Clickbait Detection
POC Report for NLP Course, Winter 2025

Jakub Sawicki
Warsaw University of Technology
jakub.sawicki3.stud@pw.edu.pl

Jędrzej Sokołowski
Warsaw University of Technology
jedrzej.sokolowski.stud@pw.edu.pl

Wiktor Woźniak
Warsaw University of Technology
wiktor.wozniak2.stud@pw.edu.pl

supervisor: Anna Wróblewska
Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

Abstract

Clickbait headlines are designed to attract attention by omitting key information or sparking curiosity. This renders them difficult to identify via automated systems. This report examines the latest methods and resources for identifying clickbait, reviews current datasets, and provides a comparison between classical and transformer-based models. The method combines publicly available datasets with pretrained transformer-based models to deliver reliable and generalizable results.

1 Introduction

Clickbait refers to headlines or social media posts that are crafted to induce user engagement, often by teasing curiosity, using sensational language, or leaving out important information. Usually, the content fails to satisfy the curiosity gap induced in the headline. This is common across news sites and social media platforms. (Setlur, 2018; Omidvar et al., 2018). As a result, it is difficult to determine whether the headlines are legitimately engaging or if they are manipulative. As a result, it is more challenging for readers to assess the quality of information.

Detecting clickbait automatically is difficult. The difference between a catchy headline and a misleading one can be subtle. Even humans often disagree on what counts as clickbait. Effective detection systems need to look at both the language of the headline and how it relates to the actual article, because a headline may seem misleading without the full context (Omidvar et al., 2018). Importantly, clickbait exists on a spectrum. This makes the problem more complex and motivates research into models that can predict degrees of "clickbaitiness" rather than just labelling content as clickbait or not.

2 Background and Motivation

Efforts to automate clickbait detection have evolved from simple hand-crafted rules to sophisticated machine learning and deep learning models. The Clickbait Challenge 2017 advanced the field by introducing the idea of treating clickbait detection as a regression problem by rating headlines along a continuous scale (Potthast et al., 2018a). This has encouraged the development of more elaborate systems that go beyond simple binary categorization, capturing the subtlety and diversity of misleading headline techniques.

Beyond detection, clickbait "spoiling" has emerged as an additional task, where systems not only label clickbait but also generate or extract the missing information that these headlines withhold (Hagen et al., 2022). This approach recognizes that clickbaits often lack simple facts, and offers the withheld information directly to users reduces unnecessary clicks and increases transparency.

As clickbait tactics became more sophisticated and sometimes automated through AI, the need for robust and adaptable models grew. Advances in the field improved due to the growing availability of open datasets and shared task competitions, allowing for creation of reproducible benchmarks and faster scientific progress.

3 Related works

3.1 Feature Engineering and Hand-crafted Methods

Early clickbait detection systems relied heavily on feature engineering. These models examined a wide range of linguistic, syntactic, and content-based data taken from headlines and their corresponding articles (Elyashar et al., 2017; Zuhroh and Rakhmawati, 2019). Typical features included word and character counts, punctuation use, n-grams, informal language markers, and the de-

gree to which the headline matched the opening paragraphs of the article. Some more advanced approaches also incorporated text extracted from images using OCR, along with metadata such as posting time or how long a post was visible (Elyashar et al., 2017). Tools like Stanford CoreNLP provide additional syntactic features, including contraction frequency and dependency lengths (Chakraborty et al., 2016). With carefully selected features and preprocessing steps such as lowercasing and removing stopwords, these systems provided a robust foundation for the field. This allowed more recent research to expand upon these findings and further extend feature engineering for clickbait detection systems (Bronakowski et al., 2023).

3.2 Model Architectures: Classical, Neural, and Transformer-based

Classical machine learning models, such as Random Forests and SVMs, perform well when combined with well-designed features, reaching accuracy levels of up to around 92% in some studies (Al-Sarem et al., 2021). As deep learning became more common, the field shifted away from manual feature engineering. Models that represented headlines using Word2Vec, GloVe, and similar embeddings allowed CNNs and RNNs to automatically learn deeper patterns in the data (Muqadas et al., 2025; Chowanda et al., 2023). Architectures such as dedicated CNNs and MLPs outperformed traditional methods, especially on larger or multilingual datasets. They were better at capturing typical clickbait techniques.

More recently, transformer-based models such as DeBERTa and RoBERTa have set a new standard. These models are fine-tuned not only for clickbait detection (Alarfaj et al., 2025) but also for the newer task of clickbait spoiling (Fröbe et al., 2023). Because transformers are pre-trained on massive text data using objectives like masked language modeling and question answering, they achieve state-of-the-art results for both classification and spoiler generation. To address the computational overhead of these large-scale architectures, researchers have also explored DistilBERT, a light-weight version of BERT that utilizes knowledge distillation during the pre-training phase (Sanh et al., 2019). By reducing the number of parameters while retaining approximately 97% of the original model's performance, DistilBERT offers a more efficient alternative for real-time clickbait detection.

