Makine Öğrenmesi Algoritmalarının Çıkarımsal Metin Özetlemede Etkililiği: K-Ortalamalar, Rastgele Orman, GBM, Lojistik Regresyon ve SVM'in Karşılaştırmalı Analizi

Taner Uçkan [1*], Kübra Karabulut[2]

[1*]Van Yüzüncü Yıl Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Böülümü,Van ,Türkiye ,taneruckan@yyu.edu.tr
[2]Van Yüzüncü Yıl Üniversitesi, Fen Bilimleri Enstitüsü, Yapay Zekâ ve Robotik Anabilim Dalı, Van, Türkiye

e-mail: taneruckan@yyu.edu.tr
DOI: 10.57244/dfbd.1538959
Geliş tarihi/Received:26/06/2024 Kabul tarihi/Accepted:11/11/2024

Özet

Günümüzdeki bilgi çağında, veri setlerinden anlamlı bilgi çıkarımı, her zamankinden daha fazla önem arz etmektedir. Çeşitli alanlarda veriler, detaylı aşamalarıyla veri tabanı sistemlerinde saklanmakta ve bu durum, daha büyük veri setleriyle karşı karşıya kalmamıza neden olmaktadır. Büyük veri dönemi, makine öğrenmesi tekniklerinin kullanılması yoluyla çıkarımsal metin özetlemenin, önemli bir araştırma alanı haline gelmesini sağlamıştır. Çıkarımsal özetleme, orijinal metinden önemli bilgileri seçerek daha kısa ve öz bir versiyon oluşturmayı amaçlamaktadır; bu süreç, metnin boyutunu küçültürken temel bilgilerin korunmasına odaklanır. Bu çalışmanın amacı, metin özetleme süreçlerini insan müdahalesi olmaksızın gerçekleştirebilmek ve orijinal metnin anlamını koruyarak etkili bir özet üretimini sağlamak için makine öğrenimi modellerinin kullanımını araştırmaktır. Makine öğrenimi temelli modeller ve çıkarımsal metin özetleme, bilgisayarların insan bilgisi ve dil yeteneğinden yoksun olmasından kaynaklanan zorlukları aşmak için çeşitli çözümler sunmaktadır. Bu çalışmada, çıkarımsal metin özetlemenin, belgelerden doğrudan cümleler alarak tutarlı bir özet oluşturmasına dair makine öğrenimi yaklaşımlarının uygulanabilirliği ve etkililiği test edilmiştir. K-Ortalamalar, Rastgele Orman, Gradyan Arttırma Makineleri (GBM)/XGBoost, Lojistik Regresyon ve Destek Vektör Makineleri (SVM) gibi makine öğrenimi yöntemleri, mevcut metin veri setleri üzerinde karşılaştırmalı olarak incelenmiş ve performansları değerlendirilmiştir.

Anahtar Kelimeler: Çıkarımsal Metin Özetleme, Makine Öğrenmesi, Otomatik Metin Özetleme, Rouge

The Effectiveness of Machine Learning Algorithms in Extractive Text Summarization: A Comparative Analysis of K-Means, Random Forest, GBM, Logistic Regression, and SVM

Abstract

In today's information age, extracting meaningful infor-mation from datasets has become more critical than ever. Data in most fields is stored in detail within database systems, leading to the encounter with larger datasets. In the era of big data, extractive text summarization using machine learning methods has become a significant area of study. Extractive summarization aims to create a concise version of the original text by extracting essential information. This process generally aims to re-duce the text's size while preserving key information. Our study aims to achieve a brief and fluent summary that maintains the original text's meaning without human intervention through machine learning models. Machine learning-based models and extractive text summarization offer various approaches to solve this challenging task, as computers lack hu-man knowledge and language ability. In this study, the feasibility and ef-fectiveness of machine learning methods were tested by forming coherent summaries directly from document sentences using extractive text summa-rization. Algorithms such as K-Means, Random Forest, Gradient Boosting Machines (GBM) / XGBoost, Logistic Regression, and Support Vector Machines (SVM) were tested, and their performance was comparatively evaluated on the "BBC News Summary" text datasets.

Keywords: Extractive Text Summarization, Machine Learning, Automatic Text Summarization, Rouge

Introduction

Nowadays, the ease of obtaining digital data and the increase in the amount of information have heightened the need for effective text summarization. Traditional text summarization methods produce superficial information, leading to a need for more extractive and meaningful summaries.

Text summarization presents the main ideas and important points of a text in a condensed form. Traditional text summarization methods are based on features such as word frequency, sentence position, title, and keywords. The first studies on text summarization were encountered in the 1950s (Luhn, 2010). These early studies focused on calculating various features of sentences and using these features to determine their importance. When word frequency was found to be insufficient, methods considering the position of the sentence within the document were developed. However, these methods produced shallow and superficial summaries as they could not fully capture the main points and semantic connections of the text. Extractive text summarization aims to create more concise and meaningful summaries by utilizing deep semantic relationships and inferences within the text. These methods can better model the meaning and structure of the text using machine learning and deep learning techniques. Through this approach, logical connections, conceptual meanings, and extractive relationships can be captured more effectively. While traditional text summarization methods produce summaries based on superficial features of the text, extractive text summarization methods create more concise, understandable, and informative summaries by relying on deeper and more meaningful information. In this method, the use of machine learning and deep learning also helps automate the extractive summarization process and increase scalability Machine learning is a branch of science that enables computers to learn from data without human intervention and perform specific tasks using the knowledge obtained from this learning. It can be used in many fields and provides great convenience. Methods such as clustering, classification, regression, forecasting, and decision support significantly impact our lives. Tasks that would take a long time and complex processes can now be done easily, and developments in this area continue.

