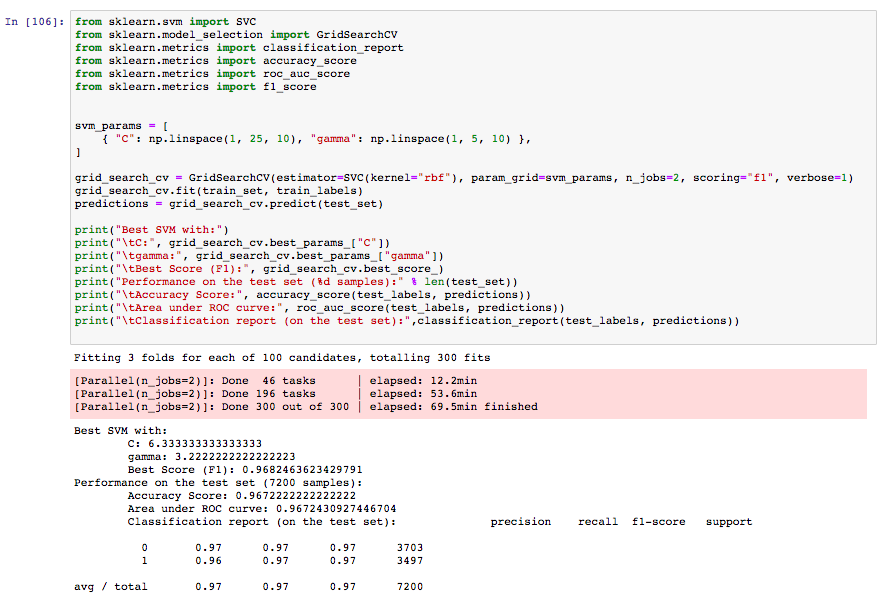


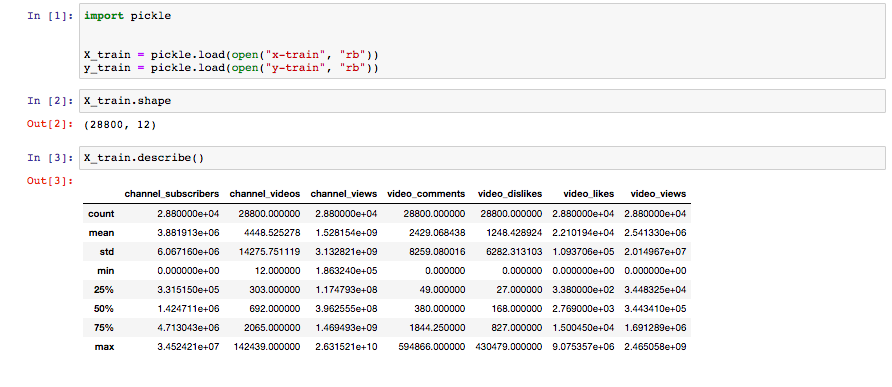
















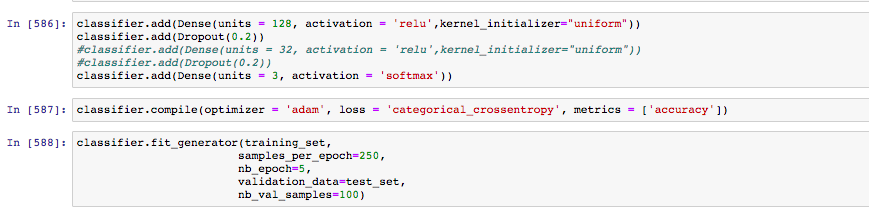






Module 3: (image)







Module 4(Deep Learning):



