Journal of Engineering Technology and Applied Sciences, 2023 *e-ISSN:* 2548-0391
Received 10 July 2023, Accepted 17 August 2023 *Vol. 8, No. 2, 87-105*
Published online 26 August 2023, Research Article *doi: 10.30931/jetas.1325483*

Citation: Varol Arısoy, M., Tunç Abubakar, T., "A Comparative Analysis of Ensemble Learning Methods on Social Media Account Detection". Journal of Engineering Technology and Applied Sciences 8 (2) 2023 : 87-105.

A COMPARATIVE ANALYSIS OF ENSEMBLE LEARNING METHODS ON SOCIAL MEDIA ACCOUNT DETECTION

Merve Varol Arisoy[a*], Tuğba Tunç Abubakar[b]

[a*]*Information Systems Engineering Department, Bucak Technology Faculty, Burdur Mehmet Akif Ersoy University, 15300, Bucak - Burdur, Turkiye (*corresponding author)*
mvarisoy@mehmetakif.edu.tr

[b]*Zeliha Tolunay High School Management Information Systems Department PhD Student, Burdur Mehmet Akif Ersoy University, 15300, Bucak - Burdur, Turkiye,*
tubatunc.2@gmail.com

Abstract

Today, social media platforms usage and benefiting rate from these environments are increasing. This rapid spread of social media has also allowed the emergence of fake accounts. Fake accounts are generally created to implement malicious activities through another user account or to spread incorrect information. To prevent the detriment that this situation may cause to real individuals, an effective fake account detection was carried out by using ensemble learning methods (Bagging, Boosting, Stacking, Voting and Blending) in this study. These methods were combined with various machine learning algorithms to measure their effectiveness in detecting fake accounts. The experimental results suggested that Bagging technique attained an accuracy level of 90.441%, Stacking technique 89.706%, Voting technique 88.971% and the Blending technique attained 88.235% in the test phase. While for the Boosting methods, XGboost technique attained accuracy level of 86.765%, whereas the AdaBoost outperformed it with an accuracy level of 91.912% in the test phase. The extant results demonstrates that ensemble learning methods combined with machine learning algorithms are efficient in detecting fake social media accounts. It is considered that additional studies with larger datasets alongside the usage of different ensemble methods can further improve the accuracy of the detection process.

Keywords: Ensemble learning, social media fake account detection, bagging, blending, boosting, stacking, voting

1. Introduction

The emergence of the concept of social media in recent years and the increase in popularity over time have made these platforms a part of daily life. People actively use social media platforms for various purposes such as information sharing, interaction, entertainment, and

commerce. However, the rapid growth and increasing the popularity of social media have also led to the emergence of fake accounts. Fake accounts can engage in malicious activities or spread incorrect information by deceiving real users. Therefore, the detection and prevention of fake accounts have become significant challenges for social media platforms. In recent years, numerous methods and techniques have been developed for fake account detection. Among these techniques, approaches based on machine learning and data mining have gained significant attention. Machine learning possesses the ability to recognize patterns in complex datasets and offers a potential solution for detecting fake accounts [1].

Several methods for detecting fake accounts have been proposed and employed by previous studies. Some studies have used machine learning algorithms to classify accounts based on specific features using crowdsourcing [3], which relies on human effort, or utilizing a graph-based approach [4]. Erşahin et al. [5] employed the Naive Bayes technique to detect fake accounts using a Twitter dataset. Adewole et al. [6] utilized from multilayer perceptron (MLP), support vector machine (SVM) and random forest classification algorithms for account detection. Gayathri et al. [7] applied varied classification algorithms such as SVM, random forest, and deep neural networks. Mulamba et al. [8] proposed a robust Sybil detection framework based on graph-based structural properties of an Online Social Networks (OSN). Stein et al. [9] developed a novel model called the Facebook Immune System using the Facebook dataset, incorporating random forest, SVM, and boosting techniques. Abokhodair et al. [10] employed a unsupervised machine learning method to detect bot accounts. Cao et al. [11] applied a system based on Sybil rank graph for detection.

