CLICKBAIT DETECTION AND CLASSIFICATION

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

Submitted By:

Aaryaman Bajaj
2K17/IT/02
Himanshi Nimesh
2K17/IT/52
Raghav Sareen
2K17/IT/90

Under the Supervision of:

Prof. Dinesh Vishwakarma
Associate Professor, Department of IT, DTU

INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CANDIDATE DECLARATION

I hereby certify that the Project Dissertation titled "**Identification and Classification of clickbait**" which is submitted by Aaryaman Bajaj(2K17/IT/02), Himanshi Nimesh(2K17/IT/52), Raghav Sareen(2K17/IT/90) of Information Technology, Delhi Technological University, Delhi, in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi **Aaryaman Bajaj**

Date: 06 December, 2020 **Himanshi Nimesh**

Raghav Sareen

INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

CERTIFICATE

On the basis of the Major Project-I, submitted by Aaryaman Bajaj (2K17/IT/002), Himanshi Nimesh (2K17/IT/052) and Raghav Sareen (2K17/IT/090) students of B.Tech. (IT). I hereby certify that the project titled "Clickbait Detection and Classification" which has been submitted to Department of Information Technology, Delhi Technological University in partial fulfilment of the requirement for the award of degree of BACHELORS OF TECHNOLOGY in Information Technology, is an original contribution with existing knowledge and faithful record of work carried out by them under my guidance and supervision.

To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi
Date: 06-12-2020
Delhi Technological University

Prof. D.K. Vishwakarma
Associate professor, Dept of IT
SUPERVISOR

INFORMATION TECHNOLOGY
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I want to express my sincere thanks to my HOD, Dr. Kapil Sharma, Department of Information Technology, Delhi Technological University, Delhi for providing well equipped infrastructure support. I would like to express my deep and sincere gratitude to my project mentor Dr. D.K. Vishwakarma , Associate Professor, Information Technology Department, Delhi Technological University, Delhi for giving us the opportunity to do research and providing invaluable guidance and constant encouragement throughout the project.

We are extremely grateful to our family for their love, prayers, caring and sacrifices for educating and preparing us for our future and for not letting us down at the time of crisis and showing me the silver linings in the dark cloud.

This thesis would have not been completed without the constant support of my best friends. Finally, I thank all the faculty and staff members of Delhi Technological University, Rohini, Delhi for extending a helping hand at every juncture of need.

AARYAMAN BAJAJ (2K17/IT/02)

HIMANSHI NIMESH (2K17/IT/52)

RAGHAV SAREEN (2K17/IT/90)

B.Tech Department of Information Technology
Delhi Technology University, Delhi

ABSTRACT

Clickbait is a term used to describe a deceiving web content that uses ambiguity to prompt the user into clicking a link. It aims to increase the number of online readers in order to generate more advertising revenue. Typically such links will forward the visitor to a page that requires payment, registration, or lead a user to a site, which tries to sell the user something or possibly extort the user, by withholding the promised "bait". Some examples are "10 places you must visit before you're 30", "5 reasons why celebrities love THIS work", etc. These kinds of clickbait actually work because words like 'you' make the headlines relatable and entice the viewers to open the link.

An element of mystery is always there along with the relatability, which makes the reader click on the link to know more about the “10 amazing inventions you won’t believe exist”.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1- INTRODUCTION

1.1 General

Clickbait is a lure for seeing the detail. It makes the person reading the content feel curious to know more about what is hidden. However, the contents behind it are some types of malware or ads for some products. Also, Clickbait has been used to spread fake news. For instance, the news “Russia uses fake photos from video games to accuse the US of supporting ISIS”. This is an unethical practice to lure readers, because these Clickbait web pages are often headlines but the content inside is nothing. Clickbait defines topics that link a webpage to a webpage which is less related or not related at all. The topics or subjects are generally presented with sensationalism and exaggerations. News having clickbait headlines was first introduced in 1883 in New York World newspaper by showcasing little or no legitimate news. The aim of clickbait is to attract the people using the Internet to visit the website in order to increase the number of web-counters. The number of website visitors reflects the popularity of that particular website. Although clickbait headline news successfully gains attention from the readers and helps generate revenue to the one who owns the website, they are annoying to the Internet users. Due to this reason, Facebook announced the technical measures to reduce the impact of clickbait on its social network. Many methods have been proposed for classifying the headline news that are actually clickbait. The links are categorised based on users’ behavior such as the time being spent on a page after clicking the link. However, such a method can detect only the news headlines that are previously known. To efficiently identify the clickbait headlines, the text mining task has been used. Given the clickbait headlines, their patterns are drawn out and a classification model is constructed for classifying the news headline. Using the data mining techniques to classify the headline news has been introduced in 2016. The methods such as Random Forest, Support Vector Machine (SVM), and Convolutional Neural Network (CNN) have been implemented to classify English news headline

