Research Article

International Journal of Data Science and Applications (JOINDATA) 7(1), 64-80, 2024

Received: 13-Oct-2024 Accepted: 21-Nov-2024

homepage: https://dergipark.org.tr/tr/pub/joindata

Turkish Clickbait News Detection using Explainable Artificial Intelligence

Alper Celal Akgün[1,*], Tülin İnkaya[2]

[1, *] Corresponding Author, Uludag University, Bursa, Turkiye, e-mail: alpercelal.akgun@hotmail.com
[2,] Uludag University, Bursa, Turkiye, e-mail: tinkaya@uludag.edu.tr

ABSTRACT

Internet users frequently prefer digital journalism to acquire information. However, the content produced by malicious news sources leads to various issues for users. One of these issues is clickbait headlines, which are used to capture users' attention and direct them to specific content. Clickbait headlines exploit users' curiosity, causing them to navigate to targeted content and spend more time on it. Such content, which can be malicious, is one of the main problems for today's internet users. In the literature, artificial intelligence-based approaches using machine learning and deep learning models have been developed for the problem of clickbait detection. However, there is a need for studies on the explainability of artificial intelligence models developed in this field. Explainable artificial intelligence (XAI) aims to explain the transparency, understandability and decision-making processes of machine learning models. This study aims to develop explainable artificial intelligence-based models for the clickbait detection problem. In this context, a Turkish dataset compiled from different news sources was used. Initially, data preprocessing activities including feature engineering, missing data handling, stemming, normalization and term frequency-inverse document-frequency (TF-IDF) transformation were performed. Subsequently, k-nearest neighbors (KNN), naive Bayes, logistic regression, decision tree, random forest, extreme gradient boosting (XGBoost), support vector machine (SVM) and multi-layer perceptron (MLP) models were developed using the dataset. Hyperparameter optimization was applied to determine the most suitable parameter values for each model. The performances of the applied models were comparatively evaluated. The models with the highest performance were XGBoost, SVM, and logistic regression, each achieving approximately 89%. Finally, to ensure the explainability of artificial intelligence models in clickbait detection, the Shapley additive explanations (SHAP) method was used for identifying the factors affecting the classification results.

Keywords: Clickbait Detection, Natural Language Processing, SHAP, Explainable Artificial Intelligence

1 Introduction

In the digital age, the internet has become the main source for users to obtain information regarding current events and developments. The fast and extensive access to information provided by the internet allows users to follow the latest news and updates instantly. Digital sources play a crucial role in sharing news and information, while also enabling knowledge acquisition on various topics. According to the annual report by We Are Social, 67.6% of the users in Türkiye use the internet to stay informed about news and events in 2023 [1]. However, with the widespread adoption of digital journalism, significant challenges regarding content quality and reliability have emerged.

Misleading content produced by malicious news sources causes various problems for the users. While aiming to reach accurate and reliable information, users frequently encounter such misleading content, making it difficult to access correct information and undermining their trust in reliable news sources.

One prominent issue for the users is clickbait headlines used to attract attention and direct users to specific content. Clickbait headlines exploit users' curiosity, leading them to targeted content and encouraging them to spend more time on these pages. These headlines often contain exaggerated, sensational, or misleading statements. Şahin et al. described clickbait content as headlines that create a click reflex by using attention-grabbing or sensational statements to attract user attention [2]. The goal is to increase site traffic. However, such headlines often lead to disappointment regarding the quality and truth of the content. This type of potentially malicious content is a major problem for today's internet users. It decreases trust in news sources and negatively impacts overall information acquisition processes.

In today's world, where artificial intelligence (AI) is rapidly gaining popularity, numerous studies are being conducted daily. However, the inner workings and decision-making mechanisms of AI methods remain a black box. In the literature, machine learning and deep learning models have been created to address the issue of clickbait detection problem. Raj et al. [3] reviewed 25 studies conducted up to 2023. Additionally, Adrian et al. proposed an approach for detecting clickbait using machine learning and deep learning [4], while Arfat et al. performed clickbait detection using a community-based method [5]. However, there is a great need for studies focusing on the explainability of AI models developed in this field. The need to understand the dynamics of AI decision-making processes has sparked to the development of explainable artificial intelligence (XAI). XAI encompasses the approaches and techniques used to clarify machine learning and deep learning models, focusing on describing their accuracy, transparency and outcomes [6]. XAI ensures that models do not remain a black box, allowing users and researchers to understand how and why these models make certain decisions. Methods like local interpretable model-agnostic explanations (LIME) [7] and Shapley additive explanations (SHAP) [8] are popular in this field.

The conducted study seeks to develop an explainable AI-based model for the clickbait detection problem. Efforts are made to understand how the models operate and on what criteria affect their decision making. This study contributes to the literature with the use of explainable AI in Turkish clickbait detection for the first time.

The other sections of the article are summarized as below: The second section provides an overview of previous studies on clickbait detection and explainable artificial intelligence. The third section includes the methodology used. The fourth section covers experimental conditions and comparative results. The last section delivers conclusions and potential upcoming studies.

2 Literature Review

In this section, studies on clickbait detection are summarized first. Subsequently, studies on clickbait detection using explainable artificial intelligence are investigated. Due to the limited number of studies in this specific area, other studies in the field of XAI are also summarized.

In 2016, Potthast et al. conducted the first clickbait study using data obtained from Twitter (x.com). This study highlighted the impact of features derived through feature engineering [9]. In the same year, Chakraborty et al. published a study, which determines the effective features in clickbait detection using machine learning algorithms [10]. Yadav and Bansal employed naive Bayes (NB) to probabilistically classify clickbait and non-clickbait headlines based on different word usage frequencies. Additionally, they used support vector machine (SVM) and random forest (RF) to leverage headline features, aiming for higher class separation [11].

Recent studies in the field of clickbait also incorporate machine learning techniques along with deep learning methods. Chowanda et al. explored various machine learning and deep learning methods for clickbait detection, including SVM, naive Bayes, logistic regression (LR), random forest, generalized linear model, fast large margin, artificial neural network (ANN), gradient boosted trees, decision tree (DT) and bidirectional encoder representations from transformers (BERT). Each model (except BERT) were trained with term frequency-inverse document frequency (TF-IDF). The best model was BERT, which achieved an accuracy of 98.86%. [12]. Coste and Bufnea proposed a language-independent approach for clickbait detection. They used deep learning models to identify patterns in headline and content features. They also used word embedding techniques to create language-agnostic vector representations for semantic inconsistencies between headlines and content [13]. Adrian et al. utilized naive Bayes to analyze word frequency in headlines to detect clickbait. They also employed long short term memory (LSTM) to evaluate word sequences and contextual relationships in headlines [4]. Mahtab et al. introduced a semi-supervised approach for clickbait detection in Bengali. The method relied on an adversarial network model that leverages unlabeled data to support the limited labeled data and utilizes language models for feature extraction [14]. Broscoțeanu and Ionescu proposed an innovative method where headlines and content are jointly encoded in a deep metric space. This method modeled high cosine similarity for non-clickbait and low similarity for clickbait articles [15]. Liu et al. introduced a deep learning model that integrates semantic and syntactic information to better understand and detect mismatches between headline and content [16]. Genç et al. achieved high accuracy in Turkish clickbait detection using various machine learning and deep learning methods with their ClickbaitTR dataset [17].