The current research adopted ensemble method to detect fake accounts on social media. Ensemble learning is a approach where several machine learning algorithms are combined and used for assessing the methods ability to reach a collective decision and which method outperforms the others. The ensemble learning methodology has been the preferred approach as it offers the potential to achieve higher accuracy rates by combining learning models trained with varied machine learning algorithms [2].

The remainder of the article is organized as follows. First of all, commonly used machine learning techniques for fake account detection and the advantages of ensemble systems are discussed. Then, the proposed method for fake account detection based on an ensemble system is presented, and the results of the evaluation conducted with the dataset are shared to demonstrate the effectiveness of the method.

This study is a significant step towards developing effective and reliable solutions for fake account detection that can be integrated into social media platforms.

2. Literature review

Related studies have used various methods to detect social media accounts that are fake in nature. Erşahin et al. [5] achieved an accuracy of 90.9% in detecting fake accounts by using Naive Bayes algorithm and preprocessed Twitter data, alongside supervised discretization technique called entropy minimization discretization (EMD).

The study in [6] investigated the identification of Twitter spam accounts and developed an initial detection of spammer classes by combining Principal Component Analysis (PCA) and k-means algorithms. In the aforementioned study, several features for spam detection in social networks were adopted and for improving the performance new features were contributed.

MLP, SVM, and Random Forest classification algorithms were utilized. The best results were obtained using the Random Forest algorithm, achieving an accuracy of 96.30%.

The study in [7] applied a machine learning workflow for OSN to identify fake accounts. Instead of predicting each individual account, they classified fake account groups thereby determining whether they were created by the same person. To this end, various classification algorithms, including SVM, Random Forest, and Deep Neural Network were recommended by them.

The work in [8], Sybilradar framework employed Bayesian inference and Monte Carlo sampling techniques based on probability-based threshold values. In [9], utilizing the Facebook dataset, Stein et. al. introduced a new model called the Facebook Immune System which combined techniques such as Random Forest, Support Vector Machine, and boosting. Additionally, it employed feature cycling as the selection technique.

Facebook bots utilize an algorithm based on the number of friends to detect fake accounts. They examine the tagging of friends or analyze relationship histories. While Facebook bots can identify fake accounts, it is not successful enough in detecting fake accounts created by humans. Unsupervised machine learning has been employed to detect bots without relying on human supervision [10]. In this technique, instead of tagging, information is aggregated based on proximity. Through grouping functions, bots can be successfully recognized as they exhibit similar characteristics.

Sybil rank, designed in 2012 [11], is built upon a graph-based system. It ranks profiles based on interactions, tags, and wall posts. Profiles with high ranks are labeled as genuine, while those with low ranks are considered as fake. However, the reliability of this method is low as a real profile with limited interactions could end up with a low rank.

Akyon and Kalfaoglu [12] introduced the problem of detecting fake Instagram accounts as a binary classification problem and proposed a cost-sensitive technique to address this problem. This technique is based on a genetic algorithm foundation for automated classification. They employed the Synthetic Minority Over-sampling Technique for Nominal and Continuous (SMOTE-NC) algorithm for synthetic minority sample augmentation in the dataset. As a result, by utilizing support vector machine and neural network-based techniques, they achieved an F1 score of 86% for detecting robotic accounts, reaching a certain level of success. However, the best neural network achieved an F1 score of 95%.

3. Material and methods

In this study, the dataset titled ‘fake_account__data_dict’ available on the Kaggle platform was used [16]. The dataset consists of a total of 696 records and includes 12 different attributes, including the target attribute. The target attribute represents a value of 0 for a real social media account and 1 for a fake profile. The attributes present in the dataset are presented in Table 1.