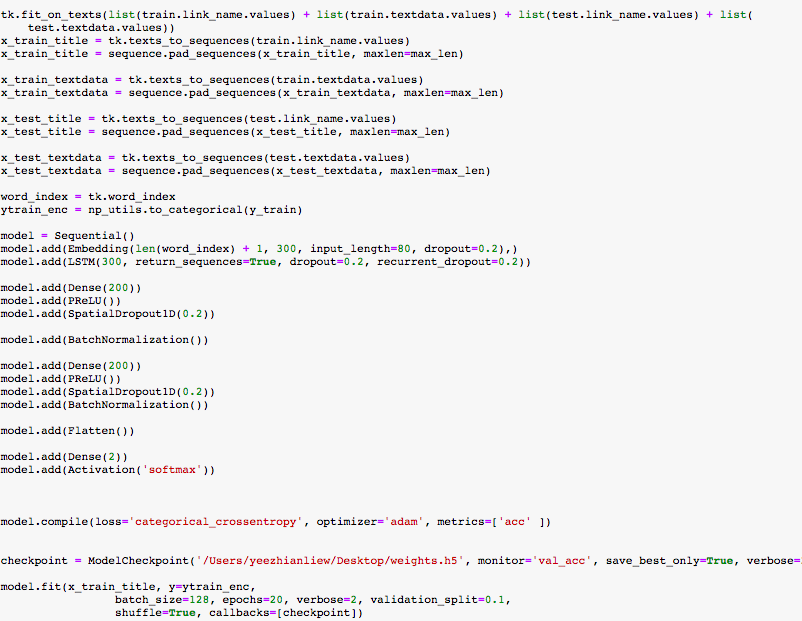












## 7. RELATED WORK

This section explains use cases or methodologies that has been done by others for a better feature selection technique.

* According to Alessiovierti’s github, he has analyzed youtube dataset which contains different features such as video likes, dislikes, comment and views. He is using SVC machine-learning techniques to determine if it will be as a effective techniques to classify video as clickbait or non clickbait. He is using parameters such as C and gamma to tune the model. By varying input features, the best classification accuracy scores around 96%.
* According to Abhishek Thakur, who is a Chief data scientist of boost.ai, he has used machine learning method such as TF-IDF, Word2Vec by using ngram parameters for data processing steps and visualization. Then, he applies algorithm which is logistic regression and gradient boosting to evaluate F1 score, precision score, recall score. The best classification is gradient boosting with accuracy around 98%.
* From the previous paper, he is revisiting back to his problem statement because he thinks content does affect the model build. He used web scraping to scrap the extra features such as number of image , number of header 2, length of the content. He is using the 5 different deep learning models to compare the accuracy, LSTM with title, LSTM with title and content, LSTM + Dense with title and content, TDD + Dense with title and content , LSTM +Dense +Numerical Dense with title and content. the best model is the LSTM +Dense +Numerical Dense model with 99.2% accuracy.

## 

## 8. CONCLUSION

We have tested using traditional Machine learning algorithm and comparing with result. From Machine learning perspective, SVM gives the most accuracy which is 90%. The second comes to Naive Bayes and Random Forest which is 86% accuracy. The worst ML algorithm for our dataset is KNN which only have 56% accuracy. Furthermore, we have added features for our model because we wanted to test whether features from websites will affects our model accuracy or not. We have implemented deep learning LSTM and ANN model and it gives us lower accuracy which is approximately 75%, comparing to traditional ML algorithm.

## 9. FUTURE WORKS

From the conclusion it can be observed that, the deep learning of LSTM model with title gradually increase the accuracy of the model. In the future work, we can use different feature reduction methodology such as Independent Component Analysis (ICA) or Distributed Stochastic Neighbor Embedding (t-SNE). PCA is a very good dimensional reduction method but it can only capture linear structures in the features. For (t-SNE), it has 2 type of approaches which is local and global approach. Local approach is mapping all the nearby point to their other nearby point in a lower dimension whereas global approach is doing the same but it also map for far point with far point in order to preserve the geometry all all scale. The main advantage for this algorithm is retaining both local and global structure of the data at the same time.

For deep learning part, we can improves our model by adding more LSTM model. This is what we call as stacking model which we are allowing the first LSTM layer to output the entire output sequence. For the second LSTM model, this entire output sequence is going to be feed in. The more layer of model, the more complexity of the algorithm calculation will become. In a single layer RNN, input is passing through a single hidden state and it might have failed to capture some interesting point and thus the accuracy will be high. With a multi-layered LSTM model, such structure is captured which results in better performance. Also, we will be applying glove embedding model into our previous model. Glove embedding is a new method of learning vector space representation of word. It sounds familiar with Word2Vec in any way and both have their own advantages. Often Word2Vec is refer as “predictive ” model whereas Glove is refer as “count based” model. Predictive model is learning from their word vector transform in order to improve predictive ability of loss. For count based model, it is reducing the dimensionality of the data and start learning the pattern from it. First, it construct a large matrix which contains words and context. As the matrix is too large, factorization is needed to squeeze it into lower dimensional space. In short, it is reducing the “construction loss” which is trying to find the representation in lower space which can show us the most variance representation in high dimensional space.