When examining studies in the area of XAI, Shu et al. used attention mechanisms and multimodal deep learning methods for fake news detection. These methods aim to make fake news detection more explainable by combining textual and visual features [18]. Chien et al. applied deep learning-based classification methods and XAI techniques for fake news detection. Their aim was to enhance model transparency through feature importance ranking and visualization techniques [19]. Sharma and Midhunchakkaravarthy used extreme gradient boosting (XGBoost) for dementia detection in young adults, applying LIME and SHAP to enhance model explainability. LIME provides explanations based on specific instances, while SHAP evaluates the contribution of each feature to the final decision, making the model’s decision processes more interpretable [20]. Pérez-Landa et al. used XAI methods for detection tasks, specifically applying SHAP to explain model decisions [21]. Zhou et al. utilized knowledge graphs and graph convolutional networks (GCN) as XAI techniques. These techniques were used to capture detailed relationships in clickbait headlines, aiming for a more nuanced and detailed detection of clickbait [22]. Turan et al. employed LIME and SHAP to predict Turkish Constitutional Court decisions [23]. Rao et al. applied LIME and SHAP for autonomous disease prediction [24].

The above studies show that research in the field of XAI has been increasing in various domains. However, there is a gap in the literature regarding clickbait detection. This study aims to address this gap using SHAP method with machine learning algorithms.

3 Methodology

This section explains the dataset, machine learning models, XAI model and proposed approach used in the study.

3.1 Dataset

In the present study, we focus on an XAI model for detecting Turkish clickbait. The dataset titled Turkish News Title 20,000+ Clickbait Classified [25], which was compiled and classified from 21 different Turkish news sources, were used Labeling process was conducted based on the general clickbait tendency published by news sites. The dataset consists of 20,038 rows and four features: title number (id), clickbait status (clickbait), source (site) and title. Among the titles, 10,030 are classified as clickbait, while 10,008 are classified as non-clickbait. The distribution of titles according to their clickbait status is given in Figure 1. As illustrated in Figure 1, the dataset has a balanced class distribution.

3.2 Data Preprocessing

This subsection explains the steps performed during the data preprocessing phase in sequential order.

A key element in the success of machine learning projects is the set of features used. Representational and independent features facilitate learning, while complex relationships make learning more difficult. Feature engineering includes deriving new features from raw data to make the data more meaningful. This process includes steps such as data collection, cleaning, preprocessing and trial-and-error feature design [26].

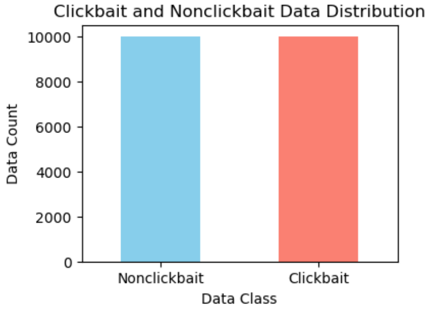

Figure 1: Data class distribution

Zemberek is an open-source natural language processing (NLP) tool developed for Turkic languages. Designed to overcome the challenges arising from the agglutinative structure of Turkish and other Turkic languages, this library performs essential NLP functions such as spell checking, morphological analysis, stemming, word suggestion, ASCII character conversion and syllabification. Zemberek provides a valuable resource for researchers working on NLP studies for Turkic languages [27].

TF-IDF is a statistical technique used to assess the significance of a term both within a single document and throughout a collection of documents. Term frequency (TF) determines how frequently a term appears in a document, expressed as the ratio of the term's occurrences to the word count in that document. Inverse document frequency (IDF) gauges the term's importance across the entire document set, calculated by taking the logarithm of the ratio of all documents to the number of documents that contain the term. The product of these two values gives the TF-IDF score, which is used to determine the importance of a term. This method increases the importance of more specific and distinctive terms instead of frequently occurring common terms [28].

Min-max normalization is a commonly applied data transformation method aimed at retaining the information in a dataset. This technique involves applying a linear transformation to the original data, making it particularly useful for classification tasks. The aim is to transform the data into the desired form with minimal information loss [29]. In this study, the values in the dataset are normalized using the minimum and maximum values.

3.3 Machine Learning Models

Studies on clickbait detection have predominantly employed machine learning and deep learning methods [30]. In this study's explainable artificial intelligence model for clickbait detection, machine learning methods were also utilized. The dataset was trained using k-nearest neighbors (KNN), naive Bayes, logistic regression, decision tree, random forest, XGBoost, SVM MLP. Information on these methods is provided below.

KNN is based on the idea that data points in a dataset are likely to be near other points with similar characteristics. When examples have assigned class labels, the class of an unlabeled data point can be inferred by examining the class labels of its closest neighbors. KNN identifies the k-nearest data points to the one in question and predicts its class by selecting common class label among those neighbors [31].

Naive Bayes is a basic Bayesian network that consists of a single parent node with several child nodes (representing observed variables) connected to it. It is based on the supposition that the child nodes are unrelated to one another given the parent node. As a classification algorithm, naive Bayes is used to estimate the class labels of observed data. This algorithm works under the assumption that features are independent and uses Bayes' theorem to calculate the probabilities for each class. The key advantage of the naive Bayes algorithm is the speed of its training process [31].

A decision tree classifies examples on the basis of the values of their attributes. Nodes in the decision tree reflect attributes in an example, and branches represent the values those attributes can take. Examples are sorted and classified starting from the root node based on their attribute values. A decision tree finds the attribute that best splits the training data, and makes this attribute the root node. This process is repeated by creating subtrees for each part of the split data. A decision tree clearly shows which tests were performed to classify an example into a specific class, making it easy to understand. Decision trees assume that examples belonging to different classes have different values for at least one attribute. Also, they often perform better with categorical attributes [31]. Figure 2 demonstrates an example decision tree.