Before the training of the model, the relevance of the features to the target variable should be calculated. This step is necessary to determine the input features that most affect the correct classification. In this way, the value of a variable can be estimated over other variables that are related. Therefore, Pearson correlation analysis has been conducted among the features. In Figure 1 and Figure 2, the correlation before feature selection and after feature selection is given, respectively.

Table 1. Features of the dataset.

Features	Description of Features
profile_pic	Whether the account has a profile picture (1) or not (0)
ratio_numlen_username	Ratio of numeric characters in the account username to its length
len_fullname	Number of characters in the user's full name
ratio_numlen_fullname	Ratio of numeric characters in the user's full name to its length
sim_name_username	Whether the user's name matches their username completely (1),not at all (0).
len_desc	Number of characters in the account description
extern_url	Whether the account description includes a URL (1) or not (2)
private	Whether the user's posts are visible only to their followers (1) or to all Instagram users (2).
num_posts	Number of posts in the user account
num_followers	Number of Instagram users following the account
num_following	Number of Instagram users followed by the account
fake	Whether the user account is real (0) or a fake account (1).

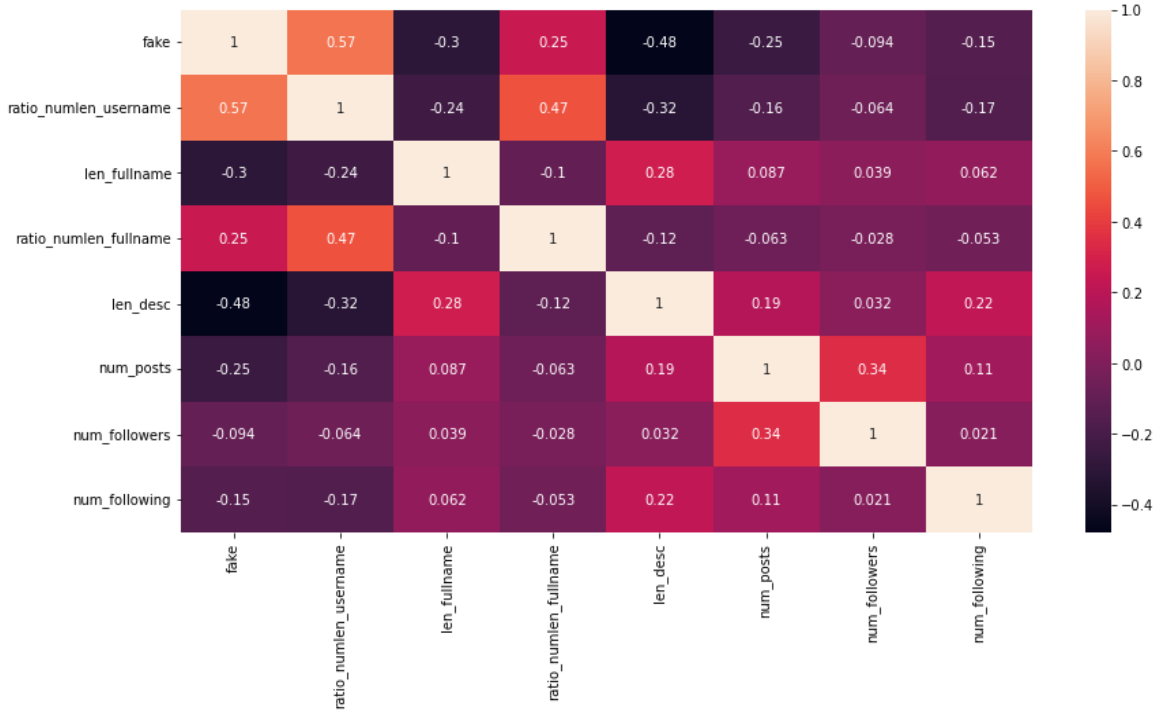

Figure 1. The correlation (pearson correlation) before feature selection

The correlation of some input features with the target variable remained below the threshold value (0.2). For this reason, considering that the contribution of the model to the prediction performance will be limited, the input features below the threshold value have been eliminated. The remaining input properties after elimination are given in Table 2. In addition, another correlation analysis was performed between the feature set in Table 2 and the results are presented in Figure 2. The first 4 features in Table 2 represent the input variables, and the fake feature represents the target variable.