1.2 Why is clickbait detection important ?

Social networks generate huge amounts of complex textual data and it is becoming increasingly difficult to process the data intelligently. Misinformation on social media, in the form of fake news, has the power to influence people, affect opinions and can also have a decisive impact on elections. To protect ourselves from manipulative misinformation, we need to develop a mechanism to be relied upon to detect fake news. Yellow journalism along with sensationalism has inflicted a lot of damage by manipulating readers into believing false narratives through hyperbole and also by misrepresenting facts. Clickbait does exactly this by employing characteristics of natural language to lure users into clicking a particular link and can hence be classified as fake news. People are increasingly turning to social media for news. Unfortunately, they are getting abused by propaganda websites. These websites use Clickbaits with the intent not only to increase revenues but also to spread their propaganda too. Clickbait articles are intentionally over-promises to increase revenues but also to spread their propaganda too. Clickbait articles are intentionally over-promises.

Identifying Clickbaits is an important part of blocking them from the user's social media feed. Facebook, Twitter and other social media websites faced much criticism for not identifying Clickbaits and down ranking them from the user's feed.

1.3 Methods Used:

1.3.1 Decision Trees

Decision trees are the most favoured and powerful tool for prediction and classification. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch shows an outcome of the test, and each leaf node (terminal node) holds a class label. A decision tree may be used to visually represent decisions and decision making in decision analysis. It uses a tree-like model of decisions, as the name suggests.

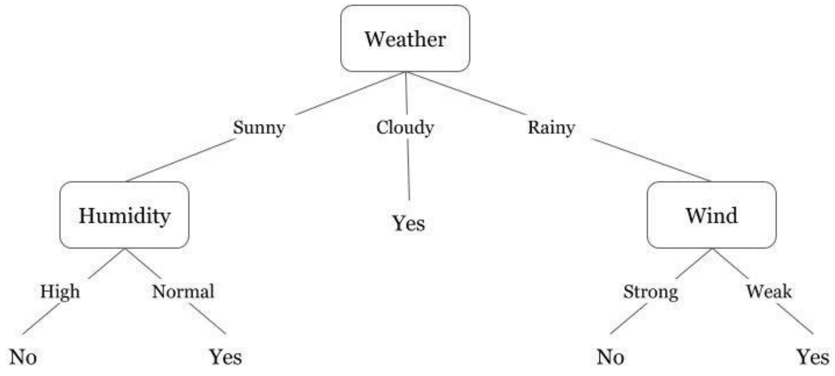


Figure 1.1- Example of a decision tree

1.3.2 Random Forests

Random Decision Forests or Random Forests are an ensemble learning method for regression, classification and other tasks that require the construction of a multitude of decision trees at the time of training and outputting the class that is the average /mean prediction (regression) of the individual trees or the mode of the classes (classification). The process of finding the root nodes and splitting the feature nodes runs randomly in the case of Random Decision Forests. In the case of classification problems, Random Forests avoid overfitting, which is one of their many advantages.