## 10. BIBLIOGRAPHY

[1] Uddin Rony, M., Hassan, N. and Yousuf, M. (2018). *Diving Deep into Clickbaits: Who Use Them to What Extents in Which Topics with What Effects?*. [online] Arxiv.org. Available at: https://arxiv.org/pdf/1703.09400.pdf [Accessed 30 Nov. 2018].

[2] Chakraborty, A., Paranjape, B., Kakarla, S. and Ganguly, N. (2018). *Stop Clickbait: Detecting and Preventing Clickbaits in Online News Media*. [online] Arxiv.org. Available at: https://arxiv.org/pdf/1610.09786.pdf [Accessed 30 Nov. 2018]

[3] Elyashar, A., Bendahan, J. and Puzis, R. (2018). *Detecting Clickbait in Online Social Media: You Won’t Believe How We Did It*. [online] Arxiv.org. Available at: https://arxiv.org/pdf/1710.06699.pdf [Accessed 30 Nov. 2018].

[4] Wiegmann, M., Völske, M., Stein, B., Hagen, M. and Potthas, M. (2018). *Heuristic Feature Selection for Clickbait Detection*. [online] Arxiv.org. Available at: https://arxiv.org/pdf/1802.01191.pdf [Accessed 30 Nov. 2018].

[5] Xing, Y. (2018). *How does clickbait work: An eye-tracking method to discover people’s reactions*. [online] Www-users.cs.york.ac.uk. Available at: https://www-users.cs.york.ac.uk/alistair/projects/yx1058.pdf [Accessed 30 Nov. 2018].

[6] Woolf, M. (2018). *Visualizing Clusters of Clickbait Headlines Using Spark, Word2vec, and Plotly*. [online] minimaxir | Max Woolf's Blog. Available at:

https://minimaxir.com/2016/08/clickbait-cluster/ [Accessed 30 Nov. 2018].

[7] Shu, K., Wang, S., Le, T., Liu, H. and Lee, D. (2018). *Deep Headline Generation for Clickbait Detection*. [online] Public.asu.edu. Available at: http://www.public.asu.edu/~skai2/papers/clickbait\_2018.pdf [Accessed 30 Nov. 2018].

[8] Cornn, K. (2018). *Clickbait article detection using deep learning: these results will shock you!*. [online] Cs230.stanford.edu. Available at: http://cs230.stanford.edu/files\_winter\_2018/projects/6931206.pdf [Accessed 30 Nov. 2018].

[9] Qu, J., Hißbach, A., Gollub, T. and Potthast, M. (2018). *Towards Crowdsourcing Clickbait Labels for YouTube Videos*. [online] Webis.de. Available at: https://webis.de/downloads/publications/papers/potthast\_2018a.pdf [Accessed 30 Nov. 2018].

[10] DeGrave, K. (2018). *Project Pages - An Integrated Scientific Blogging Template*. [online] Degravek.github.io. Available at: https://degravek.github.io/project-pages/project1/2017/04/28/New-Notebook/ [Accessed 30 Nov. 2018].

[11]Zannettou, S., Papadamou, K., Chatzis, S. and Sirivianos, M. (2018). The Good, the Bad and the Bait: Detecting and Characterizing Clickbait on YouTube. [online] Available at: [*https://www.researchgate.net/publication/323960601\_The\_Go*](https://www.researchgate.net/publication/323960601_The_Go)*od\_the\_*Bad\_and\_the\_Bait\_Detecting\_and\_Characterizing\_Clickbait\_on\_YouTube/download [Accessed 30 Nov. 2018].

[12] Thakur, A.(2018).Clickbaits Revisited: Deep Learning on Title + Content Features to Tackle Clickbaits. [online] Available at: https://www.linkedin.com/pulse/clickbaits-revisited-deep-learning-title-content-features-thakur/[Accessed 30 Nov. 2018].