Table 2. The attributes of the dataset after feature selection.

Features
ratio_numlen_username
len_fullname
len_desc
num_posts
fake

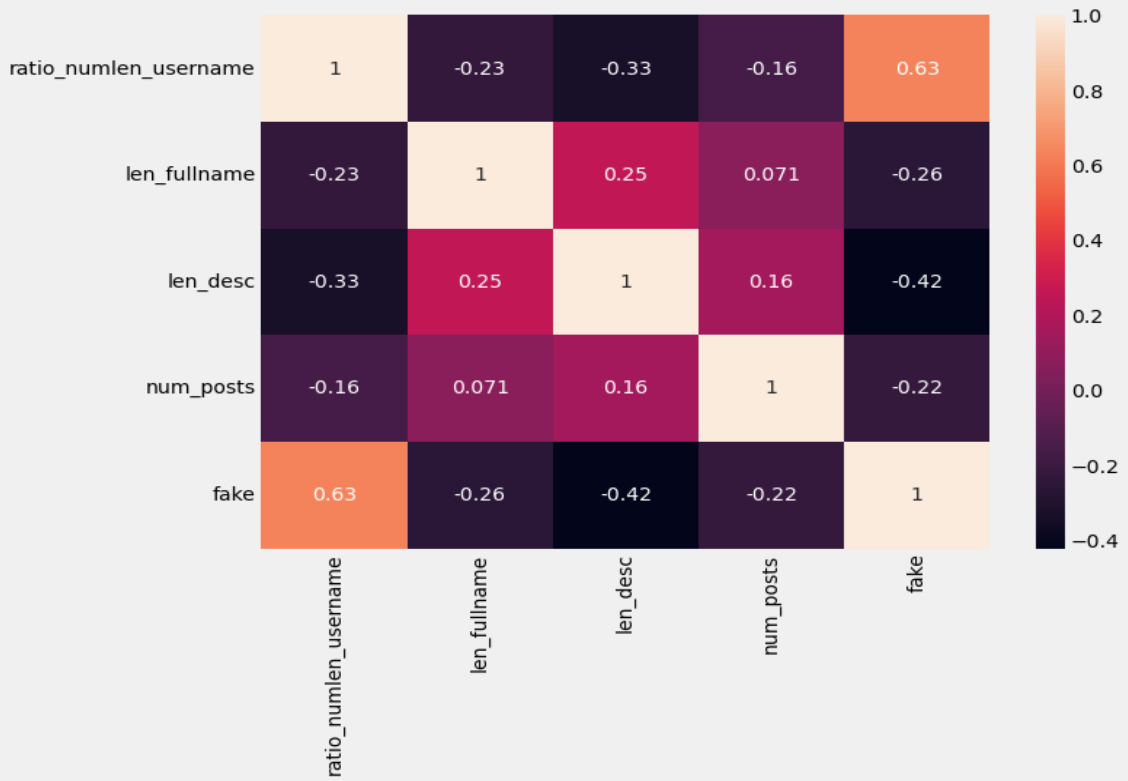

Figure 2. The correlation (pearson correlation) after feature selection

3.1 Data cleaning

In the data processing phase, it was initially checked whether there was any missing data, and it was determined that there were no missing values. Then, the presence of duplicate values was examined, and those values were removed. The categorical values in the dataset were converted into numerical representations.

Before feature extraction, the distribution was balanced with an equal number of fake and real instances, 50% fake and 50% real. This situation is shown in Figure 3.