Machine learning is also used in fields such as natural language processing and computer vision. Machine learning algorithms can be used for tasks like understanding human language, summarizing text, recognizing images, and detecting objects. The first examples of text summarization studies using machine learning methods were encountered in the 2000s (Olmez, 2024). With advancements in data storage and processing technologies, better summaries have been created using machine learning methods. It is predicted that in the future, these technologies will present new applications that will simplify and optimize our lives. It can be said that they will support and transform human activities with their automatic decision-making, prediction, and learning capabilities.Various studies have been conducted on extractive text summarization. In extractive summarization, important sentences in the text are selected to create a summary. In a 2010 study by Gupta and colleagues, significant progress was made in extractive text summarization by using machine learning and natural language processing techniques together. They compared various summarization techniques and examined how texts can be summarized more effectively (Gupta, intelligence, & 2010, 2010). Additionally, Nenkova and McKeown presented a comprehensive review in 2012 on evaluating and improving summarization techniques, proposing different methods and approaches to enhance the performance of summarization techniques (Nenkova & McKeown, 2012).

In addition to these studies, different approaches and methods are also present in the literature. In their 2023 study, Aydın and Uçkan proposed an extractive text summarization method based on independent clusters in text graphs. This study aims to convert texts into graphs, create independent clusters, and select important sentences from these clusters (Aydın and Uçkan, 2023). Erhandı used deep learning methods for text summarization in 2020. In his study, he investigated the effectiveness of text summarization using deep learning algorithms and achieved successful results (Erhandı, 2020).There have also been studies on processing and classifying court decisions using machine learning methods. These studies examine the applicability of text summarization methods in the legal field. In their 2023 study, Görentaş and Uçkan analyzed the content and structure of court decisions and created a study that clusters similar decisions using machine learning methods. This study aims to group court decisions according to specific criteria (Görentaş and Uçkan, 2023). Additionally, in a similar 2023 study, Görentaş and colleagues aimed to classify dispute court decisions based on content and structure. This study provided more effective and faster classification of dispute court decisions (Görentaş et al., 2023).

These studies significantly contribute to extractive text summarization using machine learning methods and automatic processing of legal texts, where the decision on the text is important. Martin Katz and colleagues, in a 2017 study, aimed to predict the behavior of the United States Supreme Court and increase the predictability of its decisions using machine learning techniques. They used various machine learning algorithms for analyzing and classifying court decisions in their study (Martin Katz, Bommarito, & Blackman, 2017). Mumcuoğlu and colleagues conducted a text analysis using artificial intelligence methods on Turkish Constitutional Court decisions in 2021 (Mumcuoğlu, Öztürk, Ozaktas, &, & 2021, 2021). These studies reveal the strengths and weaknesses of various methods used in text summarization and demonstrate the applicability of machine learning methods in the legal field. Studies in different fields also exist. In a 2019 study, Alpkoçak and colleagues used machine learning methods to detect cyberbullying in Turkish texts (Alpkoçak, Tocoglu, Çelikten, & Aygün, 2019). In a similar study, Yazğılı and Baykara conducted cyberbullying detection on Turkish texts using machine learning methods in 2022 (Yazğili and Baykara, 2022). Cyberbullying detection has become an important research area with the advancement of information technologies, and these studies can provide positive results to prevent such bullying. These studies are pioneering in extractive text summarization using machine learning methods and automatic processing, offering methods and results that guide other studies in this field.

In this study, the aim is to develop extractive text summarization models using machine learning methods. Various machine learning methods, such as Support Vector Machines (SVM), Gradient Boosting Machines (GBM) / XGBoost, Random Forest Algorithm, K-Means, and Logistic Regression, have been used to develop more advanced text summarization models. According to evaluation metrics, the most successful summarization was achieved with the Logistic Regression model.

This paper is organized into several sections following the introduction. First, the dataset used in the study and the data preprocessing steps are detailed. Next, the machine learning algorithms employed for extractive text summarization (K-Means, Random Forest, Gradient Boosting Machines, Logistic Regression, and Support Vector Machines) are presented alongside their performance evaluations. In the findings and discussion section, the comparisons of different algorithms based on Rouge metrics are conducted, and the results obtained are analyzed. Finally, the contributions of the study to the field of summarization and recommendations for future research are addressed in the conclusion section.

Material and Method

Dataset

The dataset used in the study is BBC News Summary. This dataset is prepared for inferential text summarization. It contains a total of 2225 news articles from BBC from 2004 to 2005 in the categories of Business, Entertainment, Politics, Sports and Technology. There are five summaries for each article in the summaries folder. The first sentence of the article text is the relevant title of the news. (Sharif, 2018).