4 Open Benchmark Datasets

Four major open datasets have played a key role in clickbait detection research. Each of them offers a different perspective on how clickbait appears across platforms and languages.

Webis Clickbait Spoiling Corpus 2022 (Fröbe et al., 2023) is a comprehensive dataset designed for both clickbait detection and spoiling. It contains 5,000 manually annotated posts collected from Twitter, Reddit, Facebook, and other social platforms. All the headlines are considered clickbait. Additionally, a short piece of text was written that reveals the information that is missing in a headline (Hagen et al., 2022).

Wikinews Clickbait Corpus includes 18,513 news articles from Wikinews, with 7,623 labeled as clickbait by crowdsourced workers (Chakraborty et al., 2016). To create a balanced dataset, 7,500 non-clickbait articles were randomly selected. Each article was judged by at least three independent experts, and labels were assigned based on majority vote, improving consistency and reliability.

Thai Headline News Dataset expands clickbait research beyond English by providing 5,000 Thai news headlines, each annotated by two experts (Wongsap et al., 2018). For benchmarking, the dataset keeps 2,000 headlines labeled as clickbait and 2,000 as non-clickbait.

Chinese News Headlines Dataset contains 14,922 headlines from major Chinese news portals (Zheng et al., 2018). Half of the headlines are labeled as clickbait. The dataset also includes metadata about article types. Its size and diversity make it valuable for multilingual and cross-domain clickbait detection, particularly when evaluating advanced neural models.

These datasets enable robust model development, fair benchmarking, and strong generalization studies across both detection and spoiling tasks.

5 Proposed Solution and Project Plan

5.1 Objective

This project aims to develop a practical and interpretable clickbait detection system for headlines

using publicly available benchmark datasets. The main research questions guiding this work are:

- How effectively can classical machine learning models perform on clickbait detection using hand-crafted linguistic and content-based features?
- To what extent do embedding-based neural models improve performance over classical approaches on these datasets?
- Can transformer-based models, fine-tuned on clickbait datasets, provide superior accuracy and generalization while maintaining efficiency?

5.2 Feature Extraction

The proposed system will extract a comprehensive set of features from headlines and their associated articles, including but not limited to:

- N-gram frequencies (unigrams, bigrams) in headlines.
- Headline and article length (character and word counts).
- Overlap of headline terms with article opening sentences.
- Presence of informal words and pronouns.
- Sentiment score of the headline.
- Counts of punctuation and exclamatory words.

Standard NLP libraries such as NLTK or spaCy will be used to automate preprocessing and feature generation.

5.3 Models to be Evaluated

We plan to implement and compare the following models:

- **Classical Machine Learning Models:** Logistic Regression, Support Vector Machines (SVM), Random Forests, and Gradient Boosting Trees trained on the hand-crafted features.
- **Neural Embedding Models:** Feedforward Neural Networks and Convolutional Neural Networks (CNNs) leveraging pre-trained word embeddings (Word2Vec, GloVe).
- **Transformer Models:** Fine-tuned transformer-based architectures such as BERT and RoBERTa pre-trained on related NLP tasks and adapted for clickbait detection.

5.4 Datasets

We will use the Webis Clickbait Spoiling Corpus from different years as the primary data source. Data will be split into training and testing sets, ensuring balanced representation of clickbait and non-clickbait examples.

5.5 Evaluation

Performance will be measured using standard metrics:

- Accuracy and F1 score for classification.
- Mean squared error for measuring clickbait strength.

Additionally, feature importance analysis will be provided to identify which textual patterns contribute most to the detection task.

6 Experiments

6.1 Data

We used data from **Webis Clickbait Corpus 2017** (Potthast et al., 2018b). It consists of Twitter posts linking to news articles. The titles of articles are available with the degree of clickbaitiness in $[0, 1]$ interval. Larger values indicate that the observation is a clickbait.

As our datasets we used **clickbait17-train-170331** and **clickbait17-train-170630** folders available for download. To address the almost 3 to 1 class imbalance, we extended our data using **Webis Clickbait Spoiling Corpus 2022** (Fröbe et al., 2023) dataset only consisting of clickbait entries.

From each item, the post text was standardized by flattening the text field into a single string, producing a normalized headline representation. The labels were binarized using a threshold of 0.5. Entries with missing or empty textual content were removed. 54 rows were removed in total. Finally, we obtained 25943 observations. The resulting cleaned dataset was then stratified into training, validation and testing sets (70/15/15) preserving class balance in sets.

To provide a balanced training dataset, we applied undersampling to it. That was the reason

of increased size of the training data during split. We randomly selected a subset of majority (non-clickbait) class in training data to make the set balanced. Finally, 13270 observations were assigned to the training set, 3891 to the validation set and 3892 to the test one.

