**CHAPTER 1**

**INTRODUCTION**

Natural language processing (NLP) is the intersection of computer science, linguistics, and [machine learning](https://builtin.com/data-science/introduction-to-machine-learning). The field focuses on communication between computers and humans in natural language and NLP is all about making computers understand and generate human language. [Applications of NLP](https://builtin.com/artificial-intelligence/natural-language-processing-examples-applications) techniques include voice assistants like Amazon's Alexa and Apple's Siri, but also things like machine translation and text-filtering.

NLP has heavily benefited from recent advances in machine learning, especially from deep learning techniques. The field is divided into three parts:

* **Speech Recognition:** The translation of spoken language into text.
* **Natural Language Understanding:** The computer’s ability to understand what we say.
* **Natural Language Generation:** The generation of natural language by a computer.

**Clickbait** is a text or a [thumbnail](https://en.wikipedia.org/wiki/Thumbnail) [link](https://en.wikipedia.org/wiki/Hyperlink) that is designed to attract attention and to entice users to follow that link and read, view, or listen to the linked piece of online [content](https://en.wikipedia.org/wiki/Content_(media)), being typically [deceptive](https://en.wikipedia.org/wiki/Deceptive), [sensationalized](https://en.wikipedia.org/wiki/Sensationalized), or otherwise [misleading](https://en.wikipedia.org/wiki/Misleading). Clickbait is primarily used to drive page views on websites,[[15]](https://en.wikipedia.org/wiki/Clickbait#cite_note-15) whether for their own purposes or to increase [online advertising](https://en.wikipedia.org/wiki/Online_advertising) revenue. There are various clickbait strategies, including the composition of headlines of news and online articles that build suspense and sensation, luring and teasing users to click.[[21]](https://en.wikipedia.org/wiki/Clickbait#cite_note-21) Some of the popular approaches in achieving these include the presentation of links and images that are interesting to the user, exploiting curiosity related to greed or prurient interest.

This project deals with the classification of the effectiveness of the headlines to be used as clickbait in the modern generation where search engines are heavily used for a website to be called popular. The data is collected from various news sites. The clickbait headlines are collected from sites such as ‘BuzzFeed’, ‘Upworthy’, ‘ViralNova’, ‘Thatscoop’, ‘Scoopwhoop’, and ‘ViralStories’. The relevant or non-clickbait headlines are collected from many trustworthy news sites such as ‘WikiNews’, ’New York Times ’, ‘The Guardian’, and ‘The Hindu’. We can apply different classification algorithms to classify the data into clickbait and non-clickbait.

**CHAPTER 2:**

**LITERATURE SURVEY**

In this project phase, we will study some of the machine learning algorithms based on a mathematical model which classifies a given dataset into one of the categories namely clickbait and non-clickbait.

The study of clickbait detection in Indonesian-language articles used NB as a classification method [10]. It measured its effect based on the number of "shares" and "likes" in a news article on Facebook, while other similar research uses Neural Network (NN) with Backpropagation algorithm [9]. Neural Network is a technology that is able to study the input of previous and new data obtained. The process of learning data and NN prediction is carried out through a network of neurons that are connected and arranged into a layered structure [40]. Conversely, Backpropagation is a hierarchical design consisting of layers or rows of processing units that are fully interconnected [41].

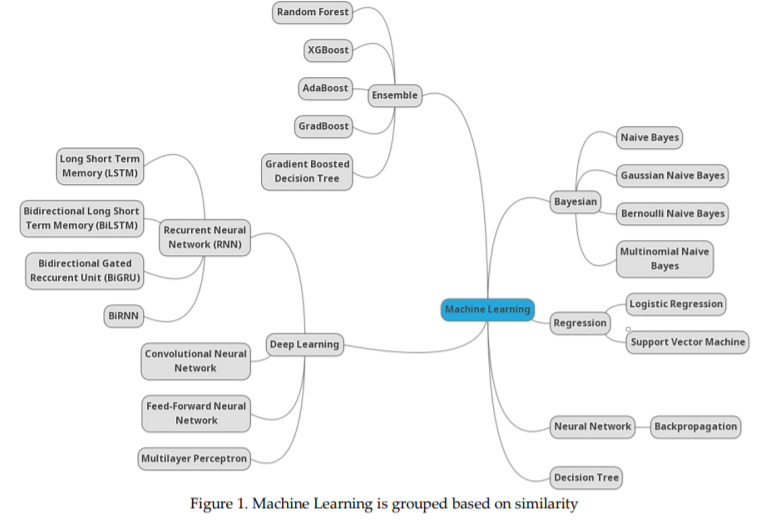
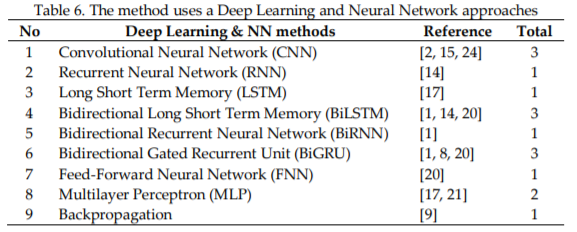


Table 6 shows some methods of deep learning (DL) approach, which is made and functions based on Artificial Neural Networks (ANN). The science of human cognition is used to understand DL [42], while [15] used the Convolutional Neural Network (CNN) as a classification method. CNN functions automatically to provide a vector that is used as a representation of news headlines. It calculates the phrase vector for every single phrase in a sentence [43]. In addition, its purpose in clickbait detection is to represent news headlines as vectors which have fixed size, continuous, and real values as classifiers [15]. Another method used to detect clickbait is Feed-Forward Neural Network (FNN), an algorithm consisting of several layers of neurons, each of which is determined by a set of synaptic weights that are appropriate [44].



Recurrent Neural Network (RNN) is a class of ANN and a system that functions in decision making by considering the things and information that already exists. It processes data sequentially, and not suitable for long processing as it is hampered by the learning process. Therefore, the LSTM model is developed to overcome this problem [45]. Manjesh et al. conducted a study using LSTM by analyzing sentiment to determine the classification of a sentence by indicating whether it is positive or negative. The analysis found that nonclickbait titles naturally tend to use more neutral language [17]. This study also analyzed the average number of words used in the clickbait title, which showed that non-clickbait articles tend to contain shorter phrases due to its ability to represent the content of news directly [17]. Manjesh et al. [17] compared several machine learning algorithms using deep learning methods (LSTM and MLP). This study showed that the deep learning method works better than the machine algorithm [17]. Figure 1 show the mind mapping of machine learning of several methods found based on similarity.

Rony et al. [10] extended the prevention of ‘Clickbait’ to a social media platform. They have used the Skip-Gram model [11] to use word embeddings, which is further used to find out the similarity between the texts. They have made the comparison of their pre-trained vectors with the Google news dataset. They compared the results with earlier findings in the field of clickbait detection with pretrained vectors and without pre-trained vectors. They have also categorized the percentage of different media in terms of clickbait and non-clickbait. They have used the headline-body similarity mainly for categorizing the clickbait article.

Chakraborty et al. analyzed the dataset linguistic using Stanford CoreNLP [5] to obtain the features used in the classification process using SVM, DT, and RF classification algorithms. The study utilized sentence structure, word patterns, language characteristics, and n-grams as the features. SVM is one of the supervised learning algorithms used for classification or regression [36] and to determine the maximum marginal hyperplane [37]. While DT uses tree concept to deduce classification rules based on practical examples [38]. Each node in the tree represents a parameter, while its branch represents a possible value of the connected top node [38]. Chakraborty et al. found different sentence structures in clickbait and non-clickbait titles. Clickbait titles tend to be wordy, contain stopwords and slangs, and hyperbolic with 40 frequent words used as the classification features [6]. Furthermore, Biyani et al. defined several types of clickbait titles using GBD [39]. The clickbait title is usually written in a hyperbole, ambiguous, or unclear statement that affects the emotions of the reader (increase the user's curiosity) [12].

**2.1 Existing system**

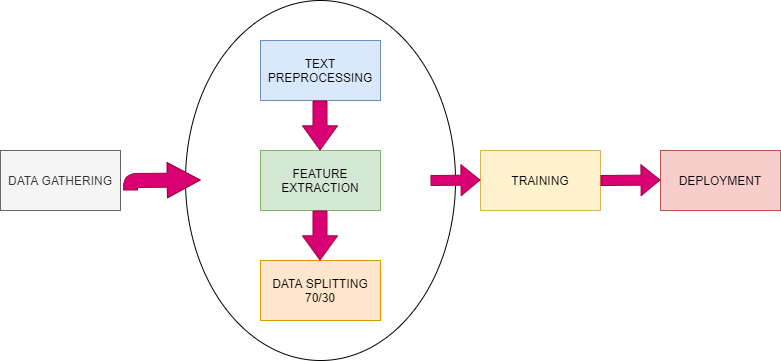
Existing system to detecting clickbait headlines uses the number of likes and shares in a news article on Facebook while other similar research uses Neural Network with Backpropagation algorithm.

**2.2 Proposed system**

We apply a simple mathematical machine learning classifier which is namely Naive, Bayes, to classify the problem into binary classes such as clickbait or not clickbait. Naïve Bayes works beautifully with fewer datasets to give the best results as seen in studies.

**CHAPTER 3**

**SYSTEM ARCHITECTURE**



**3.1 DATASET GATHERING**

The data has been collected from the Kaggle website which contains a huge number of datasets for different machine learning fields. As in the dataset’s official directory, The uploader has mentioned having collected the dataset from various news sites. The clickbait headlines are

collected from sites such as Buzzfeed, Upworthy, ViralNova, etc. We can apply different classification algorithms to classify the data into clickbait and non-clickbait.

The dataset contains two columns namely clickbait and headline. The headline column contains headlines written by actual users. The clickbait column contains two categories. 0 represents not clickbait and 1 represents clickbait.

The dataset contains unnecessary words which do not put any weight into our sentiment analysis, For our algorithm to perform better, These noises have to be processed.

Some examples of datasets.



Table 2.1.1: Example of dataset posts annotated with their corresponding label, 0 if it is not clickbait, 1 if it is clickbait.