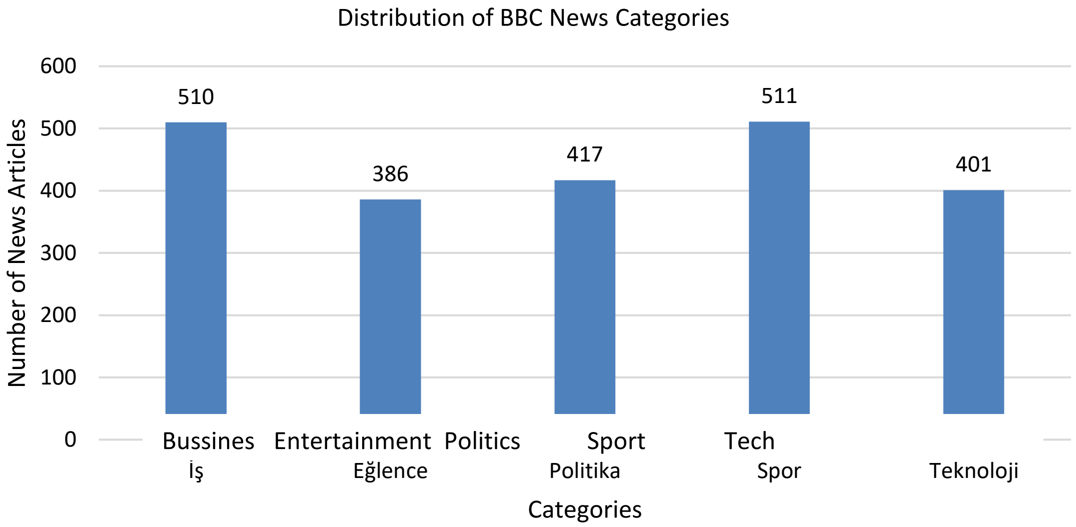

Figure 1. Distribution of BBC News Categories

Method

Text preprocessing was applied to the texts in these categories in the dataset. The raw text was preprocessed to make it suitable for further processing. The N-Gram model was used for feature extraction. The TF-IDF method was chosen to convert words into numerical form. After completing the necessary steps, models were created using K-Means, Random Forest, Gradient Boosting Machines (GBM) / XGBoost, Logistic Regression, and Support Vector Machines (SVM) algorithms. Analyses were performed, and the results were compared using evaluation metrics.

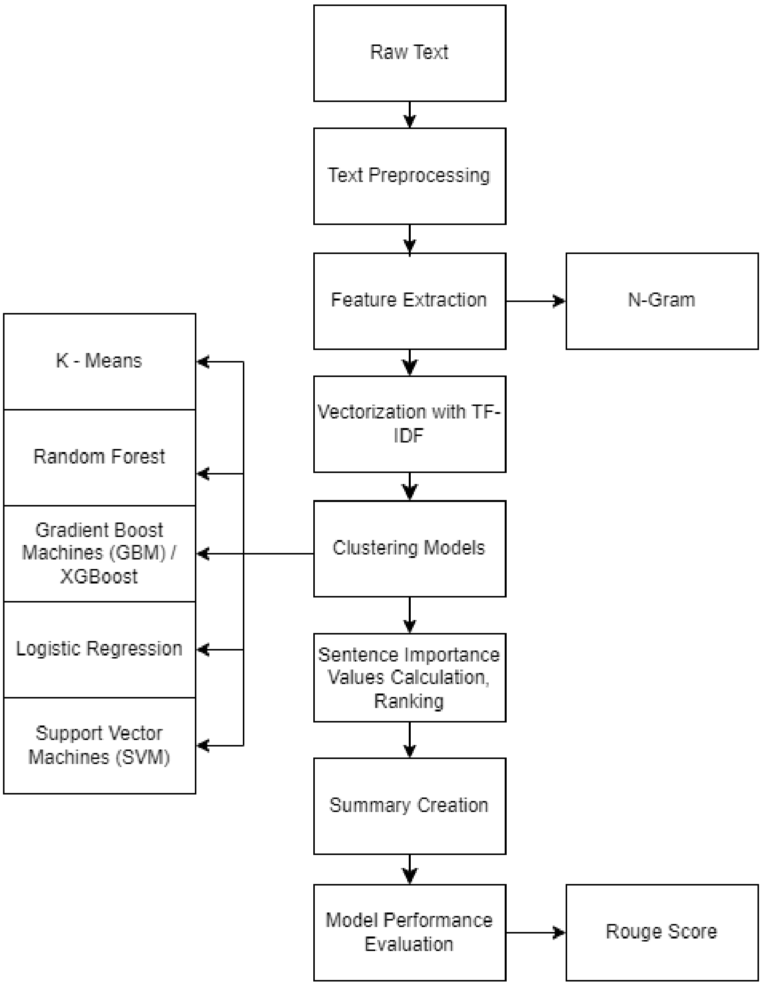

Figure 2. Flowchart Of Proposed Model

Feature Extraction with N-Gram

N-gram is one of the methods used to mathematically express consecutive sequences of words in a sentence. It is done using a probabilistic model and is typically used to calculate the likelihood of a specific word appearing, considering all the words in the entire text. It helps us understand and model the language structure in texts. When used together with existing successful methods, it has been found to complement the shortcomings of those methods (Özmutlu & Çağlar, 2009). It allows for better predictions and analyses in fields like text mining and natural language processing.