6.2 Data processing

We implemented a custom feature transformer to capture simple textual features. The extractor computes a fixed set of interpretable features for each headline. For every input text, the following attributes are derived:

- **Character count:** Total number of characters in the headline.
- **Word count:** Total number of whitespace-separated tokens.
- **Punctuation cues:** Counts of exclamation marks and question marks.
- **All-caps indicator:** Whether the entire headline is written in uppercase.
- **Sentiment polarity and subjectivity:** Obtained via `TextBlob`, capturing coarse affective tendencies.
- **Interrogative start:** A binary flag indicating whether the headline begins with typical question words (*who, what, where, why, how*), a pattern frequently associated with curiosity-inducing clickbait.

The obtained features are then added to the datasets.

6.3 Exploratory Data Analysis

By combining two datasets, class balance was improved, their distribution is shown in Figure 1.

To better understand the lexical characteristics of clickbait and non-clickbait headlines, we performed a frequency analysis of words in the training set. Headlines were first tokenized using `NLTK`'s word tokenizer, converted to lowercase, and filtered to remove stopwords and non-alphanumeric tokens.

For each class (*clickbait* and *non-clickbait*), the top 10 most frequent words were identified. The results are shown on Figures 2 and 3.

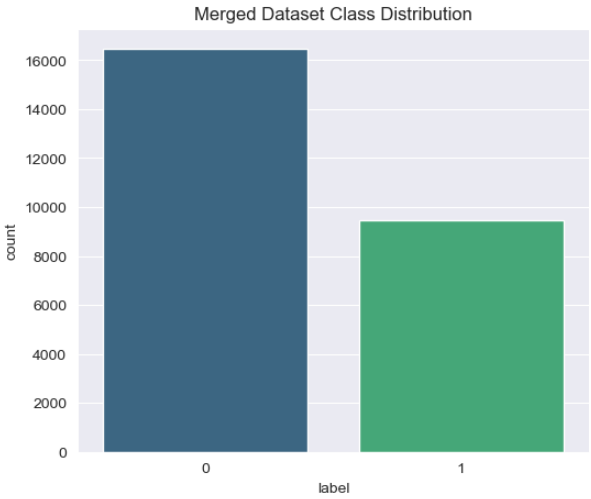

Figure 1: Class Distribution

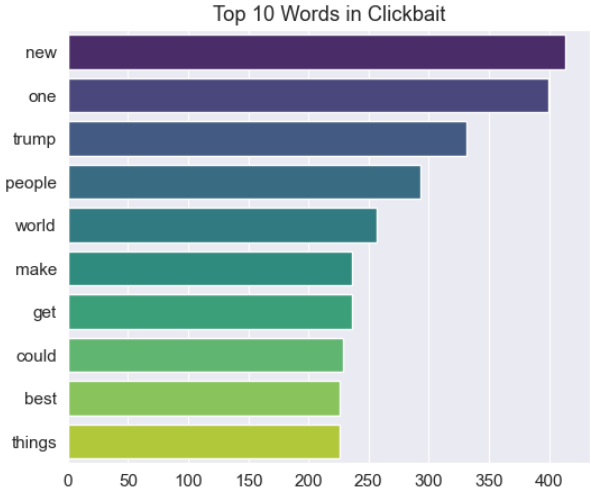

Figure 2: Top 10 words in clickbait headlines from the training set

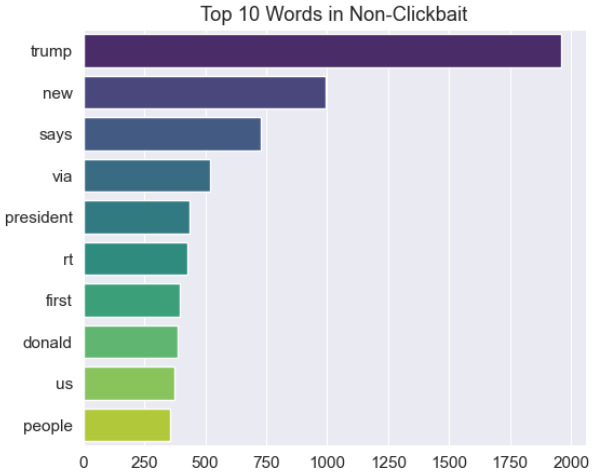

Figure 3: Top 10 words in non-clickbait headlines from the training set

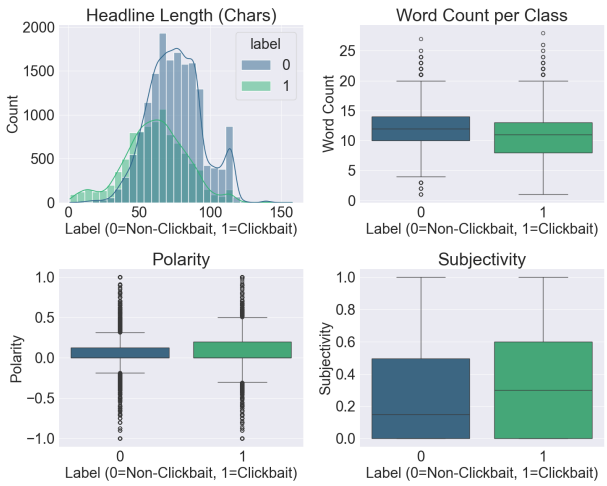

Figure 4: Distribution of selected features within clickbait and non-clickbait groups