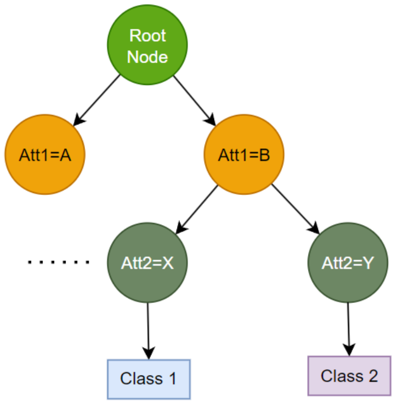

Figure 2: Example decision tree.

Logistic regression is a probability-based statistical model widely used in machine learning to solve classification problems. Logistic regression typically employs the logistic function, mathematically defined as the sigmoid function, to predict probabilities. It performs effectively if the dataset could be linearly separated since it assumes linearity between dependent and independent variables. It is applicable to both classification and regression tasks [32].

Random forest is an ensemble classification technique used in miscellaneous application fields in machine learning and data science. This method builds multiple decision tree classifiers in parallel on different data subsamples and determines the final result by plurality voting or averaging. Therefore, it reduces the issue of overfitting and enhances both prediction accuracy and control. A random forest model, which involves several decision trees, is typically shows better performance than a model has a basis of single decision tree [32].

XGBoost is a type of gradient boosting algorithm. XGBoost considers more comprehensive approaches when figuring out the best model and calculates the second-degree derivatives of the loss function. The loss function evaluates the closeness of each prediction to the actual class label. A lower loss value indicates better model performance. This helps minimize losses. XGBoost works well with large datasets [32].

One of the most common machine learning algorithms for addressing the classification challenges is SVM. SVM constructs a hyperplane or a set of hyperplanes in a multi-dimensional space. A hyperplane provides strong separation by maximizing the distance from the closest training data points in any class. In most cases, a larger margin decreases the generalization error of the classifier. SVM performs well in high-dimensional spaces. It can also work in nonlinearly separable datasets using kernel functions. Kernel functions like sigmoid, RBF, linear, and polynomial are popular choices in SVM. Nonetheless, SVM tends to perform poorly in the presence of noise, such as when target classes overlap [32]. Figure 3 shows the structure of SVM.

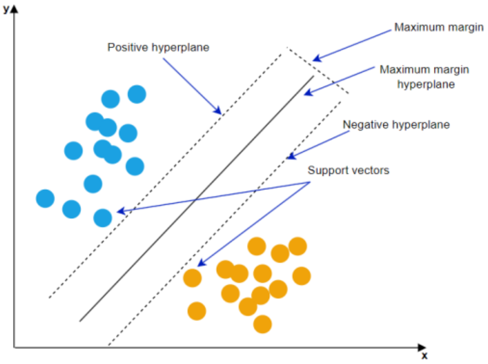

Figure 3: The structure of SVM.

MLP is a type of artificial neural network, and it is also referred to as a feedforward artificial neural network. A common MLP configuration includes an input layer, one or more hidden layers, and an output layer, forming a complete network. Each point in a layer is joined to all points in the next layer with a specific weight. MLP employs the backpropagation algorithm, a fundamental component of neural networks, to update weight values during model training. MLP is sensitive to how features are scaled and supports the tuning of hyperparameters, including the count hidden layers, neurons and iterations, which can be computationally expensive model [32]. Figure 4 presents an example of MLP with three inputs.

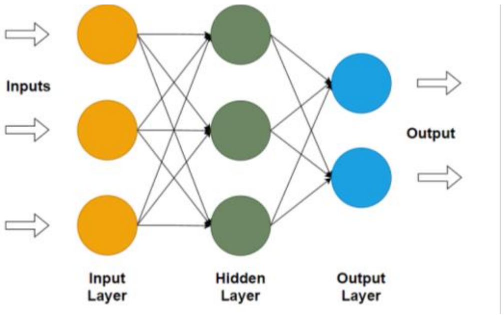

Figure 4: An example of MLP with three inputs.

3.4 Explainable Artificial Intelligence

SHAP is a technique that uses Shapley values to explain the inner workings of AI methods, often seen as black boxes. It employs concepts from game theory to measure the impact of individual features on the machine learning model's predictions. By leveraging Shapley values, SHAP determines the effect of each feature on the model's predictive performance. The primary goal of this method is to determine the contribution of each feature to the model's outcome as shown in Figure 5. SHAP treats each instance as a coalition game of features and calculates their contributions to the model's outcome. This calculation depends on the data distribution and the model's prediction function. SHAP is a popular method in the field of XAI and is widely used in various studies [33]. The developed model in this study utilized the SHAP method.

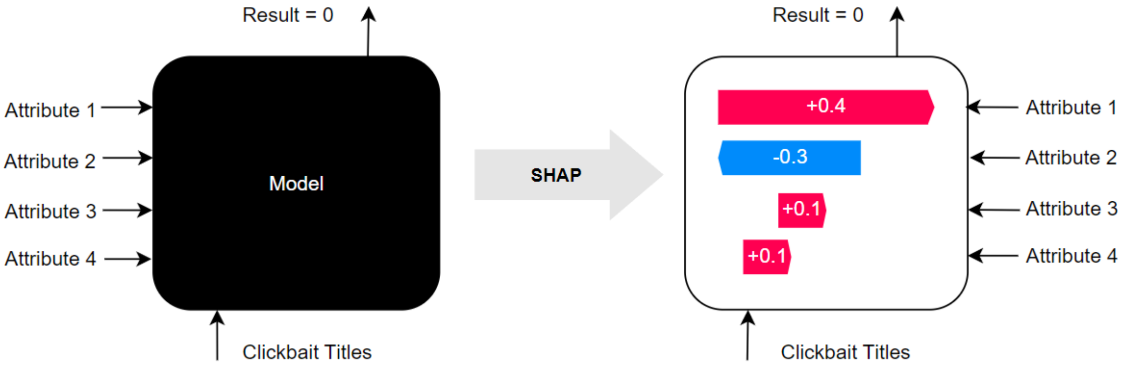

Figure 5: SHAP Method

3.5 Proposed Model

The framework of the model used in this paper is shown in Figure 6. First, to make the raw data usable, missing data analysis, visualization, feature engineering, stemming, TF-IDF transformation and normalization were employed. Then, machine learning models were trained for clickbait detection using the newly formed data. Finally, SHAP method was used to explain the factors effecting clickbait detection.

4 Experimental Study

In this part, the experimental studies and the results are explained.