In an n-gram model, the probability of a word occurring depends on the previous (n-1) words. This probability is expressed as follows:

$$P(w_i | w_i - (n-1), \ldots, w_i - 1) \quad (1)$$

P : Probability,

w_i : Examined word,

$w_i - (n-1), \ldots, w_i - 1$: Previous (n-1) word.

TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) is a vectorization technique widely used in text mining and information retrieval. TF-IDF is used to calculate the importance of words in a document. It computes the significance of a term in a document by considering the term's frequency in that document and the term's inverse frequency across all documents. Terms that occur frequently in a document but are rare across all documents get higher TF-IDF values, highlighting important terms in that document (Schütze, Manning, & Raghavan, 2008).

TF-IDF consists of two main components: Term Frequency (TF) and Inverse Document Frequency (IDF).

Term Frequency (TF)

TF measures how frequently a term (word) appears in a document. It is calculated as:

$$TF(t,d) = \frac{Frequency\ of\ Term\ in\ Document}{Total\ Number\ of\ Terms\ in\ Document} \tag{1}$$

t : Term (Word),

d : Document.

Inverse Document Frequency (IDF)

IDF measures how rare a term is across all documents. A term that appears in many documents is considered less important. IDF is calculated as:

$$IDF(t) = \log\left(\frac{N}{df(t)}\right) \tag{2}$$

N : Total number of documents,

$df(t)$: Number of documents containing the term.

TF-IDF Calculation

TF-IDF is calculated by multiplying the TF and IDF values to determine the importance of a term in a specific document:

$$TF - IDF(t,d) = TF(t,d) \times IDF(t) \tag{3}$$

Using this formula, terms that occur frequently in a document and are rare across all documents get high TF-IDF values, thus highlighting them as important terms in that document.

Clustering Models

K-Means

The K-Means algorithm is one of the well-known machine learning techniques that groups data into meaningful clusters. It clusters the input data points around a predetermined number of k centers. The algorithm follows an iterative process where each data point is assigned to the nearest center, clusters are formed, and then these centers are recalculated. This process continues until the goal of minimizing the distance between clusters is achieved. After these stages, data points with similar characteristics are gathered in the same clusters, revealing meaningful groups. The aim of K-Means is to reduce the sum of squared distances between the data points and their respective cluster centers. In this way, similarities among data points increase, and meaningful groups emerge.

Initially, k cluster centers are chosen:

$$\boldsymbol{a_1(1), a_2(1), a_3(1), \dots, a_k(1)} \tag{1}$$

The data points X are distributed into clusters according to the following relationship in k iterations:

$$\boldsymbol{X \in C_j(k)\ if\ \left\|x - a_j(k)\right\| < \|x - a_i(k)\|} \tag{2}$$

New cluster centers are calculated:

$$\boldsymbol{a_j(k+1) = \frac{1}{N_j}\sum\nolimits_{x \in C_j(k)} x} \tag{3}$$

$\boldsymbol{a_j(k+1)}$: New center of the j-th cluster,
$\boldsymbol{N_j}$: Number of data points assigned to the j-th cluster,
$\boldsymbol{x \in C_j(k)}$: Data points belonging to the cluster at the k-th iteration.

If:

$$\boldsymbol{a_j(k+1) = a_j(k)\ for\ all\ j = 1, 2, \dots, k} \tag{4}$$

the algorithm stops due to convergence (Khan, Qian, Information, & 2019, 2019).

Random Forest

Random Forest is a powerful machine learning algorithm created by combining multiple decision trees. Each decision tree is trained on randomly selected subsets of the training data and makes independent predictions. The algorithm combines the predictions of all the trees to make a final decision, without pruning like in individual decision trees. Thus, it can be described as an ensemble learning approach. Random Forest can be successfully applied to classification and regression problems.

Combining many trees generally provides higher accuracy than a single decision tree. Random Forest is more robust against overfitting problems found in Decision Trees. It can determine feature importance and perform well on large datasets (GÖRENTAŞ et al., 2023).

Basic components of the Random Forest algorithm:

Bootstrap Sampling: For each decision tree, n samples are selected from the training data using the resampling method.

Random Feature Selection: At each node, the best splitting feature is chosen from a randomly selected subset of m features. This ensures the trees are different from each other.

Decision Function: Each tree $h_i(\boldsymbol{x})$ makes a prediction. For classification, the final prediction $\hat{y}$ is determined by the majority vote of all the trees (Breiman, 2001):

$$\hat{y} = mode\{h_1(x), h_2(x), \dots, h_B(x)\} \tag{1}$$

For regression, the final prediction $\hat{y}$ is determined by the average of all the trees:

$$\hat{y} = \frac{1}{B}\sum_{i=1}^{B} h_i(x) \tag{2}$$

B : Total number of trees,

$h_i(x)$: i-th tree's prediction for input x.