We conducted further analysis to examine headline length, word count, and distributions of hand-crafted features such as punctuation counts, sentiment, and interrogative starts. Plots highlighting differences between clickbait and non-clickbait headlines are shown on Figures 4 and 5. In Figure 4 we can see subtle differences between the two labels, where clickbait headlines tend to be slightly shorter, more polarized and subjective. Figure 5 demonstrates that the proportion of headlines involving exclamation and question marks, as well as the usage of all caps case is vastly greater in clickbait ones. Similarly, they begin with question words much more often compared to non-clickbait.

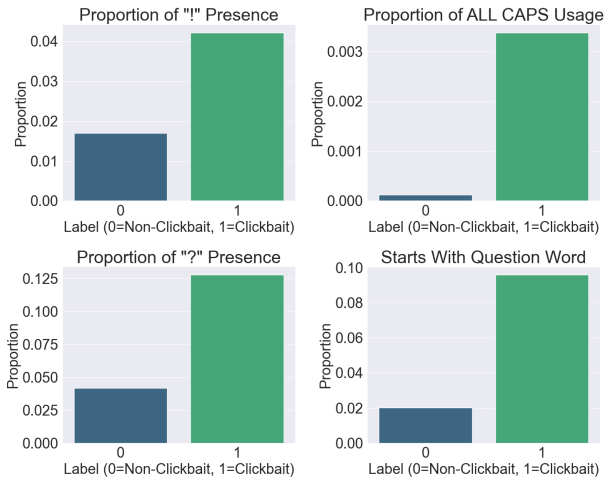

Figure 5: Distribution of selected features within clickbait and non-clickbait groups

6.4 Models

We decided to verify the results of classification for three approaches. We used Random Forest as a classical machine learning approach, a Feedforward Neural Network and a Transformer model - DistilBERT.
We used training set for training and validation set for hyperparameters tuning. We report the obtained results on the test set. Each model was trained 5 times with different random seeds. The show the average results.

6.5 Classical Machine Learning Model

We implemented a classical supervised model to classify headlines as clickbait or non-clickbait. The input features combined both textual and hand-crafted cues using a `FeatureUnion`:

- **TF-IDF n-grams:** Unigrams and bigrams extracted from headline text, limited to 1000 features.
- **Hand-crafted features:** Stylistic and lexical features such as punctuation counts, sentiment scores, all-caps usage, and interrogative start.

These features were utilized by a `RandomForestClassifier`. We performed cross-validation GridSearch for finding the best hyperparameters. We tested different number of trees, maximal depth and maximal number of features. We present the results on test set in Table 1.

Table 1: Random Forest Model Performance (Test Set)

Class	Prec.	Rec.	F1	Supp
0 (Non-Clickbait)	0.858 ± 0.003	0.821 ± 0.009	0.839 ± 0.005	2470
1 (Clickbait)	0.711 ± 0.010	0.765 ± 0.005	0.737 ± 0.006	1422
Accuracy		0.800 ± 0.006		3892
Macro Avg	0.785 ± 0.006	0.793 ± 0.005	0.788 ± 0.006	3892
Weighted Avg	0.805 ± 0.005	0.800 ± 0.006	0.802 ± 0.006	3892

We provide also a confusion matrix computed on the test set in Figure 6.

6.5.1 Feedforward Neural Network

Headlines were represented using pre-trained 50-dimensional GloVe embeddings, where each headline vector was computed as the average of its constituent word embeddings. The embeddings were concatenated with our custom features extracted from the headlines. These fixed-size representations were used as input to a feed-forward neural

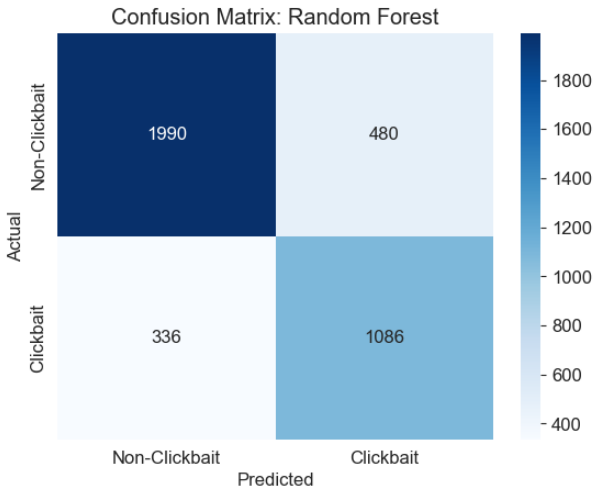

Figure 6: Confusion matrix of Random Forest.