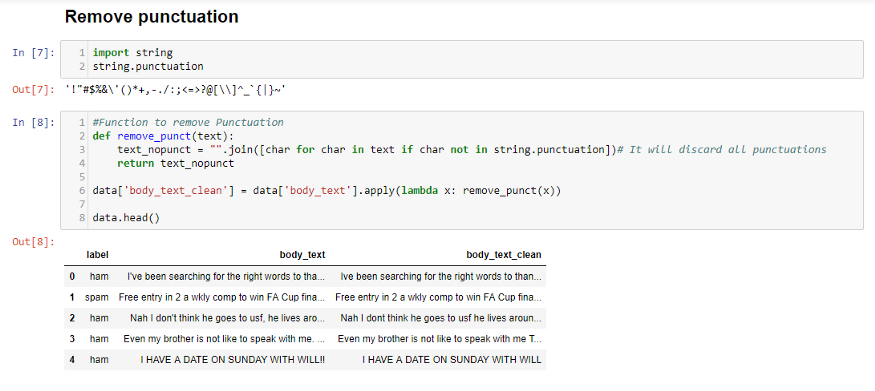
**3.2 TEXT PREPROCESSING**

Text preprocessing is a method to clean the text data and make it ready to feed data to the model. Text data contains noise in various forms like emotions, punctuation, text in different cases. When we talk about Human Language then, there are different ways to say the same thing, And this is only the main problem we have to deal with because machines will not understand words, they need numbers so we need to convert text to numbers in an efficient manner.

In this project, Various pre-processing techniques have been used.

* **Remove punctuations:** Punctuation can provide grammatical context to a sentence which supports our understanding. But for our vectorizer which counts the number of words and not the context, it does not add value, so we remove all special characters.

eg: How are you?->How are you



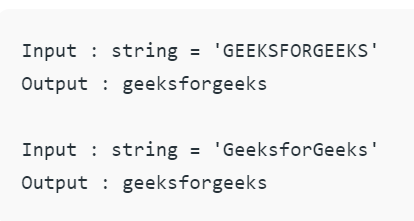
In body\_text\_clean, we can see that all punctuations like I’ve-> I’ve are omitted.

* **Remove Stopwords:** Stopwords are the most commonly occurring words in a text which do not provide any valuable information. stopwords like they, there, this, where, etc are some of the stopwords.



In body\_text\_nostop, all unnecessary words like been, for, the are removed.

* **Lower:**  All the alphabets are lowercase in case any new input appears to be uppercase and the classifier could not predict.



* **Tokenization:** Tokenizing separates text into units such as sentences or words. It gives structure to previously unstructured text.

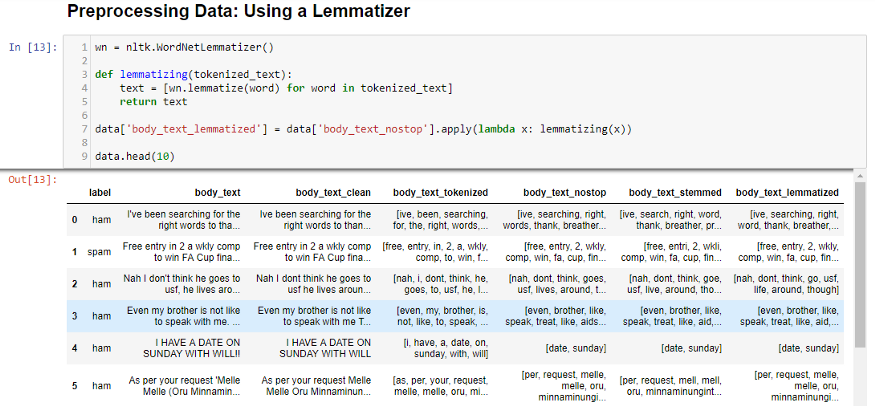
eg: Plata o Plomo-> ‘Plata’,’o’,’Plomo’.



In body\_text\_tokenized, we can see that all words are generated as tokens.

* **Lemmatization:** Lemmatizing derives the canonical form (‘lemma’) of a word. i.e the root form. It is better than stemming as it uses a dictionary-based approach i.e a morphological analysis to the root word.eg: Entitling, Entitled->Entitle

In Short, Stemming is typically faster as it simply chops off the end of the word, without understanding the context of the word. Lemmatizing is slower and more accurate as it takes an informed analysis with the context of the word in mind.



In body\_text\_stemmed, we can words like chances are lemmatized to chance whereas it is stemmed to chanc.

**3.2 FEATURE EXTRACTION**

Machine Learning algorithms learn from a predefined set of features from the training data to produce output for the test data. But the main problem in working with language processing is that machine learning algorithms cannot work on the raw text directly. So, we need some feature extraction techniques to convert text into a matrix(or vector) of features.

In this project, we use Countvectorizer or simply known as Bag of words. It is one of the most fundamental methods to transform tokens into a set of features. The BoW model is used in document classification, where each word is used as a feature for training the classifier.

**3.2.1 BAG OF WORDS**

A bag-of-words model, or BoW for short, is a way of extracting features from text for use in modeling, such as with machine learning algorithms. The approach is very simple and flexible, and can be used in a myriad of ways for extracting features from documents. A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:

* A vocabulary of known words.
* A measure of the presence of known words.

It is called a “*bag*” of words, because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document.

The intuition is that documents are similar if they have similar content. Further, that from the content alone we can learn something about the meaning of the document. The bag-of-words can be as simple or complex as you like. The complexity comes both in deciding how to design the vocabulary of known words (or tokens) and how to score the presence of known words.

## Example of the Bag-of-Words Model

Let’s make the bag-of-words model concrete with a worked example.

**Step 1: Collect Data**

Below is a snippet of the first few lines of text from the book “A Tale of Two Cities” by Charles Dickens, taken from Project Gutenberg.

* It was the best of times,
* it was the worst of times,
* it was the age of wisdom,
* it was the age of foolishness,

For this small example, let’s treat each line as a separate “document” and the 4 lines as our entire corpus of documents.

**Step 2: Design the Vocabulary**

Now we can make a list of all of the words in our model vocabulary.

The unique words here (ignoring case and punctuation) are:

* “it”
* “was”
* “the”
* “best”
* “of”
* “times”
* “worst”
* “age”
* “wisdom”
* “foolishness”

That is a vocabulary of 10 words from a corpus containing 24 words.

**Step 3: Create Document Vectors**

The next step is to score the words in each document.

The objective is to turn each document of free text into a vector that we can use as input or output for a machine learning model.

Because we know the vocabulary has 10 words, we can use a fixed-length document representation of 10, with one position in the vector to score each word.

The simplest scoring method is to mark the presence of words as a boolean value, 0 for absent, 1 for present.

Using the arbitrary ordering of words listed above in our vocabulary, we can step through the first document (“It was the best of times“) and convert it into a binary vector.

The scoring of the document would look as follows:

* “it” = 1
* “was” = 1
* “the” = 1
* “best” = 1
* “of” = 1
* “times” = 1
* “worst” = 0
* “age” = 0
* “wisdom” = 0
* “foolishness” = 0

As a binary vector, this would look as follows:

[1, 1, 1, 1, 1, 1, 0, 0, 0, 0]

The other three documents would look as follows:

* "it was the worst of times" = [1, 1, 1, 0, 1, 1, 1, 0, 0, 0]
* "it was the age of wisdom" = [1, 1, 1, 0, 1, 0, 0, 1, 1, 0]
* "it was the age of foolishness" = [1, 1, 1, 0, 1, 0, 0, 1, 0, 1]

All ordering of the words is nominally discarded and we have a consistent way of extracting features from any document in our corpus, ready for use in modeling.

New documents that overlap with the vocabulary of known words, but may contain words outside of the vocabulary, can still be encoded, where only the occurrence of known words are scored and unknown words are ignored.

You can see how this might naturally scale to large vocabularies and larger documents.

**Managing Vocabulary**

As the vocabulary size increases, so does the vector representation of documents.

In the previous example, the length of the document vector is equal to the number of known words.You can imagine that for a very large corpus, such as thousands of books, that the length of the vector might be thousands or millions of positions. Further, each document may contain very few of the known words in the vocabulary.

This results in a vector with lots of zero scores, called a sparse vector or sparse representation.

Sparse vectors require more memory and computational resources when modeling and the vast number of positions or dimensions can make the modeling process very challenging for traditional algorithms.

As such, there is pressure to decrease the size of the vocabulary when using a bag-of-words model.

There are simple text cleaning techniques that can be used as a first step, such as:

* Ignoring case
* Ignoring punctuation
* Ignoring frequent words that don’t contain much information, called stop words, like “a,” “of,” etc.
* Fixing misspelled words.
* Reducing words to their stem (e.g. “play” from “playing”) using stemming algorithms.

A more sophisticated approach is to create a vocabulary of grouped words. This both changes the scope of the vocabulary and allows the bag-of-words to capture a little bit more meaning from the document.

In this approach, each word or token is called a “gram”. Creating a vocabulary of two-word pairs is, in turn, called a bigram model. Again, only the bigrams that appear in the corpus are modeled, not all possible bigrams.

For example, the bigrams in the first line of text in the previous section: “It was the best of times” are as follows:

* “it was”
* “was the”
* “the best”
* “best of”
* “of times”

A vocabulary then tracks triplets of words is called a trigram model and the general approach is called the n-gram model, where n refers to the number of grouped words.

Often a simple bigram approach is better than a 1-gram bag-of-words model for tasks like documentation classification.

**Scoring Words**

Once a vocabulary has been chosen, the occurrence of words in example documents needs to be scored.

In the worked example, we have already seen one very simple approach to scoring: a binary scoring of the presence or absence of words.

Some additional simple scoring methods include:

* Counts. Count the number of times each word appears in a document.
* Frequencies. Calculate the frequency that each word appears in a document out of all the words in the document.

**Word Hashing**

You may remember from computer science that a hash function is a bit of math that maps data to a fixed size set of numbers.

For example, we use them in hash tables when programming where perhaps names are converted to numbers for fast lookup.

We can use a hash representation of known words in our vocabulary. This addresses the problem of having a very large vocabulary for a large text corpus because we can choose the size of the hash space, which is in turn the size of the vector representation of the document.

Words are hashed deterministically to the same integer index in the target hash space. A binary score or count can then be used to score the word.

This is called the “hash trick” or “feature hashing“.

The challenge is to choose a hash space to accommodate the chosen vocabulary size to minimize the probability of collisions and trade-off sparsity.

**TF-IDF**

A problem with scoring word frequency is that highly frequent words start to dominate in the document (e.g. larger score), but may not contain as much “informational content” to the model as rarer but perhaps domain specific words.

One approach is to rescale the frequency of words by how often they appear in all documents, so that the scores for frequent words like “the” that are also frequent across all documents are penalized.

This approach to scoring is called Term Frequency – Inverse Document Frequency, or TF-IDF for short, where:

* Term Frequency: is a scoring of the frequency of the word in the current document.
* Inverse Document Frequency: is a scoring of how rare the word is across documents.

The scores are a weighting where not all words are equally as important or interesting.

The scores have the effect of highlighting words that are distinct (contain useful information) in a given document.

**Limitations of Bag-of-Words**

The bag-of-words model is very simple to understand and implement and offers a lot of flexibility for customization on your specific text data.