Gradient Boosting Machines (GBM) / XGBoost

Gradient Boosting Machines (GBM) and XGBoost are powerful machine learning algorithms created by combining sequential weak classifiers. The algorithm creates a new weak classifier in each step by giving more weight to the examples misclassified in the previous iteration. Thus, by using the gradient descent method across iterations to reduce errors, a strong composite model is obtained. XGBoost is an advanced version of GBM, offering faster, more scalable, and higher-performing results. These algorithms can be successfully used for both classification and regression problems and show superior performance especially on high-dimensional, sparse, and noisy data. They are frequently preferred in text classification, natural language processing, finance, and other fields (Chen, international, & 2016, 2016).

Logistic Regression

Logistic Regression is a widely used classification algorithm in the field of machine learning. It is used for binary classification problems and models the effect of independent variables on the dependent variable. This method utilizes the logistic function to estimate the probability that a target variable belongs to a particular category. Logistic regression analysis is referred to as binary or multinomial logistic regression depending on the scale type and number of categories of the target variable. It predicts the probability that a data point belongs to a specific category (Görentaş Et Al., 2023).

Logistic Function (Sigmoid Function): Logistic regression takes the linear combination of independent variables and uses the logistic function to convert it into a probability between 0 and 1.

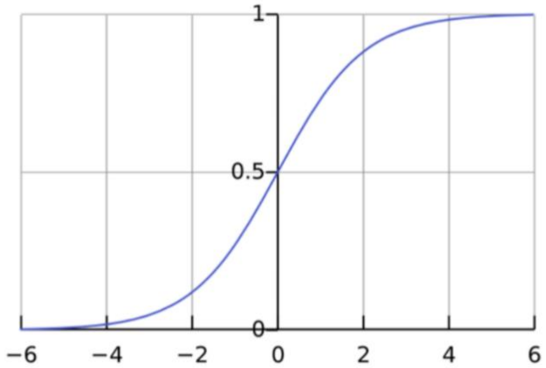

Figure 3. Sigmoid Function Curve for Logistic Regression Model [17]

The logistic function is defined as:

$$\boldsymbol{f(x) = \frac{1}{1 + e^{-x}}} \quad (\mathbf{1})$$

Model*:* The probability that the dependent variable belongs to a specific category is calculated using the logistic function.

$$x = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \quad (2)$$

β_0: Intercept term,

$\beta_1, \beta_2, \ldots, \beta_n$: Coefficients of independent variables,

$x_1, x_2, \ldots, x_n$: Independent variables.

Probability Estimate*:* The probability that the dependent variable belongs to a specific category is calculated using the logistic function.

$$P(y = 1|x) = \frac{1}{1 + e^{(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}} \quad (3)$$

$P(y = 1|x)$: Probability that the dependent variable is 1.

Logit Function*:* The core function of logistic regression. It transforms the linear combination of independent variables into a value between 0 and 1 using the logit function, ensuring that the dependent variable takes values only between 0 and 1. Logistic regression is used to predict the value of the dependent variable in cases where the data set is not linearly separable. Logistic regression separates data points using a linear decision boundary and is also used to predict the probability of a specific event.

Support Vector Machines (SVM)

Support Vector Machines (SVM) are powerful machine learning algorithms used for classification and regression problems. SVM aims to find a decision boundary that maximizes the separation margin between classes by mapping data samples into a high-dimensional space. This decision boundary is based on the principle of finding the widest margin between classes (Ayata & Çavuş, 2022). SVM performs well not only on linearly separable data but also on complex, non-linear problems by using kernel functions to map the data space into a high-dimensional space. With its ability to handle noisy and high-dimensional data, resistance to overfitting, and interpretability, SVM is frequently used in text classification, image processing, and other fields.

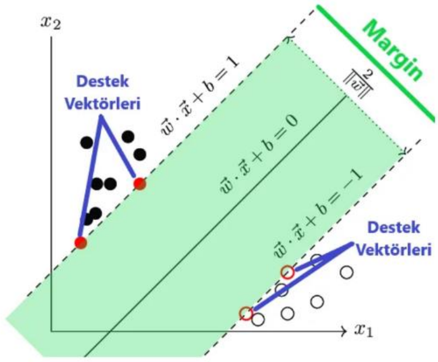

Figure 4. Support Vector Machine Model (Akca, 2020)

w: Weight vector,

x: Input vector,

b: Bias,

y: Class label ($y \in \{-1, +1\}$).

SVM classifies using a decision function:

$$\hat{y}(x) = sign(w^T x + b) \tag{1}$$

The margin (distance) between the decision boundary ($w^T x + b$) and the boundaries is maximized. Maximizing this margin enhances the generalization and performance of SVM.

$$\hat{y} = \begin{cases} 0 \ if \ w^T \ x + b < 0, \\ 1 \ if \ w^T \ x + b \geq 0 \end{cases} \tag{2}$$

If the result for a new value is less than 0, it will be closer to the white dots. Vice versa, if the result is greater than or equal to 0, then it will be closer to the black dots.

Evaluation Metrics

Rouge-1, Rouge-2, and Rouge-L are metrics used to measure the performance of machine learning-based summarization models by evaluating different aspects of these models. Rouge-1 measures the unigram overlap between the generated summary and the reference summary, which means it looks at individual word matches. Rouge-2 evaluates the bigram overlap, which refers to the matching of two-word sequences between the model-generated text and human-produced text. Rouge-L analyzes the longest common subsequences between the generated summary and the reference summary, reflecting the structural similarity at the sentence level (Chiusano, 2022).