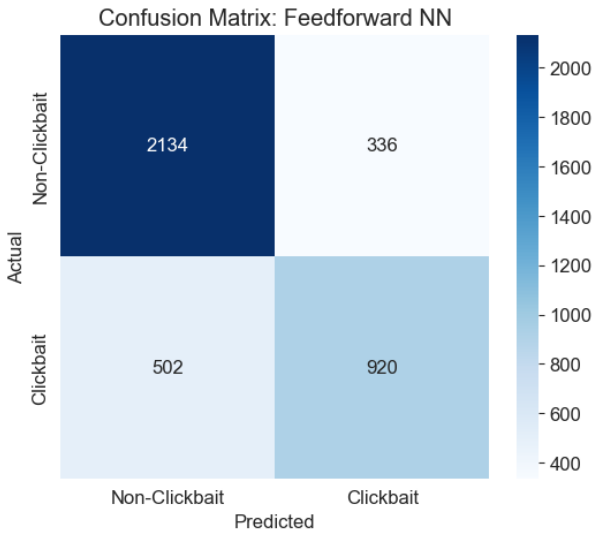

Figure 7: Confusion matrix of Feedforward NN model.

network consisting of three hidden layers (128, 64, and 32 units) with ReLU activations and dropout regularization. The model was trained for clickbait binary classification using PyTorch.

Adam optimizer with 0.003 learning rate and cross-entropy loss were used. The model was trained from scratch for 100 epochs with 32 batch size.

The validation loss was monitored every epoch and the best performing model on validation set was saved. The results of the best model can be found in Table 2 and Figure 7.

Table 2: Feed-Forward Neural Network Performance (Test Set)

Class	Prec.	Rec.	F1	Supp.
0 (Non-Clickbait)	0.827 ± 0.015	0.838 ± 0.023	0.832 ± 0.005	2470
1 (Clickbait)	0.713 ± 0.018	0.693 ± 0.039	0.702 ± 0.012	1422
Accuracy		0.785 ± 0.003		3892
Macro Avg	0.770 ± 0.003	0.766 ± 0.008	0.767 ± 0.005	3892
Weighted Avg	0.785 ± 0.004	0.785 ± 0.003	0.785 ± 0.003	3892

6.5.2 Transformer-based approach

To leverage contextual embeddings, we fine-tuned a DistilBERT model for binary clickbait classification. The text data was tokenized using the `DistilBertTokenizer` with truncation and padding to a maximum sequence length of 64 tokens. Fine-tuning was performed using the Hugging Face `Trainer` API with 2 epochs and a batch size of 8. Longer training increased the validation loss. The validation loss was monitored every 100 steps. The best model in terms of it was saved. The results are shown in Table 3 and Figure 8.

Table 3: DistilBERT Model Performance (Test Set)

Class	Prec.	Rec.	F1	Supp.
0 (Non-Clickbait)	0.881 ± 0.014	0.861 ± 0.031	0.871 ± 0.011	2470
1 (Clickbait)	0.770 ± 0.032	0.797 ± 0.032	0.782 ± 0.008	1422
Accuracy		0.838 ± 0.010		3892
Macro Avg	0.825 ± 0.011	0.829 ± 0.006	0.826 ± 0.008	3892
Weighted Avg	0.840 ± 0.006	0.838 ± 0.010	0.838 ± 0.009	3892

6.6 Matthews Correlation Coefficient analysis

The Matthews Correlation Coefficient (MCC) provides a balanced measure of binary classification performance under class imbalance. We computed the aggregated MCC values of models on the test set and we present them in Table 4. They show that DistilBERT achieves the highest and stable performance, outperforming the Random Forest and the Feed-Forward Neural Network. The lower MCC of RF and the Feedforward NN indicates they capture semantic cues less effectively than transformer-based models.

Table 4: Aggregated Matthews Correlation Coefficient (Test Set)

Model	Mean MCC	Std. MCC
Random Forest	0.5775	0.0108
Feedforward Neural Network	0.5356	0.0086
DistilBERT	0.6545	0.0149

6.7 Clickbaitness Analysis

We conducted a clickbaitness analysis using the measure proposed by (Urban et al., 2025). Bait-

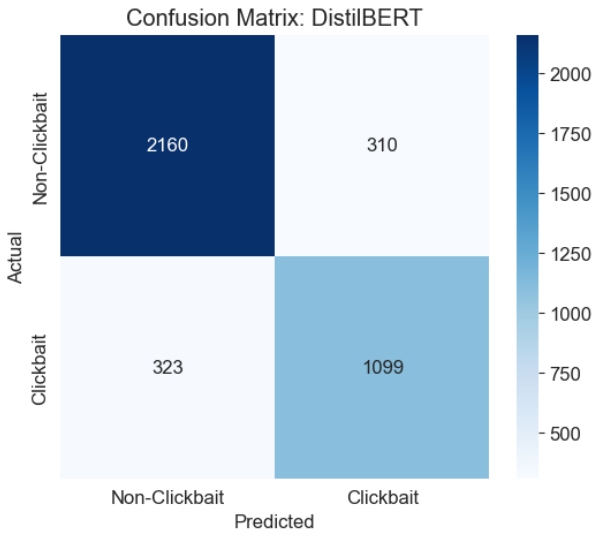

Figure 8: Confusion matrix of DistilBERT model.