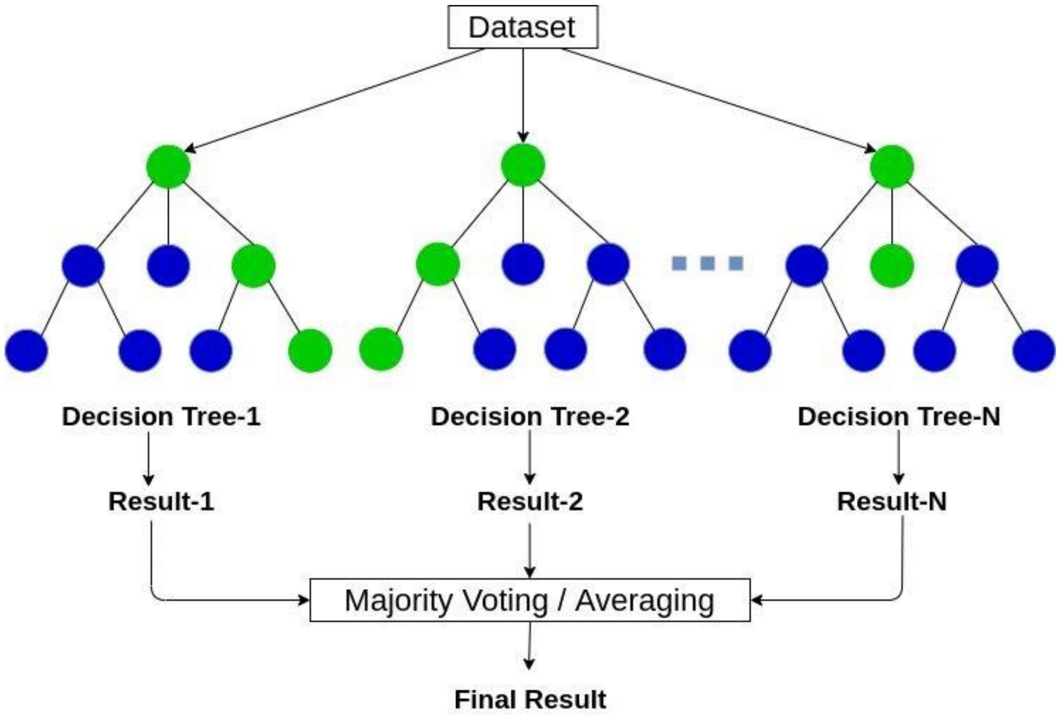


Figure 1.2- Random Forest

1.3.3 Logistic Regression

Logistic Regression is a supervised machine learning model which is used for classification problems. It is used to determine binary output based on a set of given independent variables. Logistic regression is somewhat similar to linear regression except it predicts whether something is True or False, instead of predicting something continuous like size.

Based on those number of categories, Logistic regression can be divided into following types −

- Binary or Binomial- In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.
- Multinomial- In such a kind of classification, dependent variable can have 3 or more possible unordered types or the types having no quantitative significance.
- Ordinal- In such a kind of classification, dependent variables can have 3 or more possible ordered types or the types having a quantitative significance.

We can call a Logistic Regression a Linear Regression model but the Logistic Regression uses a more complex cost function, this cost function can be defined as

the “Sigmoid function” instead of a linear function. The hypothesis of logistic regression tends it to limit the cost function between 0 and 1.

Formula of a sigmoid function: For logistic regression we are going to modify it a little bit i.e.

$\sigma (Z) = \sigma (\beta_0 + \beta_1 X)$

We have expected that our hypothesis will give values between 0 and 1.

$Z = \beta_0 + \beta_1 X$

$h\Theta (x) = \text{sigmoid} (Z)$

i.e. $h\Theta(x) = 1 /(1 + e\text{^}-(\beta_0 + \beta_1 X)$

1.3.4 Naive bayes

What is Naive Bayes algorithm?

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Above,

- P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
- P(c) is the prior probability of class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor

Chapter 2- LITERATURE REVIEW

The first method of machine learning for clickbait acquisition was proposed by Potthast et.al, in 2016. 2,992 Twitter tweets, of which 767 are clickbit, and were collected in the top 20 Twitter publishers. Tweets were divided into three categories: the teaser message, the linked web page, and the meta information. The clickbait acquisition model is built based on 215 features such as the linked web page, N-gram name, and sender name. of collected tweets. Test results showed that Random Forest Classifier scored 0.79 ROC-AUC with 0.76 accuracy and 0.76 accuracy over Logistic Regression and Naïve Bayes.