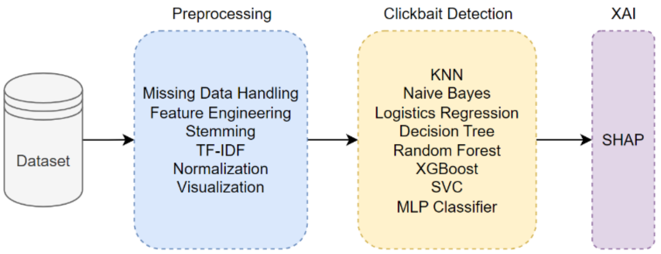

Figure 6: Model framework

4.1 Experimental Conditions

First, the dataset was checked for missing data. Two instances have no labels, so they were deleted. Nine features used by Potthast et al. [9] were created through feature engineering. These features are the number of three dots (threedotcount), the number of dots (dotcount), word count (wordcount), title length (length), the number of question marks (questionmarkcount), the number of exclamation marks (exclamationmarkcount), total punctuation count (punctuationcount), the number of uppercase letters (uppercasecount) and whether it starts with a number (startswithnumber). Due to the different scales of the features, min-max normalization was applied to bring all features to the same scale. Figure 7 shows the title length and number of question marks according to classes.

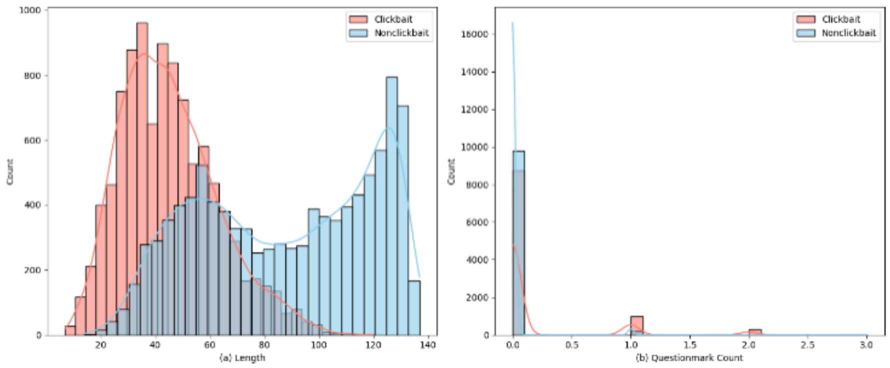

Figure 7: Length (a) and question mark (b) distribution according to classes.

Transformations such as case conversion and separation from punctuation marks were performed to process the news headlines numerically. Zemberek library was used to apply stemming to the headlines. Subsequently, a 12,144-dimensional vector was obtained through TF-IDF transformation.

After completing data preprocessing, the machine learning methods explained in Section 3.2 were applied for clickbait detection using the scikit-learn and XGBoost libraries in Python. The dataset was split, using 80% for training purposes and 20% for testing sets. GridSearchCV was used for hyperparameter optimization with 5-fold cross-validation. The best hyperparameters for each method are provided in Table 1. Machine learning models were developed using the parameters obtained from hyperparameter optimization, and each model was run five times.

Metrics such as accuracy, precision, recall, and F1-score are used for evaluating the performances of machine learning models [34]. In this study, the number of clickbaits and non-clickbaits are 10,030 and 10,008, respectively, making the classification problem balanced. Therefore, performance evaluations were conducted using accuracy, precision, recall, and F1-score.

Table 1: *Hyperparameter Optimization and Results*

Method	Parameter(s)	Range	Selected Value
KNN	number of neighbors *(n_neighbors)*	{1, 3, 5, 7, 9}	9
	distance metric *(metric)*	{euclidean, manhattan, cosine}	euclidean
Logistic Regression	regularization type *(penalty)*	{l1, l2, elasticnet, none}	l2
	regularization strength *(C)*	(-4, 4, 20)	1.624
	optimization algorithm *(solver)*	{liblinear, lbfgs, newton-cg, sag, saga}	saga
	ratio between L1 and L2 regularization *(l1_ratio)*	{0, 1, 10}	0
Decision Tree	maximum depth of the tree *(max_depth)*	{3, 5, 10}	10
	minimum number of samples required in a leaf node *(min_samples_leaf)*	{1, 3, 5}	1
	maximum number of leaf nodes *(max_leaf_nodes)*	{10, 20, 40}	40
	function to measure split quality *(criterion)*	{gini, entropy}	gini
	minimum number of samples required to split a node *(min_samples_split)*	{2, 5, 8, 10}	2
	maximum number of features considered for a split *(max_features)*	{5, 6, 7, 8, 9}	6
Random Forest	number of trees *(n_estimators)*	{50, 100, 200, 500}	500
	maximum depth of each tree *(max_depth)*	{3, 5, 7, 9, None}	9
	minimum number of samples required to split a node *(min_samples_split)*	{2, 5, 8, 10}	2
	minimum number of samples required in a leaf node *(min_samples_leaf)*	{10, 20, 30, 40, 50}	10
	maximum number of features used to create a split in each tree *(max_features)*	{5, 6, 7, 8, 9}	9
XGBoost	maximum depth of a tree *(max_depth)*	{3, 5, 7}	7
	minimum weight required to split a node *(min_child_weight)*	{1, 3}	3
	number of trees *(n_estimators)*	{100, 300, 500}	300
	boosting method *(booster)*	{gbtree, gblinear}	gbtree
	fraction of samples used for training each tree *(subsample)*	{0.6, 0.8}	0.8
	fraction of features sampled for each tree *(colsample_bytree)*	{0.6, 0.8}	0.6
	minimum loss reduction required for a split *(gamma)*	{0, 0.1, 0.2}	0.1
	step size for each boosting iteration *(learning rate)*	{0.01, 0.1, 0.3}	0.1
SVM	regularization parameter *(C)*	{0.1, 1, 10, 100, 1000}	1
	kernel function *(kernel)*	{linear, poly, rbf, sigmoid}	rbf
	parameter for kernels *(gamma)*	{scale, auto}	scale
MLP	size of the hidden layers *(hidden_layer_size)*	{(50,), (100,), (50,50), (100,50)}	(100,)
	activation function *(activation)*	{tanh, relu, logistic}	logistic
	initial learning rate *(learning_rate_init)*	{0.01, 0.1, 0.3}	0.1

4.2 Results and Discussion

The results for accuracy, precision, recall and F-score for the machine learning methods are given in Table 2. The best performance was obtained by SVM in terms of accuracy. Nevertheless, the test accuracy values of XGBoost and logistic regression are also very close to SVM. XGBoost also achieved the best values in the context of precision and F-score. Note that SVM has the same average F-score in test dataset. By contrast, the decision tree has the worst performance.