It has been used with great success on prediction problems like language modeling and documentation classification.

Nevertheless, it suffers from some shortcomings, such as:

* **Vocabulary**: The vocabulary requires careful design, most specifically in order to manage the size, which impacts the sparsity of the document representations.
* **Sparsity**: Sparse representations are harder to model both for computational reasons (space and time complexity) and also for information reasons, where the challenge is for the models to harness so little information in such a large representational space.
* **Meaning**: Discarding word order ignores the context, and in turn meaning of words in the document (semantics). Context and meaning can offer a lot to the model, that if modeled could tell the difference between the same words differently arranged (“this is interesting” vs “is this interesting”), synonyms (“old bike” vs “used bike”), and much more.

**3.2.2 COUNT VECTORIZER WORKING**

Consider the documents-

* ‘This is first document’
* ‘This is the second document’
* ‘and this is the third’

Two steps are involved in Bag of words:

1. **FIT:** Learn a vocabulary dictionary of all tokens in the raw documents and returns the list of uniquely identified words present in the documents.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **and** | **document** | **first** | **is** | **second** | **the** | **third** | **this** |

1. **TRANSFORM:** Transform the documents to the document-term matrix and return a sparse matrix with the word present in the document as 1 and not present as 0.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **and** | **document** | **first** | **is** | **second** | **the** | **third** | **this** |
| **document1** | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| **document2** | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| **document3** | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

**3.3 DATASET SPLITTING**

Splitting your dataset is essential for an unbiased evaluation of prediction performance. In most cases, it’s enough to split your dataset randomly into [three subsets](https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets):

* The training set is applied to train, or fit, your model. For example, you use the training set to find the optimal weights, or coefficients, for [linear regression](https://realpython.com/linear-regression-in-python/), or [neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network).
* The test set is needed for an unbiased evaluation of the final model. You shouldn’t use it for fitting or validation.

The dataset splitting ratio has been kept different for Naive Bayes

**3.3.1 NAIVE BAYES DATASET SPLITTING**

* **Training Set:** 80%
* **Test Set:** 20%

**3.4 MODEL TRAINING**

Model training is the phase in the data science development lifecycle where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range. The purpose of model training is to build the best mathematical representation of the relationship between data features and a target label (in supervised learning [glossary link]) or among the features themselves (unsupervised learning [glossary link]). Loss functions [glossary link] are a critical aspect of model training since they define how to optimize the machine learning algorithms. Depending on the objective, type of data and algorithm, data science practitioner use different type of loss functions. One of the popular examples of loss functions is Mean Square Error (MSE)

Once we have applied the different steps of the preprocessing part, we can now focus on the machine learning part. There are three major models used in sentiment analysis to classify a sentence into positive or negative: and Naive Bayes. Let’s first introduce the Naive Bayes, which is well known for its simplicity and efficiency for text classification.

**3.4.1 NAIVE BAYES CLASSIFIER THEORY**

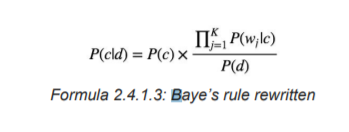
In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive)independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring several parameters linear in the number of variables (features/predictors) in a learning problem. Maximum Likelihood training can be done by evaluating a closed-form expression (a mathematical expression that can be evaluated in a finite number of operations), which takes linear time. It is based on the application of the Bayes rule given by the following formula:

P(C=c|D=d) = P(D=d|C=c) \* P(C=c) / P(D=d)

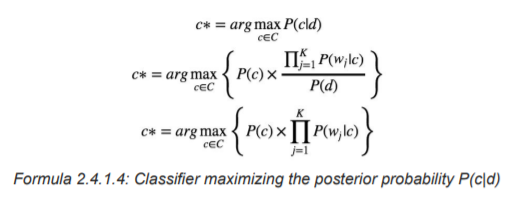
where D denotes the document and C the category (label), d and c are instances of D and C and P(D = d) = ∑ (D |C )P(C ). We can simplify this expression by,

P(c|d) = P(d|c)\*P(c)/P(d)

In our case, a headlined is represented by a vector of K attributes such as d = (w, w, ..., w ). Computing P(d|c) is not trivial and that's why the Naive Bayes introduces the assumption that all of the feature values wj are independent given the category label c. That is, for i =/ j , wi and wj are conditionally independent given the category label c. So the Bayes rule can be rewritten as



Based on this equation, maximum a posterior (MAP) classifier can be constructing by seeking the optimal category which maximizes the posterior P(c|d) :



Note that P(d) is removed since it is a constant for every category c.

There are several variants of Naive Bayes classifiers that are:

● The **Multi­variate Bernoulli Model**: Also called binomial model, useful if our feature vectors are binary (e.g 0s and 1s). An application can be text classification with a bag of words model where the 0s 1s are "word does not occur in the document" and "word occurs in the document" respectively.

● The **Multinomial Model**: Typically used for discrete counts. In-text classification, we extend the Bernoulli model further by counting the number of times a word $w\_i$ appears over the number of words rather than saying 0 or 1 if a word occurs or not.

● The **Gaussian Model**: We assume that features follow a normal distribution. Instead of discrete counts, we have continuous features.

For text classification, the most used and considered as the best choice is the Multinomial Naive Bayes.

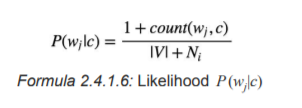
The prior distribution P(c) can be used to incorporate additional assumptions about the relative frequencies of classes. It is computed by:

**P(c) = Ni / N**

**Formula: Prior Distribution P(c)**

where N is the total number of training headlines and N is the number of training headlines in class I c.

The likelihood P(w |c) is usually computed using the formula:



where count(w, c) is the number of times that word occurs within the training headline-trained class j wj c, and |V | = ∑ the size of the vocabulary. This estimation uses the simplest smoothing j wj method to solve the zero probability problem that arises when our model encounters a word seen in the test set but not in the training set, Laplace or add­one since we use 1 as constant. We will see that Laplace smoothing method is not effective compared to other smoothing methods used in language models.

**3.4.1 NAIVE BAYES CLASSIFIER APPLICATION USING EXAMPLE**

**FORMULA:**

**P ( A | B ) = P( B | A ) \* P( A ) / P( B )**

**DATASET:**

|  |
| --- |
| **X = { x1, x2, x3 …….xn } { y }** |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **f1** | **f2** | **f3** | **…..** | **fn** | **y** | | **x1** | **x2** | **x3** | **…...** | **xn** | **y1** | |

**DERIVATION:**

|  |
| --- |
| **P ( y | x1, x2, x3…..xn ) = P( x1 | y )\* P( x1 | y )\* P( x1 | y ) ……..P( xn | y ) \* P(y)**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **P(x1)\* P(x2)\* P(x3)......P(xn) -> Constant for all**  **P ( y | x1, x2, x3…..xn ) = P(y) ∑ P(xi | y)**  **P ( y | x1, x2, x3…..xn ) = argmax ( P(y) ∑ P(xi | y) ) -> main formula for classifier** |

**FEATURES:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **the** | **food** | **delicious** | **bad** | **output** |
| **1** | **1** | **1** | **0** | **1(yes)** |
| **1** | **1** | **0** | **1** | **0(no)** |
| **0** | **1** | **0** | **1** | **0(no)** |
|  | | | | |
| **Sentence1** | **The Food is delicious** | | | |
| **Sentence2** | **The food is Bad** | | | |
| **Sentence3** | **Food is Bad** | | | |
|  | | | | |
| **To Classify**  **Sentence - Good(yes) or Bad(no)** | | | | |
| **For yes, p(y=yes) = 1/3, p(y=no) = 2/3**  **Sentence - delicious food** | | | | |
| **p(y=yes) \* p(x1|y=yes) \* p(x2|y=yes)**  **⅓ \* 1 \* ⅓** | | | | |
| **For no** | | | | |
| **p(y=yes) \* p(x1|y=yes) \* p(x2|y=yes)**  **⅔ \* 0 \* ⅔** | | | | |
| **The higher the value of the probability of yes or no, the output would be selected from there.** | | | | |

**3.4 CODING**

**index.html**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Clickbait analysis</title>**

**</head>**

**<body>**

**<h1>**

**CLICKBAIT HEADLINE CHECKER**

**</h1>**

**<form action="/" method="POST">**

**<textarea name="message" style="outline: none;" placeholder="type your headline..." rows="6" cols="50" required></textarea>**

**<br/>**

**<br/>**

**<input type="submit" value="Check">**

**</form>**

**{% if my\_nb\_prediction == 1%}**

**<h2>Click Bait</h2>**

**{% elif my\_nb\_prediction == 0%}**

**<h2>Not Click Bait</h2>**

**{% endif %}**

**</body>**

**</html>**

**app.py**

**from flask import Flask,render\_template,request**

**import pickle**

**# load the model from disk**

**filename = 'model/clickbait\_model.pkl'**

**naive\_bayes\_model = pickle.load(open(filename, 'rb'))**

**cv\_naive=pickle.load(open('model/clickbait\_transform.pkl','rb'))**

**app = Flask(\_\_name\_\_)**

**@app.route('/',methods=['POST','GET'])**

**def home():**

**if request.method == 'POST':**

**message = request.form['message'].lower()**

**data = [message]**

**vect\_naive = cv\_naive.transform(data).toarray()**

**# for naive bayes classifier**

**my\_nb\_prediction = naive\_bayes\_model.predict(vect\_naive)[0]**

**print(my\_nb\_prediction)**

**return render\_template('index.html',my\_nb\_prediction = my\_nb\_prediction)**

**return render\_template('index.html')**

**if \_\_name\_\_ == '\_\_main\_\_':**

**app.run(debug=True)**

**naivebayes.ipynb**

|  |
| --- |
| **import pandas as pd data = pd.read\_csv('../dataset/clickbait\_data.csv'); import nltk import re from nltk.corpus import stopwords from nltk.stem import WordNetLemmatizer # Lemmatization object ps = WordNetLemmatizer() corpus = [] # Text preprocessing # keep only text based # lower all the letters # split the words # for i in range(0,len(data)): # review = re.sub('[^a-zA-Z0-9]'," ",data['headline'][i]) # review = review.lower() # review = review.split() # review = [ps.lemmatize(word) for word in review if not word in stopwords.words('english')] # review = ' '.join(review) # corpus.append(review) # pd.DataFrame(corpus).to\_csv('processed\_data.csv'); clean\_data = pd.read\_csv('processed\_data.csv') corpus\_clean\_data = clean\_data.to\_numpy().flatten() corpus\_clean\_data\_usable = corpus\_clean\_data.tolist() from sklearn.feature\_extraction.text import CountVectorizer cv = CountVectorizer(ngram\_range=(1,2),max\_features=80000) X = cv.fit\_transform(corpus\_clean\_data\_usable) y = data['clickbait'].to\_numpy() #train test split from sklearn.model\_selection import train\_test\_split X\_train, X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.20) #training model using Naive bayes classifier  from sklearn.naive\_bayes import MultinomialNB model = MultinomialNB().fit(X\_train,y\_train)  # predict  y\_pred = model.predict(X\_test) from sklearn.metrics import confusion\_matrix confusion\_matrix(y\_test, y\_pred)**  **-> array([[3073, 136],**  **[ 100, 3091]], dtype=int64)**  **from sklearn.metrics import accuracy\_score accuracy\_score(y\_test,y\_pred)**  **-> 0.963125**  **# Checking the training model with custom input data negative\_message = "Bill Changing Credit Card Rules Is Sent to Obama With Gun Measure Included" message = [negative\_message] vect = cv.transform(message).toarray() my\_prediction = model.predict(vect) my\_prediction\_prob = model.predict\_proba(vect) if my\_prediction==1:  print("clickbait")  print(my\_prediction\_prob[0][1]\*100) else:  print("not clickbait")  print(my\_prediction\_prob[0][0]\*100)**  **-> not clickbait**  **-> 99.99921878962151**  **# Dump the machine learning model outsite so you can use outsite and not retrain again and again # import pickle # filename = 'clickbait\_model.pkl' # pickle.dump(model, open(filename, 'wb')) # pickle.dump(cv, open('clickbait\_transform.pkl', 'wb'))** |