These metrics help us understand how effective summarization models are. Rouge-1 typically reflects word-level accuracy, Rouge-2 indicates phrase-level accuracy, and Rouge-L demonstrates structural similarity at the sentence level. A high Rouge-1 score shows that the model accurately captures words from the reference text, indicating high word-level accuracy. A high Rouge-2 score means that the model achieves high phrase-level accuracy, meaning it correctly matches two consecutive words. A high Rouge-L score indicates that the model effectively preserves sentence structure and word ordering, resulting in high sentence-level accuracy. Low Rouge metric values suggest that the model-generated text shows low similarity with the reference text, implying lower summarization or translation quality.

Findings and Discussion

Before natural language processing, the BBC News Summary was analyzed for words that occur frequently in the categories of business, entertainment, politics, sports and technology.

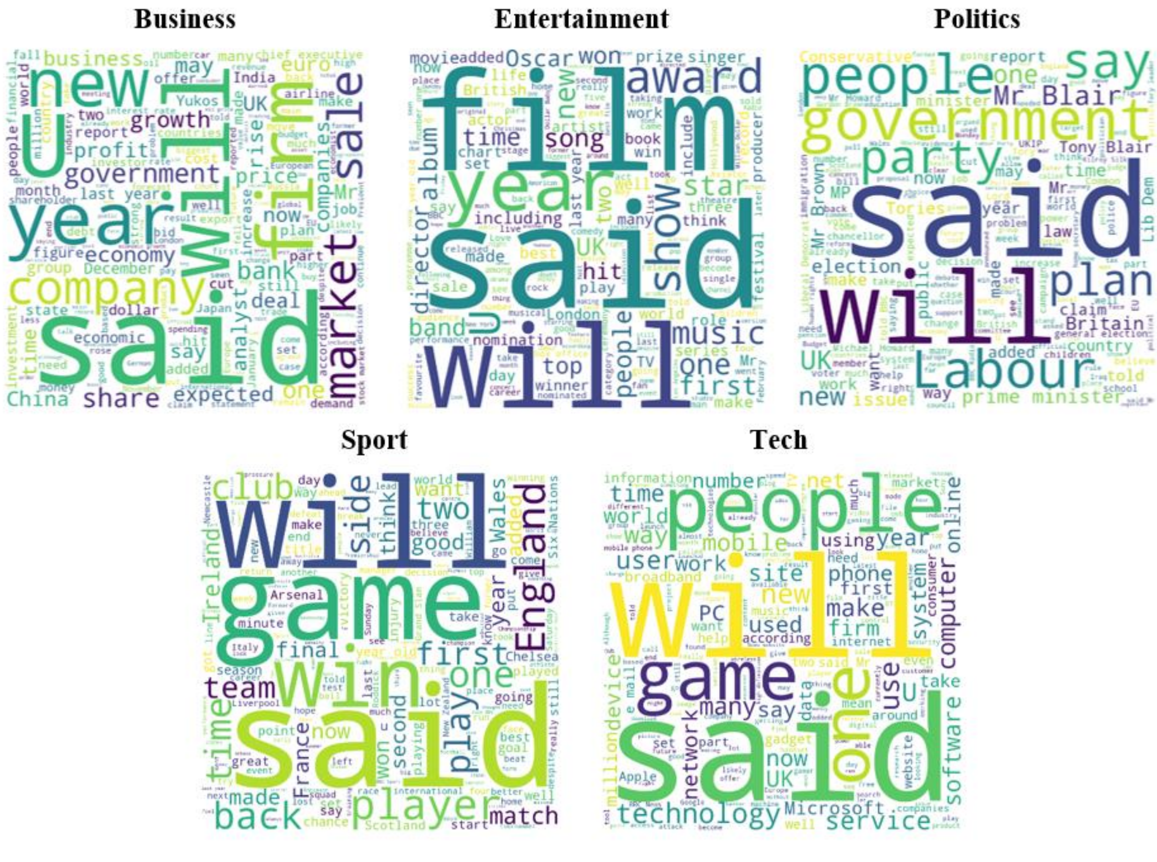

Figure 5. Most Frequent Words in Categories

In accordance with the working principles of machine learning methods, data preprocessing was performed, feature extraction was performed with N-Gram and vectorized with TF-IDF, and the BBC News Summary data was made 2-dimensional. Summaries of the news articles made suitable for use in the models were extracted with K-Means, Random Forest, Gradient Boosting Machines (GBM) / XGBoost, Logistic

Regression and Support Vector Machines (SVM) algorithms. ROUGE scores were calculated by comparing with the summaries in the dataset.