Table 2: *Model Results (The Best Performer in Each Criterion is Highlighted in Bold)*

	Training Set								**Test Set**							
Method	**Mean Acc**	**StD Acc**	**Mean Prec**	**StD Prec**	**Mean Recall**	**StD Recall**	**Mean F-Score**	**StD F-Score**	**Mean Acc**	**StD Acc**	**Mean Prec**	**StD Prec**	**Mean Recall**	**StD Recall**	**Mean F-Score**	**StD F-Score**
KNN	0.839	0.006	0.763	0.008	**0.985**	0.003	0.860	0.004	0.817	0.007	0.743	0.010	**0.971**	0.007	0.841	0.004
NB	0.899	0.001	0.854	0.002	0.963	0.001	0.905	0.001	0.853	0.004	0.812	0.006	0.918	0.007	0.862	0.004
LR	0.929	0.001	0.915	0.002	0.947	0.001	0.931	0.001	0.887	0.004	0.871	0.004	0.907	0.003	0.889	0.004
DT	0.542	0.052	0.528	0.039	0.957	0.070	0.677	0.010	0.539	0.054	0.527	0.040	0.957	0.064	0.676	0.012
RF	0.813	0.006	0.741	0.005	0.965	0.005	0.838	0.005	0.807	0.011	0.737	0.012	0.956	0.005	0.832	0.008
XGBoost	0,922	0,001	0,910	0,002	0,937	0,002	0,923	0,001	0,888	0,005	**0,877**	0,005	0,902	0,007	**0,892**	0,005
SVM	**0.966**	0.000	**0.956**	0.001	0.978	0.001	**0.967**	0.000	**0.890**	0.006	0.875	0.007	0.910	0.006	**0.892**	0.006
MLP	0.954	0.019	0.954	0.018	0.955	0.027	0.954	0.019	0.869	0.002	0.871	0.013	0.867	0.017	0.869	0.003

In Table 2, the results of logistic regression, XGBoost and SVM are very close, so Friedman test [35] was applied to determine the models' statistical differences. The Friedman test is a non-parametric statistical test, and it is distribution independent [36]. For this purpose, it is suitable for comparison of multiple machine learning models. As indicated by the results of the Friedman test in Table 3, the p-value is bigger than the significance level (α = 0.05). This shows that there is no statistically significant difference among logistic regression, XGBoost and SVM.

Table 3: *Friedman Test's Result*

Test Statistics	**p-value**	**Threshold Value**	**Result**
5.2000	0.0743	0.0500	False

The results indicate that the high-dimensional data affects the KNN's performance negatively. Moreover, most feature values are zero, and the decision tree's performance is affected from this sparsity. On the other hand, logistic regression and SVM perform well on sparse matrices due to its regularization coefficients. Similarly, XGBoost achieves good performance with its regularization coefficients and feature selection mechanism. Thus, according to the results, modern machine learning methods provide better performances.

The most successful XGBoost model was analyzed using SHAP to examine the explainability of the model. The top 20 features with the greatest impact on the model are shown in Figure 8. The X-axis represents SHAP values, which show the average effect of each feature on the model's result. According to the figure, the number of three dots and exclamation marks have the strongest effect on the model. As you move down the graph, the effect of the features on the model decreases.

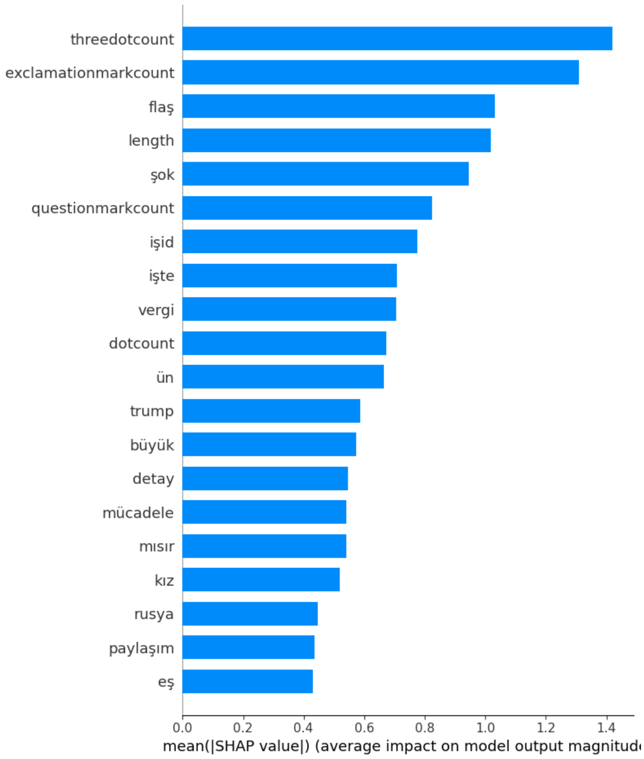

Figure 8: Bar plot of the model

The summary plot in Figure 9 shows the impact of each feature. The X-axis represents SHAP values, indicating the effects of all features on the model outcome. High feature values are indicated by red dots while low feature values are indicated by blue dots. The top features have the greatest impact on the model. Positive SHAP values make it more likely that the model will classify the example as clickbait, while negative SHAP values make it more likely that the model will classify the example as non-clickbait. In this context, the use of three dots is positively related to the likelihood of being clickbait. The use or high number of exclamation marks increases the probability of being clickbait. When evaluating the title length feature, which has a wide range, it is observed that as the length of the titles increases, the SHAP value takes negative values, thus reducing the likelihood of the title being clickbait. An increase in the number of question marks increases the likelihood of the title being clickbait. An increase in the number of dots in the title drives SHAP values towards negative, reducing the likelihood of the title being clickbait. The use of words like flaş (flash), şok (shock), işte (here) and paylaşım (share) results in positive SHAP values, increasing the likelihood of being clickbait. When the frequency of words like ün (fame), büyük (big), detay (detail), kız (girl) and eş (spouse) is low, SHAP values are positive, but as the number of these words increases, SHAP values approach to zero, increasing the likelihood of being nonclickbait. On the other hand, the use of words like işid and mısır (Egypt or corn) decreases the likelihood of being clickbait. Additionally, the words like vergi (tax), trump, mücadele (struggle) and rusya (Russia) show negative SHAP values. However, as the number of these words increases, SHAP values approach to zero, reducing the likelihood of being non-clickbait. If the title is short and contains attention-grabbing words like işte and flaş, the likelihood of the title being clickbait is higher.

Figure 10 shows the dependence plot for the number of exclamation marks and title length. The number of exclamation marks are given in the X-axis. The Y-axis shows the impact of the number of exclamation marks on the model with SHAP values. The color scale represents the length of the title. According to this graph, when exclamation marks are not used, SHAP values take small positive values in short titles, while SHAP values become positive as the title is longer. This increases the likelihood of being clickbait. In short, titles having a high number of exclamation marks further increases the likelihood of being clickbait. In line with these findings, Biyani et al. highlighted that the number of exclamation marks is among the most important features [37]. Similarly, Genç demonstrated that the number of exclamation marks is a significant feature in clickbait detection [17].