Chakraborty et.al, had collected 7,500 non-click news articles from Wikinews 1 and 7,500 clickbait articles from Clickbait websites. Stanford CoreNLP tools have been used to extract a set of 14 features such as sentence structure, word patterns, clickbait language and N-gram. Test results show that SVM successfully outperformed Decision Tree and Random Forests with an accuracy of 0.93, and 0.95 and 0.9 precision and recall respectively.

Instead of using a collection of hand-selected features such as [2,15], Agrawal [1] used an in-depth learning model called Convolutional Neural Networks (CNNs) to automatically detect the characteristics of the collected data. Top 814 clickbait and 1574 non-clickbait headlines were found on social media, namely Reddit, Facebook and Twitter. With the accuracy of clickbait data, many votes from three independent inspectors were used to assign the label. Two tests were performed. The first is a different comparison of word embedding, i.e., learning from scratch and learning from a training copy with Word2vec method. In the second test, Word2vec was used to configure hyper-parameters such as windows. size, drop rate, and embedding size. The test result showed that the CNN model with embedded words from Word2vec achieves 0.90 accuracy with 0.85 precision and 0.88 recall.

Table 2.1- The list of clickbait literature review

	Method	Dataset	Result
Clickbait Detection	Random Forest with 215 features	Clickbait: 767 Non-Clickbait: 2225	ROC-AUC: 0.79 Precision: 0.76 Recall: 0.76
Stop Clickbait: Finding and Avoiding Clickbaits in Online News Media	Support Vector Machine with 14 features	Clickbait: 7500 Non-Clickbait: 7500	ROC-AUC: 0.93 Precision: 0.90 Recall: 0.95
Clickbait Detection using Deep Learning	Convolutional Neural Network (CNN)	Clickbait: 814 Non-Clickbait: 1574	ROC-AUC: 0.90 Precision: 0.895 Recall: 0.90

Chapter 3- METHODOLOGY

3.1 Clickbait Dataset

This dataset contains headlines from various news sites such as ‘WikiNews’, ’New York Times’, ‘The Guardian’, ‘The Hindu’, ‘BuzzFeed’, ‘Upworthy’, ‘ViralNova’, ‘Thatscoop’, ‘Scoopwhoop’ and ‘ViralStories’. It has two columns, the first one contains headlines and the second one has numerical labels of clickbait in which 1 represents that it is clickbait and 0 represents that it is non-clickbait headline. The dataset contains a total 32000 rows of which 50% are clickbait and other 50% are non-clickbait.

Table 3.1- View of the dataset

	headline	clickbait
0	Earthquake reported near Rome's coast, no dama...	0
1	Taylor Swift's Cat Olivia Benson Chewed Up An...	1
2	New Jersey students protest proposed budget cuts	0
3	Can You Solve This New Years Crossword	1
4	American author Michael Crichton dies at age 66	0

‘Headline’ contains headlines from news sites in text format and ‘Clickbait’ contains numeric labels of which 1 represents it is clickbait and 0 represents non-clickbait.

```
ax=titles['clickbait'].value_counts().plot(kind="bar")
ax.set_xlabel('Value')
ax.set_ylabel('Frequency')
```

```
Text(0, 0.5, 'Frequency')
```

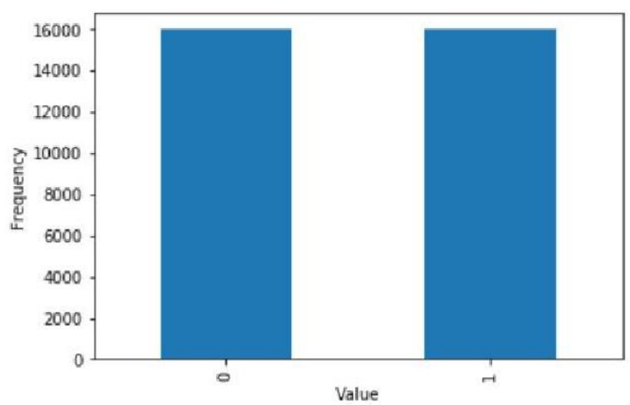


Figure 3.1- Frequency of clickbait and non-clickbait titles

3.2 Steps involved in Data Cleaning

3.2.1 Data Cleaning: Raw data may contain missing or inessential values. This may lead to inconsistencies in the dataset. All these issues are handled in the data cleaning step. All the missing values are filled by following proper criteria and all the inessential values are removed.