**requirements.txt**

|  |
| --- |
| **Flask Werkzeug gunicorn itsdangerous Jinja2 MarkupSafe scikit-learn numpy pandas** |

**w3.css**

**/\* W3.CSS 4.15 December 2020 by Jan Egil and Borge Refsnes \*/**

**html{box-sizing:border-box}\*,\*:before,\*:after{box-sizing:inherit}**

**/\* Extract from normalize.css by Nicolas Gallagher and Jonathan Neal git.io/normalize \*/**

**html{-ms-text-size-adjust:100%;-webkit-text-size-adjust:100%}body{margin:0}**

**article,aside,details,figcaption,figure,footer,header,main,menu,nav,section{display:block}summary{display:list-item}**

**audio,canvas,progress,video{display:inline-block}progress{vertical-align:baseline}**

**audio:not([controls]){display:none;height:0}[hidden],template{display:none}**

**a{background-color:transparent}a:active,a:hover{outline-width:0}**

**abbr[title]{border-bottom:none;text-decoration:underline;text-decoration:underline dotted}**

**b,strong{font-weight:bolder}dfn{font-style:italic}mark{background:#ff0;color:#000}**

**small{font-size:80%}sub,sup{font-size:75%;line-height:0;position:relative;vertical-align:baseline}**

**sub{bottom:-0.25em}sup{top:-0.5em}figure{margin:1em 40px}img{border-style:none}**

**code,kbd,pre,samp{font-family:monospace,monospace;font-size:1em}hr{box-sizing:content-box;height:0;overflow:visible}**

**button,input,select,textarea,optgroup{font:inherit;margin:0}optgroup{font-weight:bold}**

**button,input{overflow:visible}button,select{text-transform:none}**

**button,[type=button],[type=reset],[type=submit]{-webkit-appearance:button}**

**button::-moz-focus-inner,[type=button]::-moz-focus-inner,[type=reset]::-moz-focus-inner,[type=submit]::-moz-focus-inner{border-style:none;padding:0}**

**button:-moz-focusring,[type=button]:-moz-focusring,[type=reset]:-moz-focusring,[type=submit]:-moz-focusring{outline:1px dotted ButtonText}**

**fieldset{border:1px solid #c0c0c0;margin:0 2px;padding:.35em .625em .75em}**

**legend{color:inherit;display:table;max-width:100%;padding:0;white-space:normal}textarea{overflow:auto}**

**[type=checkbox],[type=radio]{padding:0}**

**[type=number]::-webkit-inner-spin-button,[type=number]::-webkit-outer-spin-button{height:auto}**

**[type=search]{-webkit-appearance:textfield;outline-offset:-2px}**

**[type=search]::-webkit-search-decoration{-webkit-appearance:none}**

**::-webkit-file-upload-button{-webkit-appearance:button;font:inherit}**

**/\* End extract \*/**

**html,body{font-family:Verdana,sans-serif;font-size:15px;line-height:1.5}html{overflow-x:hidden}**

**h1{font-size:36px}h2{font-size:30px}h3{font-size:24px}h4{font-size:20px}h5{font-size:18px}h6{font-size:16px}**

**.w3-serif{font-family:serif}.w3-sans-serif{font-family:sans-serif}.w3-cursive{font-family:cursive}.w3-monospace{font-family:monospace}**

**h1,h2,h3,h4,h5,h6{font-family:"Segoe UI",Arial,sans-serif;font-weight:400;margin:10px 0}.w3-wide{letter-spacing:4px}**

**hr{border:0;border-top:1px solid #eee;margin:20px 0}**

**.w3-image{max-width:100%;height:auto}img{vertical-align:middle}a{color:inherit}**

**.w3-table,.w3-table-all{border-collapse:collapse;border-spacing:0;width:100%;display:table}.w3-table-all{border:1px solid #ccc}**

**.w3-bordered tr,.w3-table-all tr{border-bottom:1px solid #ddd}.w3-striped tbody tr:nth-child(even){background-color:#f1f1f1}**

**.w3-table-all tr:nth-child(odd){background-color:#fff}.w3-table-all tr:nth-child(even){background-color:#f1f1f1}**

**.w3-hoverable tbody tr:hover,.w3-ul.w3-hoverable li:hover{background-color:#ccc}.w3-centered tr th,.w3-centered tr td{text-align:center}**

**.w3-table td,.w3-table th,.w3-table-all td,.w3-table-all th{padding:8px 8px;display:table-cell;text-align:left;vertical-align:top}**

**.w3-table th:first-child,.w3-table td:first-child,.w3-table-all th:first-child,.w3-table-all td:first-child{padding-left:16px}**

**.w3-btn,.w3-button{border:none;display:inline-block;padding:8px 16px;vertical-align:middle;overflow:hidden;text-decoration:none;color:inherit;background-color:inherit;text-align:center;cursor:pointer;white-space:nowrap}**

**.w3-btn:hover{box-shadow:0 8px 16px 0 rgba(0,0,0,0.2),0 6px 20px 0 rgba(0,0,0,0.19)}**

**.w3-btn,.w3-button{-webkit-touch-callout:none;-webkit-user-select:none;-khtml-user-select:none;-moz-user-select:none;-ms-user-select:none;user-select:none}**

**.w3-disabled,.w3-btn:disabled,.w3-button:disabled{cursor:not-allowed;opacity:0.3}.w3-disabled \*,:disabled \*{pointer-events:none}**

**.w3-btn.w3-disabled:hover,.w3-btn:disabled:hover{box-shadow:none}**

**.w3-badge,.w3-tag{background-color:#000;color:#fff;display:inline-block;padding-left:8px;padding-right:8px;text-align:center}.w3-badge{border-radius:50%}**

**.w3-ul{list-style-type:none;padding:0;margin:0}.w3-ul li{padding:8px 16px;border-bottom:1px solid #ddd}.w3-ul li:last-child{border-bottom:none}**

**.w3-tooltip,.w3-display-container{position:relative}.w3-tooltip .w3-text{display:none}.w3-tooltip:hover .w3-text{display:inline-block}**

**.w3-ripple:active{opacity:0.5}.w3-ripple{transition:opacity 0s}**

**.w3-input{padding:8px;display:block;border:none;border-bottom:1px solid #ccc;width:100%}**

**.w3-select{padding:9px 0;width:100%;border:none;border-bottom:1px solid #ccc}**

**.w3-dropdown-click,.w3-dropdown-hover{position:relative;display:inline-block;cursor:pointer}**

**.w3-dropdown-hover:hover .w3-dropdown-content{display:block}**

**.w3-dropdown-hover:first-child,.w3-dropdown-click:hover{background-color:#ccc;color:#000}**

**.w3-dropdown-hover:hover > .w3-button:first-child,.w3-dropdown-click:hover > .w3-button:first-child{background-color:#ccc;color:#000}**

**.w3-dropdown-content{cursor:auto;color:#000;background-color:#fff;display:none;position:absolute;min-width:160px;margin:0;padding:0;z-index:1}**

**.w3-check,.w3-radio{width:24px;height:24px;position:relative;top:6px}**

**.w3-sidebar{height:100%;width:200px;background-color:#fff;position:fixed!important;z-index:1;overflow:auto}**

**.w3-bar-block .w3-dropdown-hover,.w3-bar-block .w3-dropdown-click{width:100%}**

**.w3-bar-block .w3-dropdown-hover .w3-dropdown-content,.w3-bar-block .w3-dropdown-click .w3-dropdown-content{min-width:100%}**

**.w3-bar-block .w3-dropdown-hover .w3-button,.w3-bar-block .w3-dropdown-click .w3-button{width:100%;text-align:left;padding:8px 16px}**

**.w3-main,#main{transition:margin-left .4s}**

**.w3-modal{z-index:3;display:none;padding-top:100px;position:fixed;left:0;top:0;width:100%;height:100%;overflow:auto;background-color:rgb(0,0,0);background-color:rgba(0,0,0,0.4)}**

**.w3-modal-content{margin:auto;background-color:#fff;position:relative;padding:0;outline:0;width:600px}**

**.w3-bar{width:100%;overflow:hidden}.w3-center .w3-bar{display:inline-block;width:auto}**

**.w3-bar .w3-bar-item{padding:8px 16px;float:left;width:auto;border:none;display:block;outline:0}**

**.w3-bar .w3-dropdown-hover,.w3-bar .w3-dropdown-click{position:static;float:left}**

**.w3-bar .w3-button{white-space:normal}**

**.w3-bar-block .w3-bar-item{width:100%;display:block;padding:8px 16px;text-align:left;border:none;white-space:normal;float:none;outline:0}**

**.w3-bar-block.w3-center .w3-bar-item{text-align:center}.w3-block{display:block;width:100%}**

**.w3-responsive{display:block;overflow-x:auto}**

**.w3-container:after,.w3-container:before,.w3-panel:after,.w3-panel:before,.w3-row:after,.w3-row:before,.w3-row-padding:after,.w3-row-padding:before,**

**.w3-cell-row:before,.w3-cell-row:after,.w3-clear:after,.w3-clear:before,.w3-bar:before,.w3-bar:after{content:"";display:table;clear:both}**