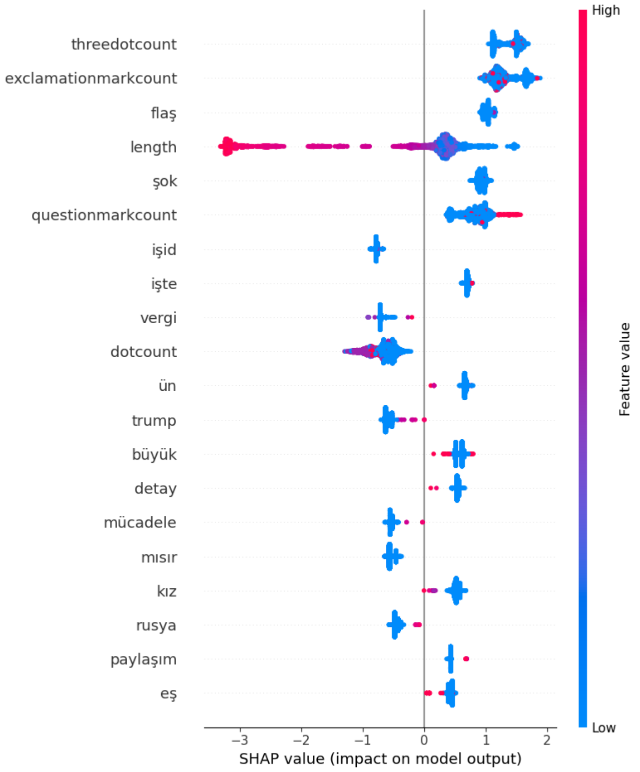

Figure 9: Summary plot of the model

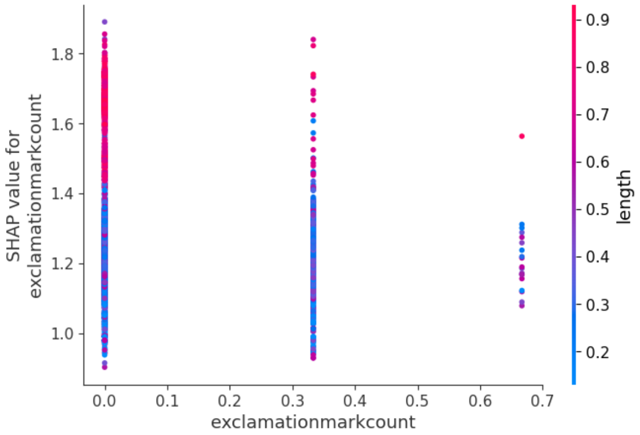

Figure 10: Dependence plot between exclamation mark and length

The relationship between the number of dots and length is shown in Figure 11. Accordingly, as the title length increases, the SHAP value decreases and even moves in a negative direction. This is another graph supporting that long titles are less likely to be clickbait. Short titles (between 0.0 and 0.5) have higher SHAP values, and are more likely to be clickbait compared to the long titles. After the value of 0.6, the use of dots increases, and SHAP values sharply move in a negative direction. From the color scale, it can be seen that the number of dots generally increases as the title length increases. If the number of dots is low in short titles, this increases the likelihood of short titles being clickbait. Potthast et al. also demonstrated the impact for the number of periods on clickbait detection [9]. Genç mentioned that the number of dots is among the effective features in clickbait detection [17].

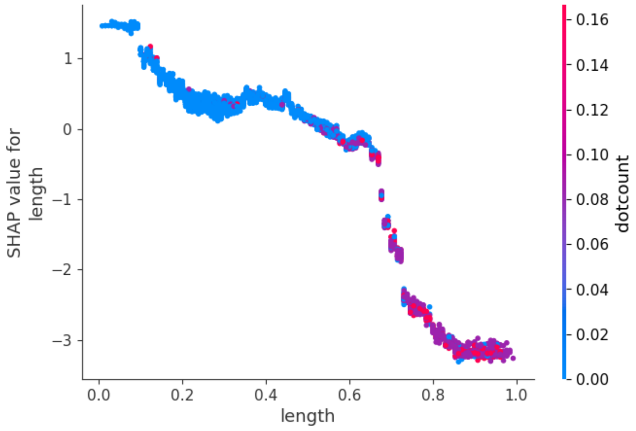

Figure 11: Dependence plot between length and number of dots

Waterfall plots created with two randomly selected examples are shown in Figures 12 (Example-1) and 9 (Example-2). The E[f(x)] value, which is equal to 0.009, represents base value of the model. For Example-1, in Figure 12, the features such as the number of three dots, exclamation marks, question marks and words like flaş, şok, işte and ün have positive SHAP values, while işid, vergi words and cumulative total of other features have negative SHAP values. With the addition of SHAP values , the base value of 0.009 changed to value of 0.162 in the graph. Therefore, it can be said that this title is classified as clickbait.

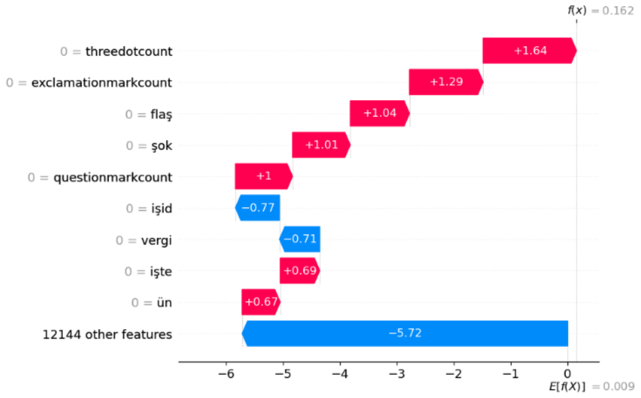

Figure 12: Waterfall plot for Example-1

For Example-2, the waterfall plot in Figure 13 shows that the length value is 0.885 (a long title), which has a negative SHAP value. It also corresponds to the feature that has the highest impact on the model. Additionally, dot count, word count features, işid and vergi words, and cumulative total of other features have negative SHAP values. The number of exclamation marks, the number of three dots, words flaş and şok words contributed positively on the nonclickbait decision. Also, value of f(x) is calculated as -4.114, and this example is not clickbait.