3.2.2 Creating Test and Training Sets: We bisect a data set into a training set, validation set, and testing set where a major section of the data is used for training (around 80%) purpose and a smaller section of the data is used for testing (10%), and validation (10%) purpose.

3.2.3 Feature Scaling: When the independent variables are measured at different scales then this may result in one variable dominating other variables. Thus feature scaling is an important step where all the data values are transformed to a certain range. This ensures all the variables having equal contribution in training the dataset. There are two widely used methods of feature scaling:

- Standardization: $X_{stand} = (X - Mean(X))/(Standard\ Deviation(X))$
- Normalization: $X_{Norm} = (X - Min(X))/(Max(X) - Min(X))$

3.3 Steps involved in making the model:

3.3.1 Preprocessing and Analysis

Implement the following operations on each string:

- converts to lower-case
- expand contractions
- remove punctuation
- lemmatize words

We also count the number of contractions for each headline. This is because, usually clickbait headlines have a higher ratio of contracted words.We have defined all the parsing functions we need.

3.3.2 Bag of words

Some words are frequently use in Clickbait headlines to make it look more alluring to the readers without giving any contextual clarity about what is discussed in the article. After the removal of stop words, we extract the 100 most frequently used words in Clickbait titles and the 200 most commonly used words in non-Clickbait titles. We have extracted words that are exclusively used in Clickbait titles. ‘Trump’, Obama’, etc are then deleted. The common words used in Clickbaits are ‘you’, ‘this’, ‘people’.

Table 3.2- 13 most common clickbait words (in alphabetical order):

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
No Clickbait	after	and	as	at	be	by	for	from	in	kill	new	num	of	on
Clickbait	about	and	be	do	for	have	in	make	num	of	on	that	the	thing
Full	and	at	be	do	for	from	have	in	num	of	on	that	the	this

3.3.3 Create the Pipeline

Time to create a Pipeline that generates all the features. Together with the words count we add 4 additional features

- headline length (number of words)
- stopwords ratio
- contractions ratio
- a flag if the headline starts with a number

3.3.4 Train some classifier

- Train Naive Bayes only on the Bag of Words

  First we tried it on Train_mini, then on the full rain set. Now we fit the model on the full train set. We used it's output as a feature for a SVC and Random Forest.

- Train Random Forest on Naive Bayes probabilities and non-word features

  First we train it on train_mini. Then we fine tune some hyperparameters.
  And now use Cross Validation on the full train set to get an estimate of the performances.

- ➢ Train SVM on Naive Bayes probabilities and non-word features
- ➢ Train naive Random Forest on all the features- We try a blind Random forest on all the 204 features.
- ➢ Train naive SVM on all the features

  We trained a blind SVC on all the 204 features. Before doing that we rescaled the features.

3.3.5 Test on Validation set

Evaluate the 5 classifiers we've built on the Validation set to see which performs better.

3.3.6. Finally we evaluated the Test Set

Chapter 4- DISCUSSION

4.1 Evaluation

Given a testing headline the evaluations is measured using Precision (P), Recall (R), F-measure (F1), and Accuracy. Precision was defined as the percentage of the correctly predicted headlines for a clickbait by the total number of headlines predicted to be clickbait. Recall was defined as the percentage of correctly predicted headlines for a clickbait divided by the total number of clickbait headlines over the test set. F-measure is the weight of precision and recall. Accuracy is defined as the percentage of the retrieved headlines that are correctly in the same class as the query by the number of headlines in the test set. The four measurements can be computed using the following Equation:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F1 = \frac{2*P*R}{P+R} \quad (3)$$

$$Accuracy = \frac{TP+FP}{N} \quad (4)$$

where TP is a true positive which represents the number of labels in clickbait set those are predicted as clickbait, FP is false positive which shows the number of labels in non clickbait set those are predicted as clickbait, FN is false negative which is the number of labels in clickbait set those are predicted as non-clickbait, and N is the number of data in the test set.