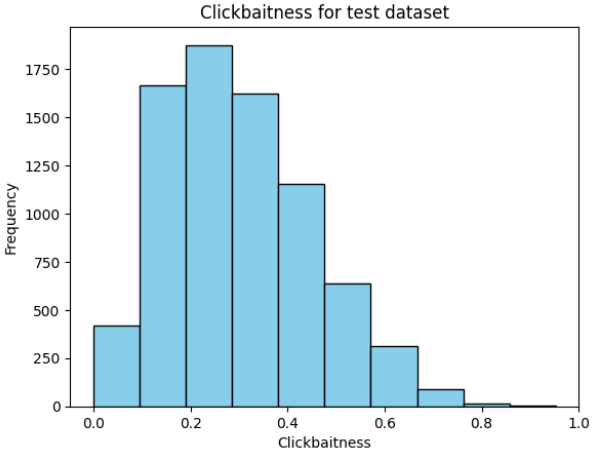

Figure 9: Histogram of clickbaitness scores in the test dataset.

ness is calculated using below equation:

$$\text{Baitness} = \frac{\text{EC} + \text{C} + \text{S} + \text{EOT}}{4} \qquad (1)$$

where EC is Eye catchness, C is Curiosity, S is Sentiment and EOT is Ease of text. More detailed describtion is present in (Urban et al., 2025).

Figure 9 presents the histogram of clickbaitness scores for the test dataset. As observed, most posts have clickbaitness values between 0.3 and 0.5. This distribution aligns with the label distribution shown in Figure 1, which indicates a higher number of non-clickbait posts compared to clickbait posts.

The analysis also highlights that this task is challenging due to the lack of extreme clickbaitness values, suggesting that most posts fall into a moderate range rather than being clearly clickbait or non-clickbait.

We also compare clickbaitness with the class probabilities returned by the models. Figure 10 presents the distribution of predicted clickbait probabilities for the test dataset. As shown, the models tend to be more confident in their predictions than would be expected based on the clickbaitness scores of the samples.

We also constructed a histogram of the absolute errors between the predicted clickbait probabilities and the clickbaitness scores for samples in the test dataset. This distribution is shown in Figure 11. Compared to the probability histograms, these results appear more informative. In particular, the neural network exhibits the smallest absolute er-

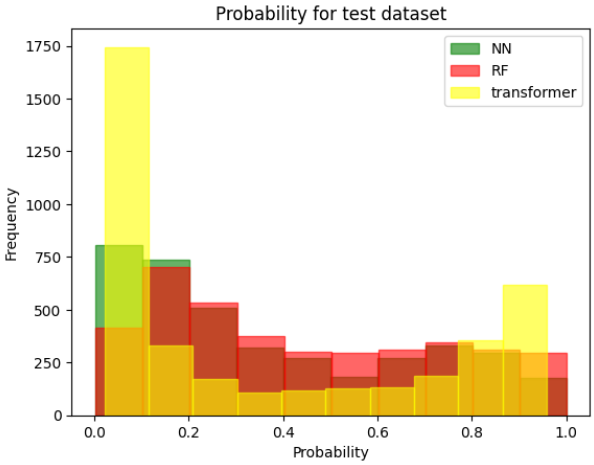

Figure 10: Histogram of predicted clickbait probabilities for the evaluated models.

rors, indicating that its predicted probabilities are most closely aligned with the clickbaitness values.

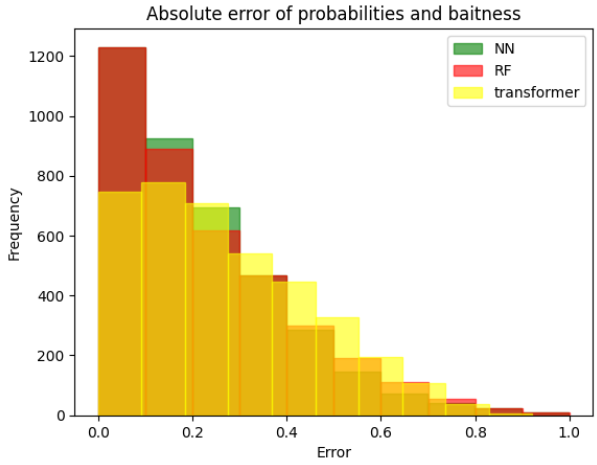

Figure 11: Histogram of absolute errors between predicted clickbait probabilities and clickbaitness scores.

Table 5 presents the error statistics for the evaluated models. While the mean errors appear reasonably low, the large standard deviations indicate substantial variability, suggesting that raw probability estimates should not be used directly for clickbaitness analysis.

Table 5: Error statistics between predicted clickbait probabilities and clickbaitness scores.