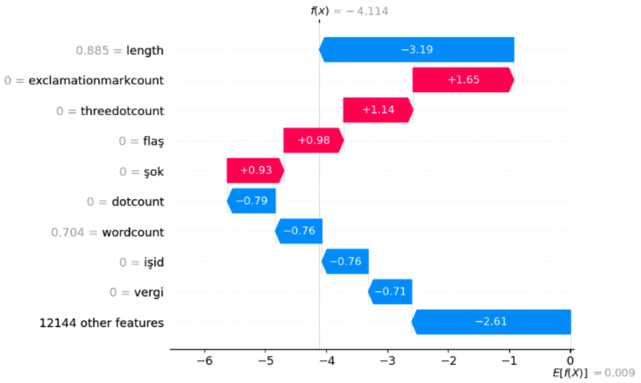

Figure 13: Waterfall plot for Example-2

Figure 14 demonstrates the force plot for Example-2. Accordingly, the number of exclamation marks, the number of three dots and word flaş increase the likelihood of this title being clickbait, while the length of the title prevents it from being clickbait. Additionally, it can be said that the model evaluates the likelihood of this title being clickbait as low. One of the features that contributes most to this prediction is the length of the title, meaning that longer titles decrease the probability of being classified as clickbait.

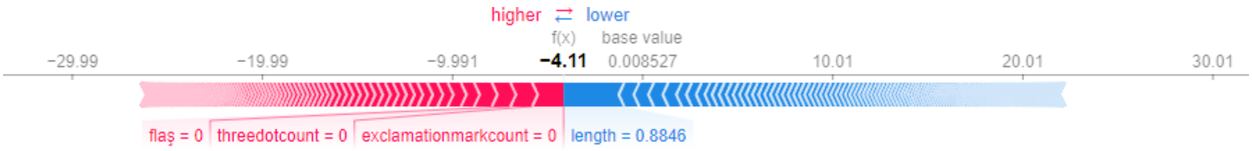

Figure 14: Force plot for Example-2

5 Conclusions

Clickbait content and clickbait producers are increasing day by day on the internet. Therefore, there is a growing need for artificial intelligence models that are capable of automatically detecting clickbait. Moreover, there is still a significant need for research on XAI models.

This study contributes to the literature with an explainable artificial intelligence model for Turkish clickbait detection. The model was developed using data mining, machine learning, and XAI methods. Eight classification methods were implemented. When these methods were compared based on test accuracy, the decision tree showed the worst performance, while SVM demonstrated the best performance. The result of the Friedman test revealed no significant difference among the performances of SVM, XGBoost and logistic regression.

Examining the graphs created with XGBoost and SHAP from both local and global perspectives, it became clear that the features obtained through feature engineering had an important impact on the model's decisions. Specifically, the five most important features in detecting clickbait news are the number of three dots, the number of exclamation marks, title length, and the words flaş and şok.

In the future, a comparative study can be conducted with different models and different datasets. Also, the effects of diversified features on Turkish clickbait detection can be investigated with other explainable artificial intelligence methods.

6 Declarations

6.1 Study Limitations

-

6.2 Acknowledgements

Part of this work was presented orally at the 6th International Conference on Data Science and Applications 2024 (ICONDATA'24).

6.3 Funding source

-

6.4 Competing Interests

There is no conflict of interest in this study.

6.5 Authors' Contributions

Corresponding Author: Developing ideas or hypotheses for the research and article, searching and gathering data for data analysis, taking responsibility for the experiments, organizing and reporting the data, taking responsibility for the explanation and presentation of the results, taking responsibility for the literature review during the research, taking responsibility for the creation of the entire manuscript.

2. Author: Developing ideas or hypotheses for the research and article, developing scientific model of the article, planning the materials and methods to reach the results, taking responsibility for the literature review during the research, taking the responsibility of project and process design, discussing the results and academic parts of the article, validation of the results, reworking not only in terms of spelling and grammar but also intellectual content.

References

[1] We Are Social, "Digital 2023 Global Overview Report," [Online]. Available: https://wearesocial.com/wp-content/uploads/2023/03/Digital-2023-Global-Overview-Report.pdf. Accessed: Aug. 9, 2024.

[2] Z. B. Şahin and Y. Birincioğlu, "Tık odaklı başlıklar ve okuyucu refleksleri üzerine bir araştırma: Odak grup çalışması," *TRT Akademi*, vol. 7, no. 14, pp. 236–261, 2022.

[3] R. Raj, C. Sharma, R. Uttara, and C. R. Animon, "A Literature Review on Clickbait Detection Techniques for Social Media," *Proc. 2024 11th Int. Conf. Reliability, Infocom Technol. Optimization (ICRITO)*, pp. 1–5, Mar. 2024. http://dx.doi.org/10.1109/ICRITO61523.2024.10522359

[4] A. F. H. N. Adrian, N. N. Handradika, A. E. Prasojo, A. A. S. Gunawan, and K. E. Setiawan, "Clickbait Detection on Online News Headlines Using Naive Bayes and LSTM," *Proc. 2024 IEEE Int. Conf. Artificial Intell. Mechatronics Syst. (AIMS)*, pp. 1–6, Feb. 2024. https://doi.org/10.1109/AIMS61812.2024.10512986

[5] Y. Arfat and S. C. Tista, "Bangla Misleading Clickbait Detection Using Ensemble Learning Approach," *Proc. 2024 6th Int. Conf. Electrical Eng. Inf. Commun. Technol. (ICEEICT)*, pp. 184–189, May 2024. https://doi.org/10.1109/ICEEICT62016.2024.10534333

[6] W. Yang, Y. Wei, H. Wei, Y. Chen, G. Huang, X. Li, and B. Kang, "Survey on explainable AI: From approaches, limitations and applications aspects," *Human-Centric Intell. Syst.*, vol. 3, no. 3, pp. 161–188, 2023. https://doi.org/10.1007/s44230-023-00038-y

[7] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, pp. 1135–1144, Aug. 2016. https://doi.org/10.1145/2939672.2939778

[8] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017. https://doi.org/10.48550/arXiv.1705.07874

[9] M. Potthast, S. Köpsel, B. Stein, and M. Hagen, "Clickbait detection," *Adv. Inf. Retrieval: Proc. 38th European Conf. IR Res. (ECIR)*, pp. 810–817, Mar. 2016. https://doi.org/10.1007/978-3-319-30671-1_72

[10] A. Chakraborty, B. Paranjape, S. Kakarla, and N. Ganguly, "Stop clickbait: Detecting and preventing clickbaits in online news media," *Proc. 2016 IEEE/ACM Int. Conf. Advances Social Networks Anal. Mining (ASONAM)*, pp. 9–16, Aug. 2016. https://doi.org/10.1109/ASONAM.2016.7752207

[11] K. K. Yadav and N. Bansal, "A Comparative Study on Clickbait Detection using Machine Learning Based Methods," *Proc. 2023 Int. Conf. Disruptive Technol. (ICDT)*, pp. 661–665, May 2023. https://doi.org/10.1109/ICDT57929.2023.10150475

[12] A. Chowanda, N. Nadia, and L. M. M. Kolbe, "Identifying clickbait in online news using deep learning," *Bull. Electrical Eng. Informatics*, vol. 12, no. 3, pp. 1755–1761, 2023. https://doi.org/10.11591/eei.v12i3.4444