4.2 Results

The codes and corresponding results for each of the classifier is given in appendices (A.1, A.2, A.3, A.4)

Plotting the graphs gives us the following:

Figure 4.1 Precision

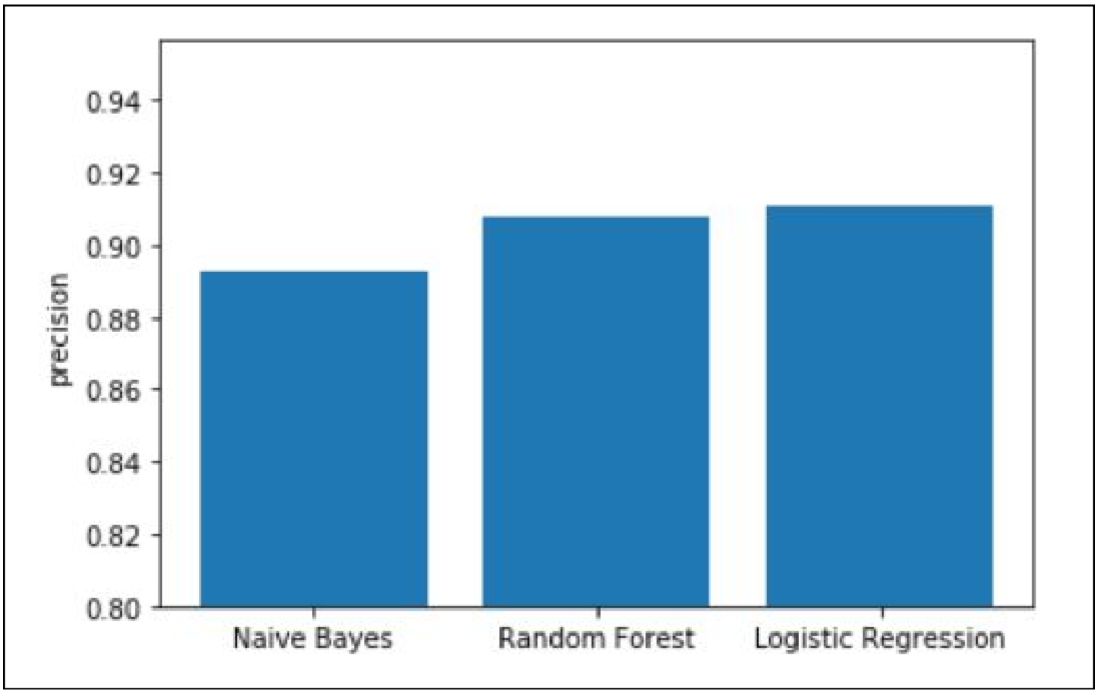


Figure 4.2 Recall

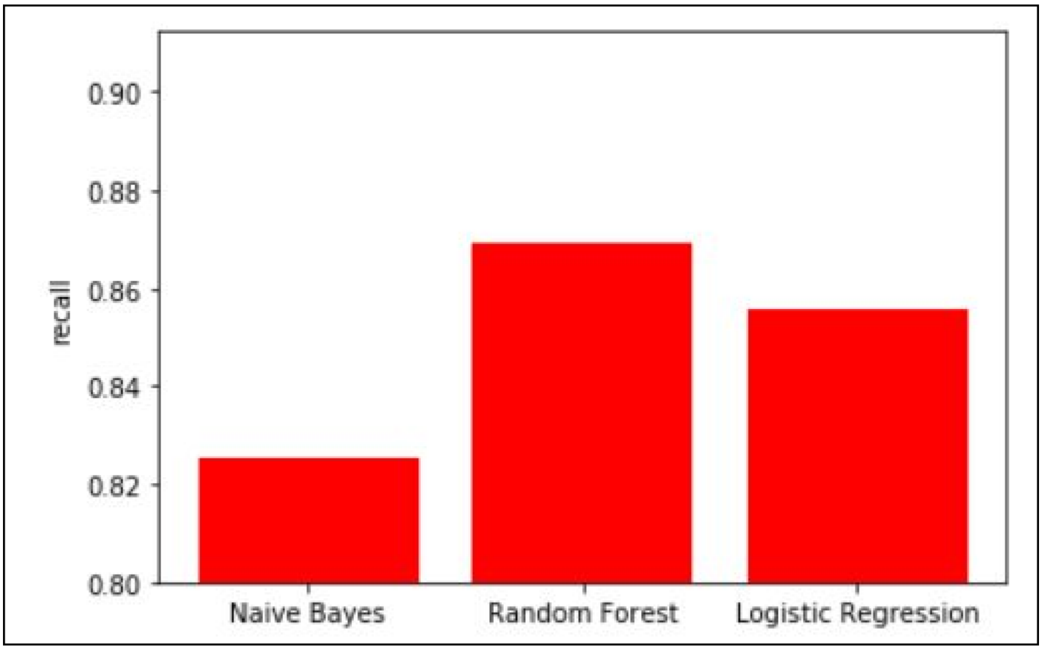
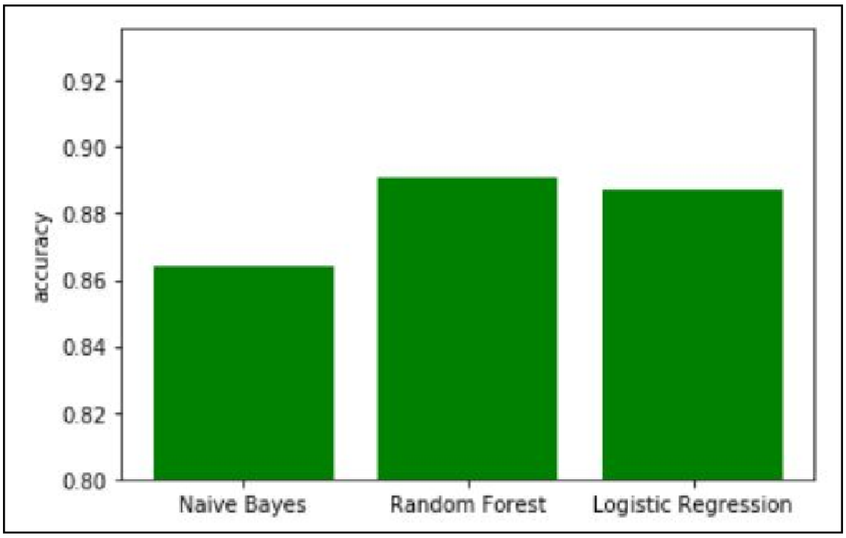


Figure 4.3 Accuracy



We obtain the following accuracy scores on applying the methods:

1. Naive Bayes: 0.864
2. Random Forest: 0.891
3. Logistic Regression: 0.887

We get the best results from Random Forests. This is because we first train it on a miniature dataset of 1000 instances only, and then we fine tune some hyperparameters. After tuning the parameters, our measures show better results.

4.3 Future Study

Future work may include different learning approaches and different methods for extracting features and some modifications that can be even more useful in a product than in research.

However, we are planning our future work by:

- Restricting the training of our model to the features extracted from all the parts of a post excluding the article should reduce the storage of data and processing required to fetch and save the article and should also make the classifier faster.

- Discovering the features using unsupervised ML techniques could result in higher accuracy but cannot be done as of now due to the need for a larger and even more diverse dataset and the longer time needed for the completion of each in the learning process.

CONCLUSION

In this research, we presented a different approach to the Clickbait classification problem. In the method, we showed the relevance of social media post elements. Moreover, we showed the importance of new features that were extracted from different elements of social media as well as features extracted from the title and the article. We demonstrated that the detection of Clickbaits could be done using a minimal number of features. Based on the above we can conclude the following:

- Clickbait detection is possible on social media platforms with better performance if elements of posts on such platforms are correctly used.
- A low number of features can still be sufficient to classify Clickbaits, which helps in building a real-time classifier moving this idea from theory to application.