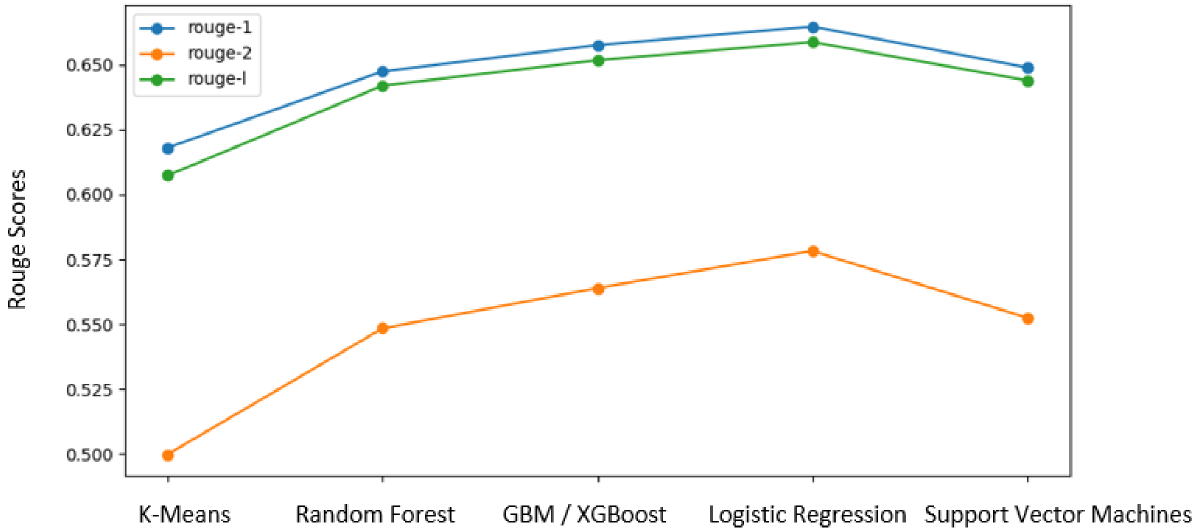

Figure 6. Comparison of Rouge Scores

Table 1. Evaluation Results of Summaries Extracted with Machine Learning Methods

Model	Average Rouge-1	Average Rouge-2	Average Rouge-L
K-Means	0.6180	0.4999	0.6073
Random Forest	0.6474	0.5484	0.6419
GBM / XGBoost	0.6575	0.5639	0.6517
Logistic Regression	**0.6647**	0.5782	0.6587
Support Vector Machines	0.6489	0.5525	0.6439

Upon examining the results in Figure 6 and Table 1, the K-Means model shows average performance in Rouge-1 and Rouge-L scores but exhibits the lowest performance in Rouge-2 scores. Rouge-1 and Rouge-2 scores reflect how models perform in terms of unigram and bigram matches, while Rouge-L score indicates their sensitivity to document length and structural similarity.

The Random Forest model performs better in Rouge scores compared to K-Means but falls behind GBM / XGBoost and Logistic Regression.

The GBM / XGBoost model demonstrates better performance than Random Forest in Rouge-1 and Rouge-L scores, although it performs lower than Logistic Regression. It achieves a good result in Rouge-2 scores compared to Random Forest.

The Logistic Regression model shows the highest performance across all Rouge scores and can be considered more effective compared to other models.

The Support Vector Machines model exhibits similar performance to Random Forest and GBM / XGBoost in Rouge-1 and Rouge-L scores but shows slightly lower performance in Rouge-2 scores.

Overall, all models have demonstrated some success in summarization, but each has its strengths and weaknesses. Logistic Regression, in particular, performs better in Rouge-2 scores, which measure the success of matching consecutive words compared to other models. It can be concluded that Logistic Regression may produce more consistent and meaningful sentences in summarization and translation tasks.

Conclusion and Discussion

In this study, extractive text summarization was performed using machine learning methods. Initially, the raw texts from the BBC News Summary dataset underwent necessary text preprocessing steps. During this phase, the texts were cleaned, tokenized, and irrelevant elements were removed. Subsequently, the n-gram method was used for feature extraction, and the texts were vectorized using TF-IDF.

Clustering was performed on the processed data using five different machine learning methods, and these were applied in the extractive text summarization process. For each model, the importance scores of the sentences were calculated, and summaries were generated based on these scores.

According to the study results, the Logistic Regression model stood out by showing the highest performance in Rouge-1, Rouge-2, and Rouge-L scores. These results indicate that the Logistic Regression model is superior to other models, especially in the Rouge-2 metric, which measures the success of matching consecutive words. This suggests that Logistic Regression may produce more consistent and meaningful sentences in summarization and translation tasks.

Other machine learning models, such as K-Means, Random Forest, GBM/XGBoost, and Support Vector Machines (SVM), were also evaluated for their text summarization performance. The GBM/XGBoost model showed better performance than the Random Forest model in Rouge-1 and Rouge-L scores but was less successful than the Logistic Regression model. The Random Forest model performed better than K-Means in Rouge scores but showed lower performance compared to other models. The Support Vector Machines model exhibited similar performance to other models in Rouge-1 and Rouge-L scores but showed lower performance in Rouge-2 scores.

The presented results evaluate the text summarization performance of different machine learning models. The K-Means, Random Forest, GBM/XGBoost, Logistic Regression, and Support Vector Machines methods were used to extract meaningful information from datasets and present this information in a summarized format through text processing, model training, and evaluation stages. The findings from these stages demonstrate that these models exhibit differences in various performance metrics.