Model	MAE	MSE
Neural Network	0.217 ± 0.174	0.077 ± 0.118
Random Forest	0.226 ± 0.188	0.086 ± 0.130
Transformer	0.269 ± 0.184	0.106 ± 0.127

7 Summary

This project aimed to build a clear and effective system to detect clickbait headlines. We compared different approaches, from traditional machine learning using hand-picked features to advanced neural networks and transformer models. By using benchmark datasets, we measured the efficacy of each method.

The results suggest that the most prominent approach from the tested ones is Transformer-based. It demonstrated superior precision–recall balance across both classes.

The NN model achieved comparative results to the RF. RF performance was even more stable.

Nevertheless, the discrepancy in models' performance was not huge. All three models achieved promising results and were able to effectively handle the minority class.

We also performed a clickbaitness analysis by comparing it with the probabilities returned by the models. However, the results of this approach are unstable due to the high variance of the errors. This suggests that clickbaitness estimation may be better treated as a separate task rather than as a subtask of clickbait detection.

It is noteworthy that limitation of the presented work exist. All models were trained and evaluated only on english social media data, and may not generalize to other clickbait sources and languages.

References

Mohammed Al-Sarem, Faisal Saeed, Zeyad Al-Mekhlafi, Badiea Al-Shaibani, Mohammed Hadwan, Tawfik Al-Hadhrami, Mohammad Alshammari, Abdulrahman Alreshidi, and Talal Alshammari. 2021. An improved multiple features and machine learning-based approach for detecting clickbait news on social networks. *Applied Sciences*, 11:9487, 10.

Fawaz Alarfaj, Amara Muqadas, Hikmat Khan, and Anam Naz. 2025. Clickbait detection in news headlines using roberta-large language model and deep embeddings. *Scientific Reports*, 16, 12.

Mark Bronakowski, Mahmood Al-Khassaweneh, and Ali Al Bataineh. 2023. Automatic detection of clickbait headlines using semantic analysis and machine learning techniques. *Applied Sciences*, 13(4):2456.

Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop clickbait: Detecting and preventing clickbaits in online news media. 10.

Andry Chowanda, Nadia Nadia, and Lie Kolbe. 2023. Identifying clickbait in online news using deep learning. *Bulletin of Electrical Engineering and Informatics*, 12:1755–1761, 06.

Aviad Elyashar, Jorge Bendahan, and Rami Puzis. 2017. Detecting clickbait in online social media: You won't believe how we did it.

Maik Fröbe, Tim Gollub, Benno Stein, Matthias Hagen, and Martin Potthast. 2023. SemEval-2023 Task 5: Clickbait Spoiling. In *17th International Workshop on Semantic Evaluation (SemEval 2023)*, pages 2278–2289, Toronto, Canada, July. Association for Computational Linguistics.

Matthias Hagen, Maik Fröbe, Artur Jurk, and Martin Potthast. 2022. Clickbait spoiling via question answering and passage retrieval.

Amara Muqadas, Hikmat Khan, Muhammad Ramzan, Anam Naz, and Ali Daud. 2025. Deep learning and sentence embeddings for detection of clickbait news from online content. *Scientific Reports*, 15, 04.

Amin Omidvar, Hui Jiang, and Aijun An. 2018. Using neural network for identifying clickbaits in online news media.

Martin Potthast, Tim Gollub, Matthias Hagen, and Benno Stein. 2018a. The clickbait challenge 2017: Towards a regression model for clickbait strength.

Martin Potthast, Tim Gollub, Matti Wiegmann, Benno Stein, Matthias Hagen, Kelsey Komlossy, Sebastian Schuster, and Eduardo Pedroza Garcia Fernandez. 2018b. Webis clickbait corpus 2017 (webis-clickbait-17). Data set.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Amrith Rajagopal Setlur. 2018. Semi-supervised confidence network aided gated attention based recurrent neural network for clickbait detection.

Tymoteusz Urban, Mateusz Kubita, and Wojciech Michaluk. 2025. Clickbait news detection and analysis.

Natnicha Wongsap, Tastanya Prapphan, Lisha Lou, Sarawoot Kongyoung, Sasiwimol Jumun, and Natsuda Kaothanthong. 2018. Thai clickbait headline news classification and its characteristic. In *2018 International Conference on Embedded Systems and Intelligent Technology International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES)*, pages 1–6.

Hai-Tao Zheng, Jin-Yuan Chen, Xin Yao, Arun Kumar, Yong Jiang, and Cong-Zhi Zhao. 2018. Clickbait convolutional neural network. *Symmetry*, 10:138, 05.

Nurrida Zuhroh and Nur Rakhmawati. 2019. Clickbait detection: A literature review of the methods used. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, 6:1, 10.