APPENDIX

A.1 Naive bayes-

```
y_val_pred = mnb_clf.predict(X_val_prep.toarray()[:,:200])

print('Score on the test set: %.3f' % accuracy_score(y_val, y_val_pred))
print('Precision: %.3f' % precision_score(y_val, y_val_pred))
print('Recall: %.3f' % recall_score(y_val, y_val_pred))
```

```
Score on the test set: 0.864
Precision: 0.893
Recall: 0.825
```

A.2 Random forest on 5 features-

```
forest_clf.fit(X_train_forest, y_train)
y_val_pred = forest_clf.predict(X_val_forest)

print('Score on the test set: %.3f' % accuracy_score(y_val, y_val_pred))
print('Precision: %.3f' % precision_score(y_val, y_val_pred))
print('Recall: %.3f' % recall_score(y_val, y_val_pred))
```

```
Score on the test set: 0.864
Precision: 0.891
Recall: 0.828
```

A.3 Random forest on all features:

```
forest_clf_v1.fit(X_train_prep, y_train)
y_val_pred = forest_clf_v1.predict(X_val_prep)

print('Score on the test set: %.3f' % accuracy_score(y_val, y_val_pred))
print('Precision: %.3f' % precision_score(y_val, y_val_pred))
print('Recall: %.3f' % recall_score(y_val, y_val_pred))
```

```
Score on the test set: 0.891
Precision: 0.908
Recall: 0.869
```

1. Logistic Regression

```
y_val_pred = logi_clf.predict(X_val_prep)

print('Score on the test set: %.3f' % accuracy_score(y_val, y_val_pred))
print('Precision: %.3f' % precision_score(y_val, y_val_pred))
print('Recall: %.3f' % recall_score(y_val, y_val_pred))
```

```
Score on the test set: 0.887
Precision: 0.911
Recall: 0.856
```

2. Functions for calculating score

```
def print_scores(clf_cv, acc=True, prec=True, rec=True):
    if acc:
        print('accuracy: %.3f' % clf_cv['test_accuracy_score'].mean())
    if prec:
        print('precision: %.3f' % clf_cv['test_precision_score'].mean())
    if rec:
        print('recall: %.3f' % clf_cv['test_recall_score'].mean())
```

REFERENCES

1. A.Chakraborty, B.Paranjape, S.Kakarla, N.Ganguly, "Stop Clickbait: Detecting and Preventing Clickbaits in Online News Media", https://www.researchgate.net/publication/309572896_Stop_Clickbait_Detecting_and_Preventing_Clickbaits_in_Online_News_Media
2. L.Bordin, "Clickbait Classifier", https://www.kaggle.com/lorenzobordin/clickbait-classifier
3. Y.Zhou, "Clickbait Detection in Tweets Using Self-attentive Network", https://paperswithcode.com/paper/clickbait-detection-in-tweets-using-self
4. D.Daoud, S.A.El-Seoud, "An Effective Approach for Clickbait Detection Based on Supervised Machine Learning Technique", https://www.researchgate.net/publication/331109738_An_Effective_Approach_for_Clickbait_Detection_Based_on_Supervised_Machine_Learning_Technique
5. G.Karadzhov, P.Gencheva, P.Nakov, I.Koychev, "We Built a Fake News / Clickbait Filter: What Happened Next Will Blow Your Mind!", https://paperswithcode.com/paper/we-built-a-fake-news-click-bait-filter-what-1
6. S.Kongyoung, "Clickbait Detection for Thai News Headline", http://ethesisarchive.library.tu.ac.th/thesis/2017/TU_2017_5922040604_8706_6829.pdf
7. A.Anand, T.Chakraborty, N.Park, "We used Neural Networks to Detect Clickbaits: You won't believe what happened Next!", https://paperswithcode.com/paper/we-used-neural-networks-to-detect-clickbaits
8. O.Papadopoulou, M.Zampoglou, S.Papadopoulos, I.Kompatsiaris, "A Two-Level Classification Approach for Detecting Clickbait Posts using Text-Based Features", https://paperswithcode.com/paper/a-two-level-classification-approach-for
9. T.Jain, "InstaAsli", https://github.com/tanamania5555/InstaAsli
10. A.Anand, "Deep Clickbait Detection", https://github.com/ankeshanand/deep-clickbait-detection
11. N.Hurst, "To clickbait or not to clickbait?", https://mospace.umsystem.edu/xmlui/bitstream/handle/10355/56080/research.pdf?sequence=2&isAllowed=y