**.w3-col,.w3-half,.w3-third,.w3-twothird,.w3-threequarter,.w3-quarter{float:left;width:100%}**

**.w3-col.s1{width:8.33333%}.w3-col.s2{width:16.66666%}.w3-col.s3{width:24.99999%}.w3-col.s4{width:33.33333%}**

**.w3-col.s5{width:41.66666%}.w3-col.s6{width:49.99999%}.w3-col.s7{width:58.33333%}.w3-col.s8{width:66.66666%}**

**.w3-col.s9{width:74.99999%}.w3-col.s10{width:83.33333%}.w3-col.s11{width:91.66666%}.w3-col.s12{width:99.99999%}**

**@media (min-width:601px){.w3-col.m1{width:8.33333%}.w3-col.m2{width:16.66666%}.w3-col.m3,.w3-quarter{width:24.99999%}.w3-col.m4,.w3-third{width:33.33333%}**

**.w3-col.m5{width:41.66666%}.w3-col.m6,.w3-half{width:49.99999%}.w3-col.m7{width:58.33333%}.w3-col.m8,.w3-twothird{width:66.66666%}**

**.w3-col.m9,.w3-threequarter{width:74.99999%}.w3-col.m10{width:83.33333%}.w3-col.m11{width:91.66666%}.w3-col.m12{width:99.99999%}}**

**@media (min-width:993px){.w3-col.l1{width:8.33333%}.w3-col.l2{width:16.66666%}.w3-col.l3{width:24.99999%}.w3-col.l4{width:33.33333%}**

**.w3-col.l5{width:41.66666%}.w3-col.l6{width:49.99999%}.w3-col.l7{width:58.33333%}.w3-col.l8{width:66.66666%}**

**.w3-col.l9{width:74.99999%}.w3-col.l10{width:83.33333%}.w3-col.l11{width:91.66666%}.w3-col.l12{width:99.99999%}}**

**.w3-rest{overflow:hidden}.w3-stretch{margin-left:-16px;margin-right:-16px}**

**.w3-content,.w3-auto{margin-left:auto;margin-right:auto}.w3-content{max-width:980px}.w3-auto{max-width:1140px}**

**.w3-cell-row{display:table;width:100%}.w3-cell{display:table-cell}**

**.w3-cell-top{vertical-align:top}.w3-cell-middle{vertical-align:middle}.w3-cell-bottom{vertical-align:bottom}**

**.w3-hide{display:none!important}.w3-show-block,.w3-show{display:block!important}.w3-show-inline-block{display:inline-block!important}**

**@media (max-width:1205px){.w3-auto{max-width:95%}}**

**@media (max-width:600px){.w3-modal-content{margin:0 10px;width:auto!important}.w3-modal{padding-top:30px}**

**.w3-dropdown-hover.w3-mobile .w3-dropdown-content,.w3-dropdown-click.w3-mobile .w3-dropdown-content{position:relative}**

**.w3-hide-small{display:none!important}.w3-mobile{display:block;width:100%!important}.w3-bar-item.w3-mobile,.w3-dropdown-hover.w3-mobile,.w3-dropdown-click.w3-mobile{text-align:center}**

**.w3-dropdown-hover.w3-mobile,.w3-dropdown-hover.w3-mobile .w3-btn,.w3-dropdown-hover.w3-mobile .w3-button,.w3-dropdown-click.w3-mobile,.w3-dropdown-click.w3-mobile .w3-btn,.w3-dropdown-click.w3-mobile .w3-button{width:100%}}**

**@media (max-width:768px){.w3-modal-content{width:500px}.w3-modal{padding-top:50px}}**

**@media (min-width:993px){.w3-modal-content{width:900px}.w3-hide-large{display:none!important}.w3-sidebar.w3-collapse{display:block!important}}**

**@media (max-width:992px) and (min-width:601px){.w3-hide-medium{display:none!important}}**

**@media (max-width:992px){.w3-sidebar.w3-collapse{display:none}.w3-main{margin-left:0!important;margin-right:0!important}.w3-auto{max-width:100%}}**

**.w3-top,.w3-bottom{position:fixed;width:100%;z-index:1}.w3-top{top:0}.w3-bottom{bottom:0}**

**.w3-overlay{position:fixed;display:none;width:100%;height:100%;top:0;left:0;right:0;bottom:0;background-color:rgba(0,0,0,0.5);z-index:2}**

**.w3-display-topleft{position:absolute;left:0;top:0}.w3-display-topright{position:absolute;right:0;top:0}**

**.w3-display-bottomleft{position:absolute;left:0;bottom:0}.w3-display-bottomright{position:absolute;right:0;bottom:0}**

**.w3-display-middle{position:absolute;top:50%;left:50%;transform:translate(-50%,-50%);-ms-transform:translate(-50%,-50%)}**

**.w3-display-left{position:absolute;top:50%;left:0%;transform:translate(0%,-50%);-ms-transform:translate(-0%,-50%)}**

**.w3-display-right{position:absolute;top:50%;right:0%;transform:translate(0%,-50%);-ms-transform:translate(0%,-50%)}**

**.w3-display-topmiddle{position:absolute;left:50%;top:0;transform:translate(-50%,0%);-ms-transform:translate(-50%,0%)}**

**.w3-display-bottommiddle{position:absolute;left:50%;bottom:0;transform:translate(-50%,0%);-ms-transform:translate(-50%,0%)}**

**.w3-display-container:hover .w3-display-hover{display:block}.w3-display-container:hover span.w3-display-hover{display:inline-block}.w3-display-hover{display:none}**

**.w3-display-position{position:absolute}**

**.w3-circle{border-radius:50%}**

**.w3-round-small{border-radius:2px}.w3-round,.w3-round-medium{border-radius:4px}.w3-round-large{border-radius:8px}.w3-round-xlarge{border-radius:16px}.w3-round-xxlarge{border-radius:32px}**

**.w3-row-padding,.w3-row-padding>.w3-half,.w3-row-padding>.w3-third,.w3-row-padding>.w3-twothird,.w3-row-padding>.w3-threequarter,.w3-row-padding>.w3-quarter,.w3-row-padding>.w3-col{padding:0 8px}**

**.w3-container,.w3-panel{padding:0.01em 16px}.w3-panel{margin-top:16px;margin-bottom:16px}**

**.w3-code,.w3-codespan{font-family:Consolas,"courier new";font-size:16px}**

**.w3-code{width:auto;background-color:#fff;padding:8px 12px;border-left:4px solid #4CAF50;word-wrap:break-word}**

**.w3-codespan{color:crimson;background-color:#f1f1f1;padding-left:4px;padding-right:4px;font-size:110%}**

**.w3-card,.w3-card-2{box-shadow:0 2px 5px 0 rgba(0,0,0,0.16),0 2px 10px 0 rgba(0,0,0,0.12)}**

**.w3-card-4,.w3-hover-shadow:hover{box-shadow:0 4px 10px 0 rgba(0,0,0,0.2),0 4px 20px 0 rgba(0,0,0,0.19)}**

**.w3-spin{animation:w3-spin 2s infinite linear}@keyframes w3-spin{0%{transform:rotate(0deg)}100%{transform:rotate(359deg)}}**

**.w3-animate-fading{animation:fading 10s infinite}@keyframes fading{0%{opacity:0}50%{opacity:1}100%{opacity:0}}**

**.w3-animate-opacity{animation:opac 0.8s}@keyframes opac{from{opacity:0} to{opacity:1}}**

**.w3-animate-top{position:relative;animation:animatetop 0.4s}@keyframes animatetop{from{top:-300px;opacity:0} to{top:0;opacity:1}}**

**.w3-animate-left{position:relative;animation:animateleft 0.4s}@keyframes animateleft{from{left:-300px;opacity:0} to{left:0;opacity:1}}**

**.w3-animate-right{position:relative;animation:animateright 0.4s}@keyframes animateright{from{right:-300px;opacity:0} to{right:0;opacity:1}}**

**.w3-animate-bottom{position:relative;animation:animatebottom 0.4s}@keyframes animatebottom{from{bottom:-300px;opacity:0} to{bottom:0;opacity:1}}**

**.w3-animate-zoom {animation:animatezoom 0.6s}@keyframes animatezoom{from{transform:scale(0)} to{transform:scale(1)}}**

**.w3-animate-input{transition:width 0.4s ease-in-out}.w3-animate-input:focus{width:100%!important}**

**.w3-opacity,.w3-hover-opacity:hover{opacity:0.60}.w3-opacity-off,.w3-hover-opacity-off:hover{opacity:1}**

**.w3-opacity-max{opacity:0.25}.w3-opacity-min{opacity:0.75}**

**.w3-greyscale-max,.w3-grayscale-max,.w3-hover-greyscale:hover,.w3-hover-grayscale:hover{filter:grayscale(100%)}**

**.w3-greyscale,.w3-grayscale{filter:grayscale(75%)}.w3-greyscale-min,.w3-grayscale-min{filter:grayscale(50%)}**

**.w3-sepia{filter:sepia(75%)}.w3-sepia-max,.w3-hover-sepia:hover{filter:sepia(100%)}.w3-sepia-min{filter:sepia(50%)}**

**.w3-tiny{font-size:10px!important}.w3-small{font-size:12px!important}.w3-medium{font-size:15px!important}.w3-large{font-size:18px!important}**

**.w3-xlarge{font-size:24px!important}.w3-xxlarge{font-size:36px!important}.w3-xxxlarge{font-size:48px!important}.w3-jumbo{font-size:64px!important}**

**.w3-left-align{text-align:left!important}.w3-right-align{text-align:right!important}.w3-justify{text-align:justify!important}.w3-center{text-align:center!important}**

**.w3-border-0{border:0!important}.w3-border{border:1px solid #ccc!important}**

**.w3-border-top{border-top:1px solid #ccc!important}.w3-border-bottom{border-bottom:1px solid #ccc!important}**

**.w3-border-left{border-left:1px solid #ccc!important}.w3-border-right{border-right:1px solid #ccc!important}**

**.w3-topbar{border-top:6px solid #ccc!important}.w3-bottombar{border-bottom:6px solid #ccc!important}**

**.w3-leftbar{border-left:6px solid #ccc!important}.w3-rightbar{border-right:6px solid #ccc!important}**

**.w3-section,.w3-code{margin-top:16px!important;margin-bottom:16px!important}**

**.w3-margin{margin:16px!important}.w3-margin-top{margin-top:16px!important}.w3-margin-bottom{margin-bottom:16px!important}**

**.w3-margin-left{margin-left:16px!important}.w3-margin-right{margin-right:16px!important}**

**.w3-padding-small{padding:4px 8px!important}.w3-padding{padding:8px 16px!important}.w3-padding-large{padding:12px 24px!important}**