A Reproducibility checklist

- **Model Description:** `distilbert/distilbert-base-uncased`: DistilBERT is a smaller and faster model than BERT, which was pretrained on the same corpus in a self-supervised fashion, using the BERT base model as a teacher. It is available in HuggingFace transformers python package.
- **Link to Code:** `https://github.com/sokolowskij/nlp2025`
- **Infrastructure:**
  - CPU: Intel(R) Core(TM) i7-14700F @ 2.10 GHz
  - RAM: 64 GB
  - GPU: NVIDIA GeForce RTX 4070 Ti SUPER
  - OS: Microsoft Windows 11 Home (64-bit)
- **Runtime Parameters:**
  - Random Forest:
    - Training time (5 runs): 8 min 44 s
    - Data size (sparse features): 2.66 MB
  - Feed-Forward Neural Network (NN):
    - Training time (5 runs): 5 min 38 s
    - Batch size: 32
    - Learning rate: 0.003 (Adam optimizer)
    - Epochs: 100
    - Validation monitoring: loss monitored every epoch
    - Early stopping: best model saved based on validation loss
    - Data size (with handcrafted features): 9.48 MB
  - DistilBERT:
    - Training time (5 runs): 15 min 16 s
    - Batch size: 8
    - Epochs: 2
    - Tokenization: `DistilBertTokenizer` with truncation and padding to 64 tokens
    - Validation monitoring: loss monitored every 100 steps
    - Early stopping: best model saved based on validation loss
    - Data size: 18.70 MB
- **Parameters:**
  - Random Forest:
    - Number of trees, maximal depth, and maximal features tuned via GridSearch
    - Model size: 147.28 MB
  - Feed-Forward Neural Network (NN):
    - Input features: 50-dimensional GloVe embeddings + handcrafted features
    - Architecture: 3 hidden layers (128, 64, 32 units) with ReLU and dropout (0.3)
    - Total trainable parameters: 17,954 (0.068 MB)
  - DistilBERT:
    - Transformer layers: 6
    - Hidden size: 768
    - Attention heads: 12
    - Total trainable parameters: 66,955,010 (255.41 MB)

- **Test Performance:**
  - Random Forest: see Table 1
  - Feed-Forward Neural Network (NN): see Table 2
  - DistilBERT: see Table 3
- **Data Sizes:**
  - Merged dataset: 3.79 MB
  - Merged dataset with handcrafted features: 5.38 MB
- **Data Download:**
  - webis-clickbait-22.zip from `https://zenodo.org/records/6362726`
  - clickbait17-train- 170331.zip and clickbait17-train-170630.zip from `https://zenodo.org/records/5530410`
- **Data Languages:** English

B Reviews answers

B.1 Answer to Mateusz Andryszak, Michał Chęć, Aleks Kapich, Zuzanna Piróg

We fully acknowledge your comments. While the suggestion to include DeBERTa in our comparison is valid, by basing our comparison on a classical machine learning model and a simple feedforward neural network, we focused at speed and efficiency of the chosen Transformer model and that is the reason why we decided to choose DistilBERT.

In alignment with your suggestions, we added the missing explanation of one of the used metrics (Urban et al., 2025) and a proper reference to DistilBERT.

To consider the lack of statistical significance testing, we performed multiple runs of the models and described the results.

Furthermore, we included a description of highlighted limitations in the summary.

B.2 Answer to Magdalena Jeczeń, Piotr Rowicki, Krzysztof Wolny

We took your suggestions into consideration. To account for possible bias in our dataset, according to your advice, we implemented additional undersampling. Furthermore, we included the information about all caps and punctuations in feedforward model.

Regarding using accuracy score in a scenario with class imbalance, we address it by utilizing it alongside precision, recall, F1-score and averages to provide a wider overview of the performance. To account for this insight, we further added the usage of Matthews Correlation Coefficient (MCC).

The dataset choice, while indeed focused on social media, was based on the fact that clickbaits are most common on social media websites, moreover, this domain and the datasets used are one of the most well-documented and researched in this field, providing a fair and validated setting for model comparison.

B.3 Answer to the teachers

We have incorporated the suggested improvements. A contribution table has been added at the end of the report. The text has been revised for clarity and to adopt a more formal style. We also reviewed recent literature and updated the discussion to include the latest advances in transformer models. Additionally, MCC has been included as an evaluation metric for our models. A new section on reproducibility has been added, detailing the computer specifications and computational costs. Finally, comments have been added to our code notebooks.

C Division of Work

Task	Team Member	Time Estimate
EDA	Jędrzej Sokołowski	3h
Data preprocessing	Jędrzej Sokołowski	2h
Related Works refinement	Jędrzej Sokołowski	2h
Initial POC	Jędrzej Sokołowski	1h
Writing report	Jędrzej Sokołowski	2h
Feedback analysis	Jędrzej Sokołowski	1h
Initial literature review	Jakub Sawicki	2h
Training models (POC)	Jakub Sawicki	2h
Final models training	Jakub Sawicki	4h
Writing report	Jakub Sawicki	2h
Literature review	Wiktor Woźniak	1.5h
Baitness analysis (POC)	Wiktor Woźniak	2h
Baitness analysis (Final)	Wiktor Woźniak	2h
Writting report	Wiktor Woźniak	2h
Feedback analysis	Wiktor Woźniak	1h