Overall, this study provides valuable insights that could impact model selection in AI-based text processing applications. The Logistic Regression model emerges as a more effective option compared to other models for summarization and translation tasks and could be preferred in practical applications. These findings offer significant information that may influence model selection in AI-based text processing applications and could serve as a crucial foundation for developing new strategies and improving existing methods in the field of text summarization.

References

AKCA, M. F. (2020). Nedir Bu Destek Vektör Makineleri? (Makine Öğrenmesi Serisi-2) | by Mehmet Fatih AKCA | Deep Learning Türkiye | Medium. *Medium*.

Alpkoçak, A., Tocoglu, M. A., Çelikten, A., & Aygün, İ. (2019). Türkçe Metinlerde Duygu Analizi için Farklı Makine Öğrenmesi Yöntemlerinin Karşılaştırılması. *Deu Muhendislik Fakultesi Fen ve Muhendislik*, *21*(63), 719–725. https://doi.org/10.21205/deufmd.2019216303

Ayata, F., & Çavuş, H. (2022). Yüz Tanıma Sistemlerinde Kullanılan ESA, YGH-DVM ve DSA Algoritmalarının Performans Testleri. *Fırat Üniversitesi Fen Bilimleri Dergisi*, *34*(1), 39–48.

AYDIN, A., & UÇKAN, T. (2023). Çizgeler Üzerinde Farklı Ağırlıklandırma Yöntemleri Ve Merkezilik Ölçütleri İle Çıkarımsal Metin Özetleme. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, *35*(1), 71–79. https://doi.org/10.35234/fumbd.1155617

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chen, T., international, C. G.-P. of the 22nd acm sigkdd, & 2016, undefined. (2016). Xgboost: A scalable tree boosting system. *Dl.Acm.Org*, *13-17-Augu*, 785–794. https://doi.org/10.1145/2939672.2939785

Chiusano, F. (2022). Two minutes NLP — Learn the ROUGE metric by examples _ by Fabio Chiusano _ NLPlanet _ Medium.

Erhandı, B. (2020). Derin öğrenme ile metin özetleme, 34.

GÖRENTAŞ, M. B., & UÇKAN, T. (2023). Makine Öğrenmesi Yöntemleri Kullanılarak Mahkeme Kararlarlarının Kümelenmesi. *Computer Science*, (8), 148–158. https://doi.org/10.53070/bbd.1318518

GÖRENTAŞ, M. B., UÇKAN, T., & BAYRAM ARLI, N. (2023). Uyuşmazlık Mahkemesi Kararlarının Makine Öğrenmesi Yöntemleri ile Sınıflandırılması. *Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *28*(3), 947–961. https://doi.org/10.53433/yyufbed.1292275

Gupta, V., intelligence, G. L.-J. of emerging technologies in web, & 2010, undefined. (2010). A survey of text summarization extractive techniques. *CiteseerV Gupta, GS LehalJournal of Emerging Technologies in Web Intelligence, 2010•Citeseer*. https://doi.org/10.4304/jetwi.2.3.157-157

Khan, R., Qian, Y., Information, S. N.-I. J. of, & 2019, undefined. (2019). Extractive based text summarization using k-means and tf-idf. *Researchgate.NetR Khan, Y Qian, S NaeemInternational Journal of Information Engineering and Electronic Business, 2019•researchgate.Net*, *3*, 33–44. https://doi.org/10.5815/ijieeb.2019.03.05

Luhn, H. P. (2010). The Automatic Creation of Literature Abstracts. *IBM Journal of*

Research and Development, *2*(2), 159–165. https://doi.org/10.1147/rd.22.0159

Martin Katz, D., Bommarito, M. J., & Blackman, J. (2017). A general approach for predicting the behavior of the Supreme Court of the United States. *PLoS ONE*, *12*(4). https://doi.org/10.1371/JOURNAL.PONE.0174698

Mumcuoğlu, E., Öztürk, C., Ozaktas, H., & A. K.-I. P., & 2021, undefined. (2021). Natural language processing in law: Prediction of outcomes in the higher courts of Turkey. *Elsevier*, *58*(5), 306–4573. https://doi.org/10.1016/j.ipm.2021.102684

Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In *Mining Text Data* (Vol. 9781461432, pp. 43–76). https://doi.org/10.1007/978-1-4614-3223-4_3

Olmez, R. (2024). T5 modelini Kullanarak Geçmişten Günümüze Metin Özetleme | by Ramazan Olmez | Medium.

Özmutlu, H. C., & Çağlar, B. (2009). ARAMA MOTORLARINDA YENİ KONU TANILAMADA KARAKTER N-GRAM VE YAPAY SİNİR AĞLARI UYGULAMASI. *Acikerisim.Uludag.Edu.Tr*, *14*.

Schütze, H., Manning, C., & Raghavan, P. (2008). Introduction to information retrieval.

Sharif, P. (2018). BBC News Summary.

YAZĞILI, E., & BAYKARA, M. (2022). Türkçe metinlerde makine öğrenmesi yöntemleri ile siber zorbalık tespiti. *Gümüşhane Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *12*(2), 443–453. https://doi.org/10.17714/gumusfenbil.935448