**.w3-padding-16{padding-top:16px!important;padding-bottom:16px!important}.w3-padding-24{padding-top:24px!important;padding-bottom:24px!important}**

**.w3-padding-32{padding-top:32px!important;padding-bottom:32px!important}.w3-padding-48{padding-top:48px!important;padding-bottom:48px!important}**

**.w3-padding-64{padding-top:64px!important;padding-bottom:64px!important}**

**.w3-padding-top-64{padding-top:64px!important}.w3-padding-top-48{padding-top:48px!important}**

**.w3-padding-top-32{padding-top:32px!important}.w3-padding-top-24{padding-top:24px!important}**

**.w3-left{float:left!important}.w3-right{float:right!important}**

**.w3-button:hover{color:#000!important;background-color:#ccc!important}**

**.w3-transparent,.w3-hover-none:hover{background-color:transparent!important}**

**.w3-hover-none:hover{box-shadow:none!important}**

**/\* Colors \*/**

**.w3-amber,.w3-hover-amber:hover{color:#000!important;background-color:#ffc107!important}**

**.w3-aqua,.w3-hover-aqua:hover{color:#000!important;background-color:#00ffff!important}**

**.w3-blue,.w3-hover-blue:hover{color:#fff!important;background-color:#2196F3!important}**

**.w3-light-blue,.w3-hover-light-blue:hover{color:#000!important;background-color:#87CEEB!important}**

**.w3-brown,.w3-hover-brown:hover{color:#fff!important;background-color:#795548!important}**

**.w3-cyan,.w3-hover-cyan:hover{color:#000!important;background-color:#00bcd4!important}**

**.w3-blue-grey,.w3-hover-blue-grey:hover,.w3-blue-gray,.w3-hover-blue-gray:hover{color:#fff!important;background-color:#607d8b!important}**

**.w3-green,.w3-hover-green:hover{color:#fff!important;background-color:#4CAF50!important}**

**.w3-light-green,.w3-hover-light-green:hover{color:#000!important;background-color:#8bc34a!important}**

**.w3-indigo,.w3-hover-indigo:hover{color:#fff!important;background-color:#3f51b5!important}**

**.w3-khaki,.w3-hover-khaki:hover{color:#000!important;background-color:#f0e68c!important}**

**.w3-lime,.w3-hover-lime:hover{color:#000!important;background-color:#cddc39!important}**

**.w3-orange,.w3-hover-orange:hover{color:#000!important;background-color:#ff9800!important}**

.w3-card,.w3-card-2{box-shadow:0 2px 5px 0 rgba(0,0,0,0.16),0 2px 10px 0 rgba(0,0,0,0.12)}

.w3-card-4,.w3-hover-shadow:hover{box-shadow:0 4px 10px 0 rgba(0,0,0,0.2),0 4px 20px 0 rgba(0,0,0,0.19)}

.w3-spin{animation:w3-spin 2s infinite linear}@keyframes w3-spin{0%{transform:rotate(0deg)}100%{transform:rotate(359deg)}}

.w3-animate-fading{animation:fading 10s infinite}@keyframes fading{0%{opacity:0}50%{opacity:1}100%{opacity:0}}

.w3-animate-opacity{animation:opac 0.8s}@keyframes opac{from{opacity:0} to{opacity:1}}

.w3-animate-top{position:relative;animation:animatetop 0.4s}@keyframes animatetop{from{top:-300px;opacity:0} to{top:0;opacity:1}}

.w3-animate-left{position:relative;animation:animateleft 0.4s}@keyframes animateleft{from{left:-300px;opacity:0} to{left:0;opacity:1}}

.w3-animate-right{position:relative;animation:animateright 0.4s}@keyframes animateright{from{right:-300px;opacity:0} to{right:0;opacity:1}}

.w3-animate-bottom{position:relative;animation:animatebottom 0.4s}@keyframes animatebottom{from{bottom:-300px;opacity:0} to{bottom:0;opacity:1}}

.w3-animate-zoom {animation:animatezoom 0.6s}@keyframes animatezoom{from{transform:scale(0)} to{transform:scale(1)}}

.w3-animate-input{transition:width 0.4s ease-in-out}.w3-animate-input:focus{width:100%!important}

.w3-opacity,.w3-hover-opacity:hover{opacity:0.60}.w3-opacity-off,.w3-hover-opacity-off:hover{opacity:1}

.w3-opacity-max{opacity:0.25}.w3-opacity-min{opacity:0.75}

.w3-greyscale-max,.w3-grayscale-max,.w3-hover-greyscale:hover,.w3-hover-grayscale:hover{filter:grayscale(100%)}

.w3-greyscale,.w3-grayscale{filter:grayscale(75%)}.w3-greyscale-min,.w3-grayscale-min{filter:grayscale(50%)}

.w3-sepia{filter:sepia(75%)}.w3-sepia-max,.w3-hover-sepia:hover{filter:sepia(100%)}.w3-sepia-min{filter:sepia(50%)}

.w3-tiny{font-size:10px!important}.w3-small{font-size:12px!important}.w3-medium{font-size:15px!important}.w3-large{font-size:18px!important}

.w3-xlarge{font-size:24px!important}.w3-xxlarge{font-size:36px!important}.w3-xxxlarge{font-size:48px!important}.w3-jumbo{font-size:64px!important}

.w3-left-align{text-align:left!important}.w3-right-align{text-align:right!important}.w3-justify{text-align:justify!important}.w3-center{text-align:center!important}

.w3-border-0{border:0!important}.w3-border{border:1px solid #ccc!important}

.w3-border-top{border-top:1px solid #ccc!important}.w3-border-bottom{border-bottom:1px solid #ccc!important}

.w3-border-left{border-left:1px solid #ccc!important}.w3-border-right{border-right:1px solid #ccc!important}

.w3-topbar{border-top:6px solid #ccc!important}.w3-bottombar{border-bottom:6px solid #ccc!important}

.w3-leftbar{border-left:6px solid #ccc!important}.w3-rightbar{border-right:6px solid #ccc!important}

.w3-section,.w3-code{margin-top:16px!important;margin-bottom:16px!important}

.w3-margin{margin:16px!important}.w3-margin-top{margin-top:16px!important}.w3-margin-bottom{margin-bottom:16px!important}

.w3-margin-left{margin-left:16px!important}.w3-margin-right{margin-right:16px!important}

.w3-padding-small{padding:4px 8px!important}.w3-padding{padding:8px 16px!important}.w3-padding-large{padding:12px 24px!important}

.w3-padding-16{padding-top:16px!important;padding-bottom:16px!important}.w3-padding-24{padding-top:24px!important;padding-bottom:24px!important}

.w3-padding-32{padding-top:32px!important;padding-bottom:32px!important}.w3-padding-48{padding-top:48px!important;padding-bottom:48px!important}

.w3-padding-64{padding-top:64px!important;padding-bottom:64px!important}

.w3-padding-top-64{padding-top:64px!important}.w3-padding-top-48{padding-top:48px!important}

.w3-padding-top-32{padding-top:32px!important}.w3-padding-top-24{padding-top:24px!important}

.w3-left{float:left!important}.w3-right{float:right!important}

.w3-button:hover{color:#000!important;background-color:#ccc!important}

.w3-transparent,.w3-hover-none:hover{background-color:transparent!important}

.w3-hover-none:hover{box-shadow:none!important}

/\* Colors \*/

.w3-amber,.w3-hover-amber:hover{color:#000!important;background-color:#ffc107!important}

.w3-aqua,.w3-hover-aqua:hover{color:#000!important;background-color:#00ffff!important}

.w3-blue,.w3-hover-blue:hover{color:#fff!important;background-color:#2196F3!important}

.w3-light-blue,.w3-hover-light-blue:hover{color:#000!important;background-color:#87CEEB!important}

.w3-brown,.w3-hover-brown:hover{color:#fff!important;background-color:#795548!important}

.w3-cyan,.w3-hover-cyan:hover{color:#000!important;background-color:#00bcd4!important}

.w3-blue-grey,.w3-hover-blue-grey:hover,.w3-blue-gray,.w3-hover-blue-gray:hover{color:#fff!important;background-color:#607d8b!important}

.w3-green,.w3-hover-green:hover{color:#fff!important;background-color:#4CAF50!important}

.w3-light-green,.w3-hover-light-green:hover{color:#000!important;background-color:#8bc34a!important}

.w3-indigo,.w3-hover-indigo:hover{color:#fff!important;background-color:#3f51b5!important}

.w3-khaki,.w3-hover-khaki:hover{color:#000!important;background-color:#f0e68c!important}

.w3-lime,.w3-hover-lime:hover{color:#000!important;background-color:#cddc39!important}

.w3-orange,.w3-hover-orange:hover{color:#000!important;background-color:#ff9800!important}

.w3-deep-orange,.w3-hover-deep-orange:hover{color:#fff!important;background-color:#ff5722!important}

.w3-pink,.w3-hover-pink:hover{color:#fff!important;background-color:#e91e63!important}

.w3-purple,.w3-hover-purple:hover{color:#fff!important;background-color:#9c27b0!important}

.w3-deep-purple,.w3-hover-deep-purple:hover{color:#fff!important;background-color:#673ab7!important}

.w3-red,.w3-hover-red:hover{color:#fff!important;background-color:#f44336!important}

.w3-sand,.w3-hover-sand:hover{color:#000!important;background-color:#fdf5e6!important}

.w3-teal,.w3-hover-teal:hover{color:#fff!important;background-color:#009688!important}

.w3-yellow,.w3-hover-yellow:hover{color:#000!important;background-color:#ffeb3b!important}

.w3-white,.w3-hover-white:hover{color:#000!important;background-color:#fff!important}

.w3-black,.w3-hover-black:hover{color:#fff!important;background-color:#000!important}

.w3-grey,.w3-hover-grey:hover,.w3-gray,.w3-hover-gray:hover{color:#000!important;background-color:#9e9e9e!important}

.w3-light-grey,.w3-hover-light-grey:hover,.w3-light-gray,.w3-hover-light-gray:hover{color:#000!important;background-color:#f1f1f1!important}

.w3-dark-grey,.w3-hover-dark-grey:hover,.w3-dark-gray,.w3-hover-dark-gray:hover{color:#fff!important;background-color:#616161!important}

.w3-pale-red,.w3-hover-pale-red:hover{color:#000!important;background-color:#ffdddd!important}

.w3-pale-green,.w3-hover-pale-green:hover{color:#000!important;background-color:#ddffdd!important}

.w3-pale-yellow,.w3-hover-pale-yellow:hover{color:#000!important;background-color:#ffffcc!important}