[13] C. I. Coste, D. Bufnea, and V. Niculescu, "A new language independent strategy for clickbait detection," *Proc. 2020 Int. Conf. Software, Telecommun. Comput. Networks (SoftCOM)*, pp. 1–6, Sep. https://doi.org/10.23919/SoftCOM50211.2020.9238342

[14] M. M. Mahtab, M. Haque, M. Hasan, and F. Sadeque, "Banglabait: Semi-supervised adversarial approach for clickbait detection on Bangla clickbait dataset," *14th International Conference on Recent Advances in Natural Language Processing*, pp 748–758, 2023. https://doi.org/10.26615/978-954-452-092-2_081

[15] D. M. Broscoteanu and R. T. Ionescu, "A Novel Contrastive Learning Method for Clickbait Detection on RoCliCo: A Romanian Clickbait Corpus of News Articles," *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9547–9555, 2023. https://doi.org/10.18653/v1/2023.findings-emnlp.640

[16] T. Liu, K. Yu, L. Wang, X. Zhang, H. Zhou, and X. Wu, "Clickbait detection on WeChat: a deep model integrating semantic and syntactic information," *Knowledge-Based Syst.*, vol. 245, p. 108605, 2022. https://doi.org/10.1016/j.knosys.2022.108605

[17] Ş. Genç, "Turkish clickbait detection in social media via machine learning algorithms," MSc Thesis, Middle East Technical University, Ankara, 2021. https://hdl.handle.net/11511/92039

[18] K. Shu, L. Cui, S. Wang, D. Lee, and H. Liu, "DEFEND: Explainable fake news detection," *Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, pp. 395–405, Jul. 2019. https://doi.org/10.1145/3292500.3330935

[19] S. Y. Chien, C. J. Yang, and F. Yu, "XFlag: Explainable fake news detection model on social media," *Int. J. Human–Comput. Interaction*, vol. 38, no. 18-20, pp. 1808–1827, 2022. https://doi.org/10.1080/10447318.2022.2062113

[20] V. Sharma and D. Midhunchakkaravarthy, "XGBoost classification of XAI based LIME and SHAP for detecting dementia in young adults," *Proc. 2023 14th Int. Conf. Comput. Commun. Networking Technol. (ICCCNT)*, pp. 1–6, Jul. 2023. https://doi.org/10.1109/ICCCNT56998.2023.10307791

[21] G. I. Pérez-Landa, O. Loyola-González, and M. A. Medina-Pérez, “An explainable artificial intelligence model for detecting,” *Human-Centric Intell. Syst.*, vol. 2, no. 3, pp. 160–188, 2021. https://doi.org/10.3390/app112210801

[22] M. Zhou, W. Xu, W. Zhang, and Q. Jiang, “Leverage knowledge graph and GCN for fine-grained-level clickbait detection,” *World Wide Web*, vol. 25, no. 3, pp. 1243–1258, 2022. https://doi.org/10.1007/s11280-022-01032-3

[23] T. Turan, E. Küçüksille, and N. K. Alagöz, “Prediction of Turkish Constitutional Court decisions with explainable artificial intelligence,” *Bilge Int. J. Sci. Technol. Res.*, vol. 7, no. 2, pp. 128–141, 2023. https://doi.org/10.30516/bilgesci.1317525

[24] S. Rao, S. Mehta, S. Kulkarni, H. Dalvi, N. Katre, and M. Narvekar, “A study of LIME and SHAP model explainers for autonomous disease predictions,” *Proc. 2022 IEEE Bombay Sect. Signature Conf. (IBSSC)*, pp. 1–6, Dec. 2022. https://doi.org/10.1109/IBSSC56953.2022.10037324

[25] Turkish News Title 20000+ Clickbait Classified, [Online]. Available: https://www.kaggle.com/datasets/suleymancan/turkishnewstitle20000clickbaitclassified . Accessed: Aug. 9, 2024.

[26] P. Domingos, “A few useful things to know about machine learning,” *Commun. ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012. http://dx.doi.org/10.1145/2347736.2347755

[27] A. A. Akın and M. D. Akın, “Zemberek, an open source NLP framework for Turkic languages,” *Structure*, vol. 10, pp. 1–5, 2007.

[28] K. Sparck Jones, “A statistical interpretation of term specificity and its application in retrieval,” *J. Documentation*, vol. 28, no. 1, pp. 11–21, 1972. https://doi.org/10.1108/eb026526

[29] A. Kiran and D. Vasumathi, “Data mining: Min–max normalization based data perturbation technique for privacy preservation,” *Proc. Third Int. Conf. Comput. Intell. Informatics: ICCII 2018*, pp. 723–734, Mar. 2020. https://doi.org/10.1007/978-981-15-1480-7_66

[30] N. A. Zuhroh and N. A. Rakhmawati, “Clickbait detection: A literature review of the methods used,” *Register: J. Ilmiah Teknologi Sistem Informasi*, vol. 6, no. 1, pp. 1–10, 2022. http://dx.doi.org/10.26594/register.v6i1.1561

[31] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, “Supervised machine learning: A review of classification techniques,” *Emerg. Artificial Intell. Applications Comput. Eng.*, vol. 160, no. 1, pp. 3–24, 2007.

[32] I. H. Sarker, “Machine learning: Algorithms, real-world applications and research directions,” *SN Comput. Sci.*, vol. 2, no. 3, p. 160, 2021. https://doi.org/10.1007/s42979-021-00592-x

[33] G. Van den Broeck, A. Lykov, M. Schleich, and D. Suciu, “On the tractability of SHAP explanations,” *J. Artificial Intell. Res.*, vol. 74, pp. 851–886, 2022. https://doi.org/10.1613/jair.1.13283

[34] D. M. Powers, “Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation,” *International Journal of Machine Learning Technology* vol. 2, no. 1, pp.37-63, 2011. https://doi.org/10.48550/arXiv.2010.16061

[35] M. Friedman, “The use of ranks to avoid the assumption of normality implicit in the analysis of variance,” *J. American Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, 1937. https://doi.org/10.1080/01621459.1937.10503522

[36] A. Benavoli, G. Corani, and F. Mangili, “Should we really use post-hoc tests based on mean-ranks?” *J. Machine Learn. Res.*, vol. 17, no. 1, pp. 152–161, 2016. https://doi.org/10.48550/arXiv.1505.02288

[37] P. Biyani, K. Tsioutsiouliklis, and J. Blackmer, “‘8 amazing secrets for getting more clicks’: Detecting clickbaits in news streams using article informality,” *Proc. 2016 AAAI Conf. Artificial Intell.*, vol. 30, no. 1, 2016. https://doi.org/10.1609/aaai.v30i1.9966