.w3-pale-blue,.w3-hover-pale-blue:hover{color:#000!important;background-color:#ddffff!important}

.w3-text-amber,.w3-hover-text-amber:hover{color:#ffc107!important}

.w3-text-aqua,.w3-hover-text-aqua:hover{color:#00ffff!important}

.w3-text-blue,.w3-hover-text-blue:hover{color:#2196F3!important}

.w3-text-light-blue,.w3-hover-text-light-blue:hover{color:#87CEEB!important}

.w3-text-brown,.w3-hover-text-brown:hover{color:#795548!important}

.w3-text-cyan,.w3-hover-text-cyan:hover{color:#00bcd4!important}

.w3-text-blue-grey,.w3-hover-text-blue-grey:hover,.w3-text-blue-gray,.w3-hover-text-blue-gray:hover{color:#607d8b!important}

.w3-text-green,.w3-hover-text-green:hover{color:#4CAF50!important}

.w3-text-light-green,.w3-hover-text-light-green:hover{color:#8bc34a!important}

.w3-text-indigo,.w3-hover-text-indigo:hover{color:#3f51b5!important}

.w3-text-khaki,.w3-hover-text-khaki:hover{color:#b4aa50!important}

.w3-text-lime,.w3-hover-text-lime:hover{color:#cddc39!important}

.w3-text-orange,.w3-hover-text-orange:hover{color:#ff9800!important}

.w3-text-deep-orange,.w3-hover-text-deep-orange:hover{color:#ff5722!important}

.w3-text-pink,.w3-hover-text-pink:hover{color:#e91e63!important}

.w3-text-purple,.w3-hover-text-purple:hover{color:#9c27b0!important}

.w3-text-deep-purple,.w3-hover-text-deep-purple:hover{color:#673ab7!important}

.w3-text-red,.w3-hover-text-red:hover{color:#f44336!important}

.w3-text-sand,.w3-hover-text-sand:hover{color:#fdf5e6!important}

.w3-text-teal,.w3-hover-text-teal:hover{color:#009688!important}

.w3-text-yellow,.w3-hover-text-yellow:hover{color:#d2be0e!important}

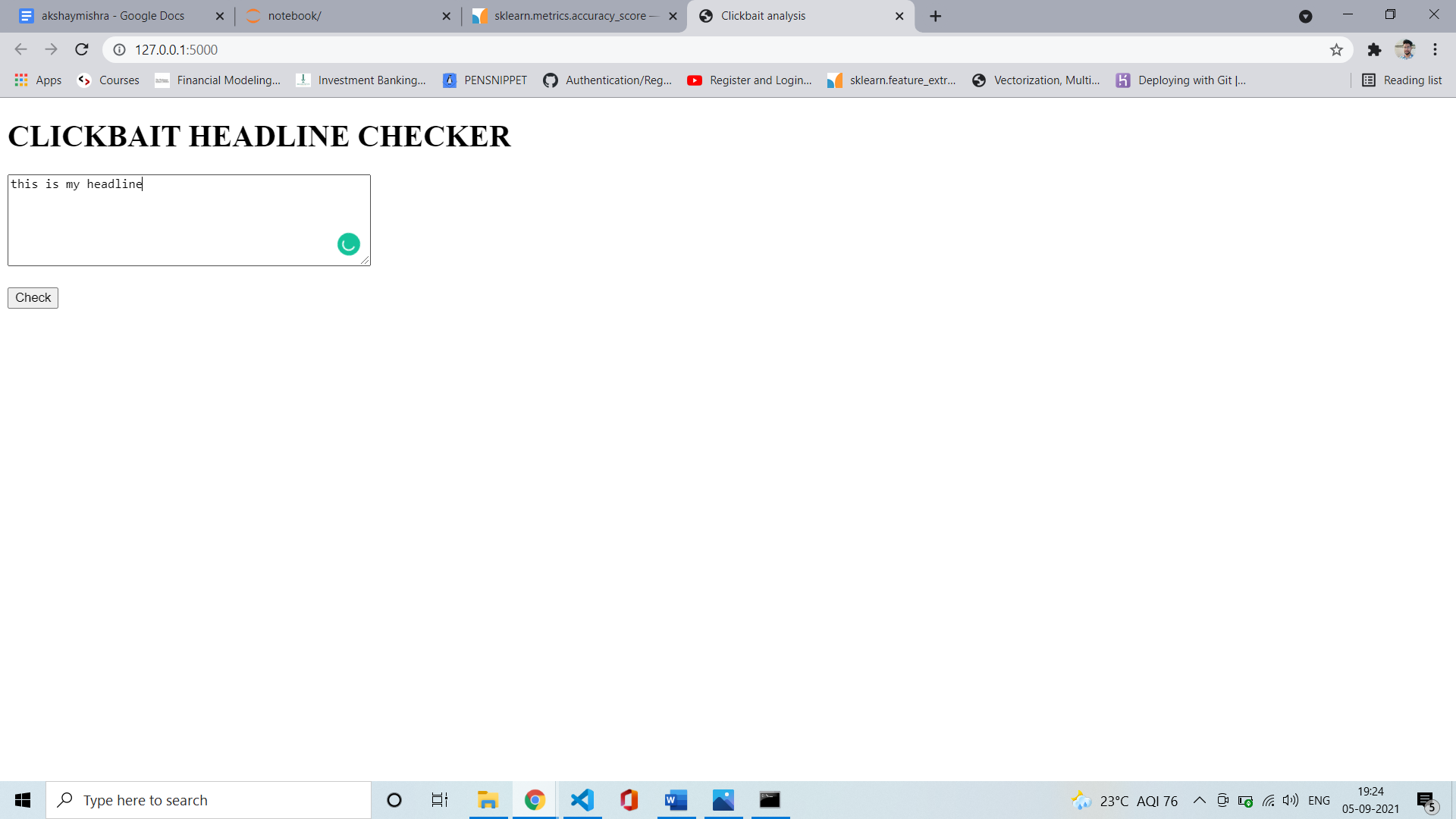
.w3-text-white,.w3-hover-text-white:hover{color:#fff!important}

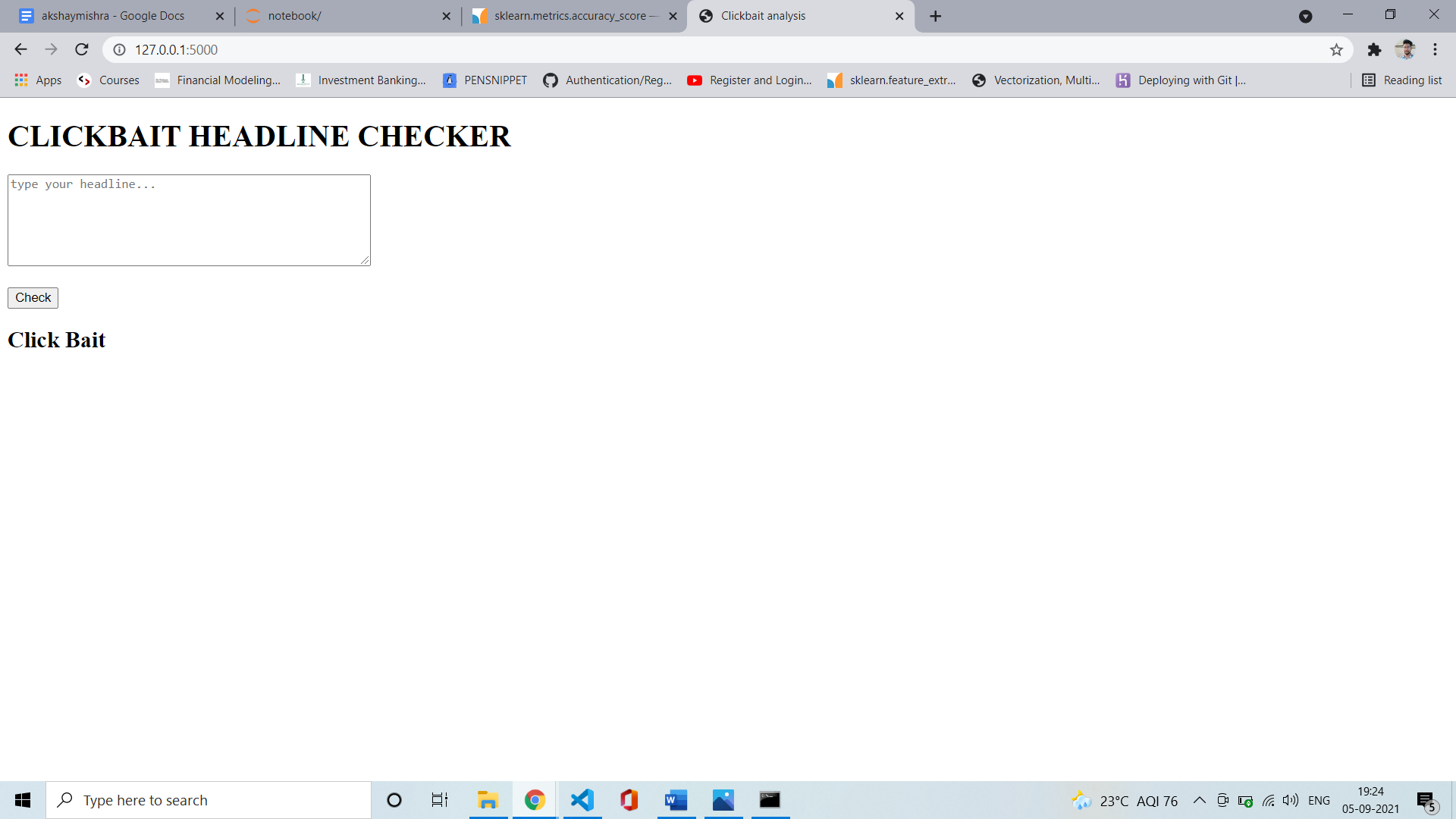
.w3-text-black,.w3-hover-text-black:hover{color:#000!important}

**3.5 MODEL DEPLOYMENT**

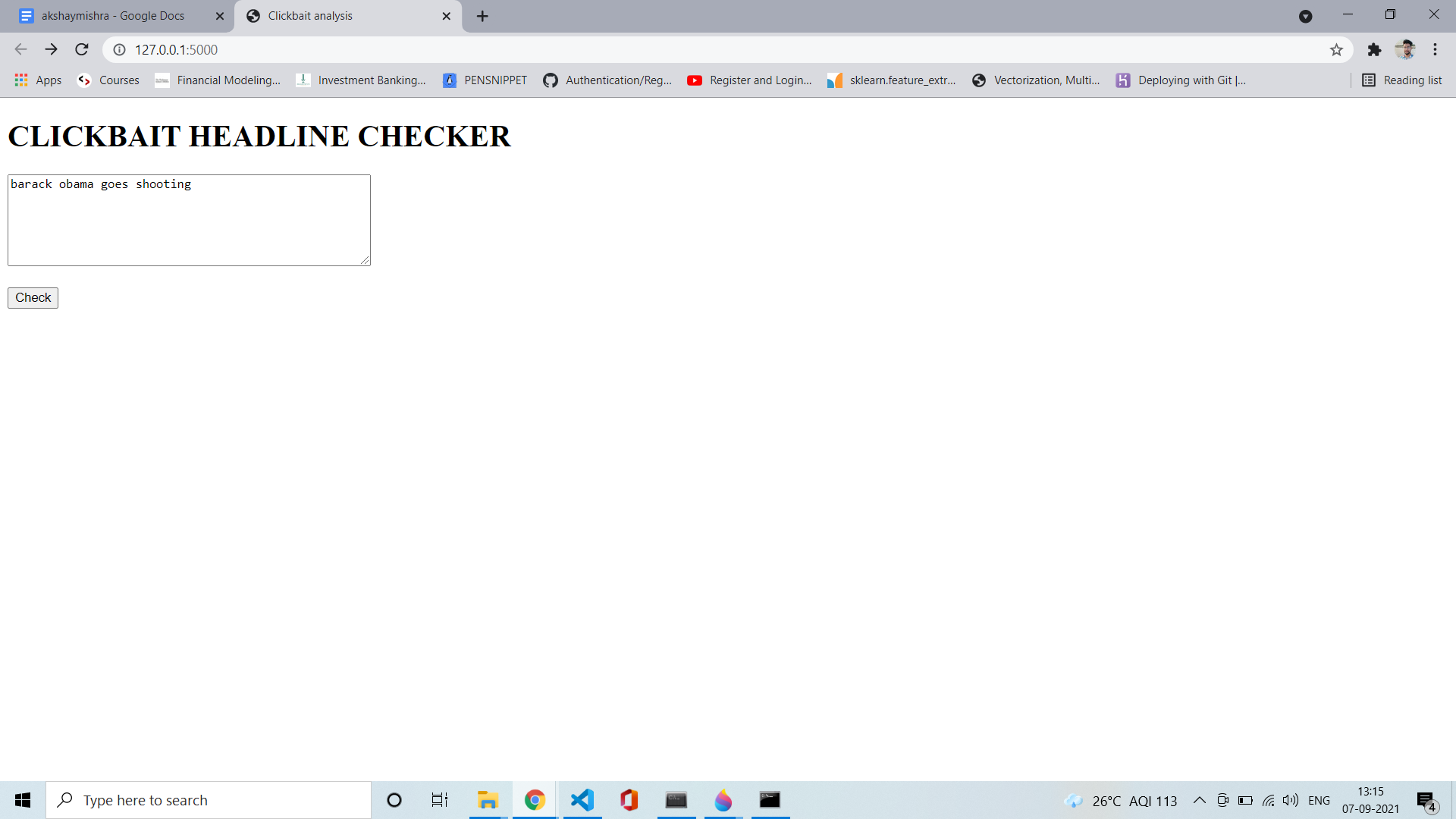
The model is deployed using Flask Framework

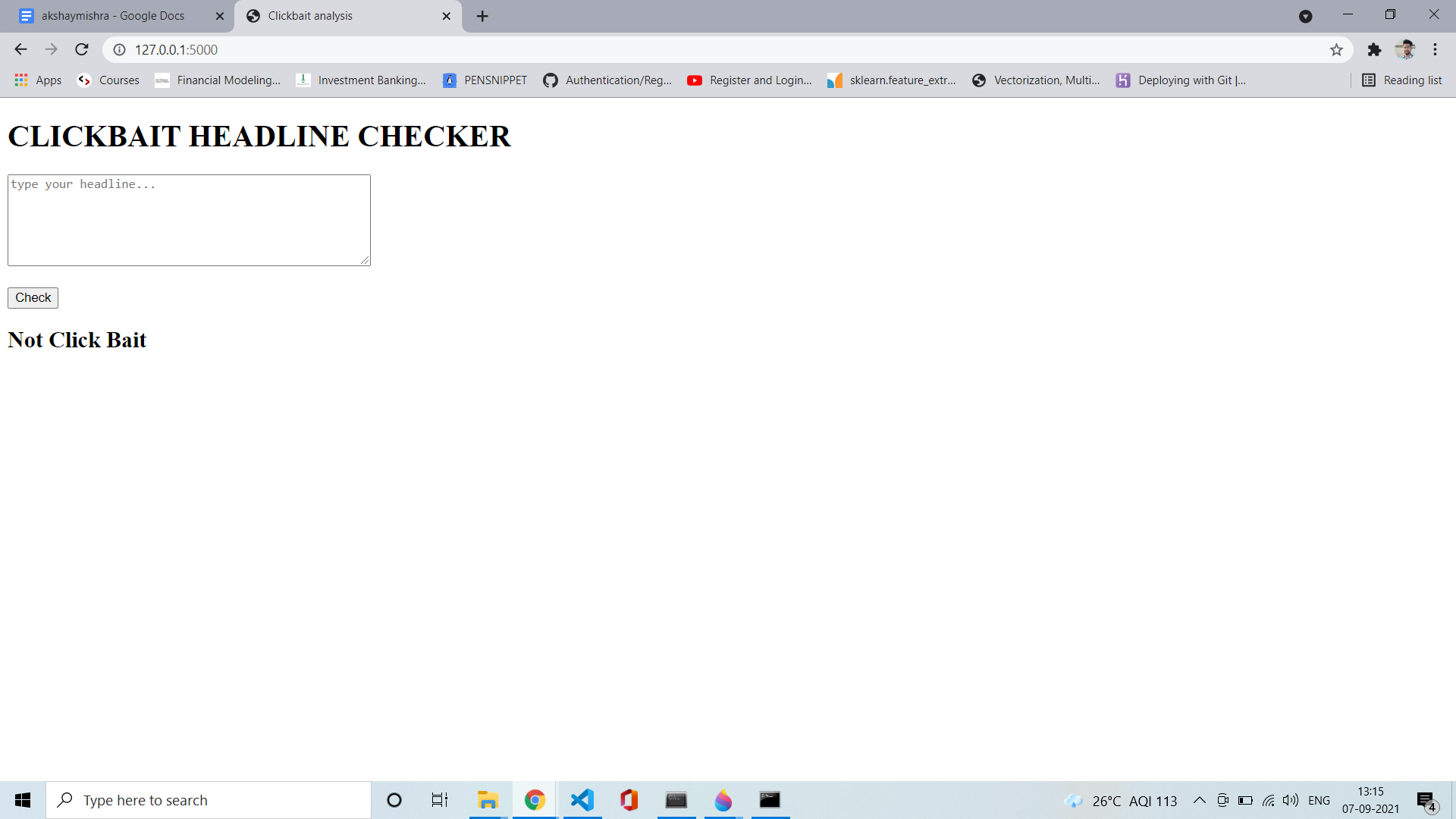
**CLICKBAIT**





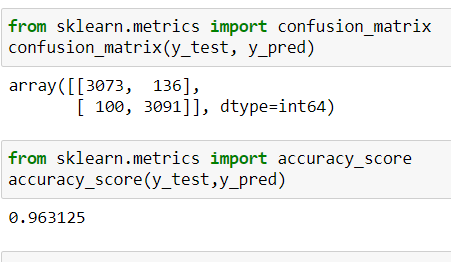
**NOT CLICKBAIT**





**CHAPTER 4**

**PERFORMANCE METRICS**



To evaluate our classifier, We use two methods: the accuracy score and a confusion matrix.

### Accuracy Score

Classification Accuracy is what we usually mean, when we use the term accuracy. It is the ratio of number of correct predictions to the total number of input samples.



It works well only if there are equal number of samples belonging to each class.

For example, consider that there are 98% samples of class A and 2% samples of class B in our training set. Then our model can easily get 98% training accuracy by simply predicting every training sample belonging to class A.

When the same model is tested on a test set with 60% samples of class A and 40% samples of class B, then the test accuracy would drop down to 60%. Classification Accuracy is great, but gives us the false sense of achieving high accuracy.

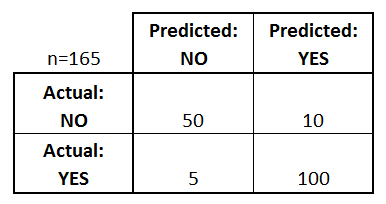
The real problem arises, when the cost of misclassification of the minor class samples are very high. If we deal with a rare but fatal disease, the cost of failing to diagnose the disease of a sick person is much higher than the cost of sending a healthy person to more tests.

### 

### Confusion Matrix

Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.

Let’s assume we have a binary classification problem. We have some samples belonging to two classes: YES or NO. Also, we have our own classifier which predicts a class for a given input sample. On testing our model on 165 samples,we get the following result.



There are 4 important terms :

* True Positives: The cases in which we predicted YES and the actual output was also YES.
* True Negatives: The cases in which we predicted NO and the actual output was NO.
* False Positives: The cases in which we predicted YES and the actual output was NO.
* False Negatives: The cases in which we predicted NO and the actual output was YES.

Accuracy for the matrix can be calculated by taking average of the values lying across the **“main diagonal”** i.e





**CONCLUSION AND FUTURE ENHANCEMENT**

Clickbait headlines have become a major issue in online news media. Hence the automated prevention is necessary.

We can conclude from the experimental results section that, only one categorization technique is not efficient enough to combat click bait articles. The results are quite acceptable by integrating more than one feature. Further, the websites are changing their strategies concerning time also.

Hence, the manufacturer will also find out the bypass of the categorization of clickbait articles. Hence a robust detection should be build that can predict future changes in the detection procedure by taking the time domain features. Also, the websites are using graphical images as clickbait headlines, which cannot be detected using text processing. So, by applying some image processing and pattern recognition schemes, we can detect the same after extracting the sentences. These can be the future scope of research in this domain.

\

**REFERENCE**

* Y. Chen, N. J. Conroy, and V. L. Rubin, “Misleading online content: Recognizing clickbait as false news,” in Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection, 2015, pp. 15–19.
* M. Potthast, S. Köpsel, B. Stein, and M. Hagen, “Clickbait detection,” in European Conference on Information Retrieval, 2016, pp. 810–817
* N. Hurst, “To clickbait or not to clickbait? an examination of clickbait headline effects on source credibility,” University of Missouri--Columbia, 2016.
* A. Agrawal, “Clickbait detection using deep learning,” in Next Generation Computing Technologies (NGCT), 2016 2nd International Conference on, 2016, pp. 268–272.