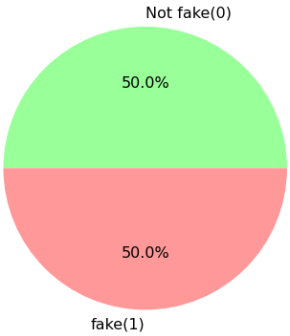

Figure 3. Before feature extraction, the distribution of fake and real instances

3.2 EDA(Univariate-Bivariate)

Univariate analysis, which is a step in Exploratory Data Analysis (EDA), focuses on understanding the characteristics and distribution of a single variable. This analysis is used to uncover the basic statistics, visualizations, and patterns of the dataset. When Figure 4 is examined, it is seen that the "len_desc" and "num_posts" properties have many outliers indicated by black dots. Since the data does not have a normal distribution, it may adversely affect the decision of the model, so it is necessary to make the data follow a normal distribution as much as possible. Therefore, log transform method, which is a feature transformation method, is applied to have a more normal distribution of "len_desc", "len_fullname" and "num_posts" properties. The results of this transformation are given in Figure 5.

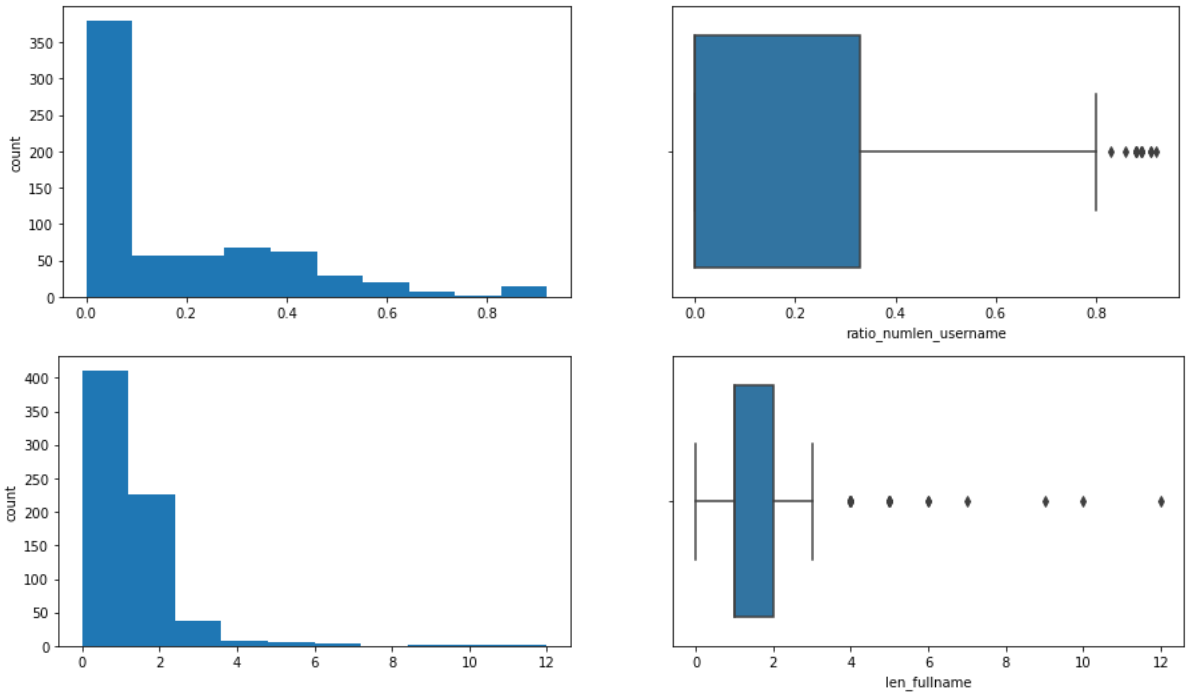

Figure 4. EDA univariate analysis

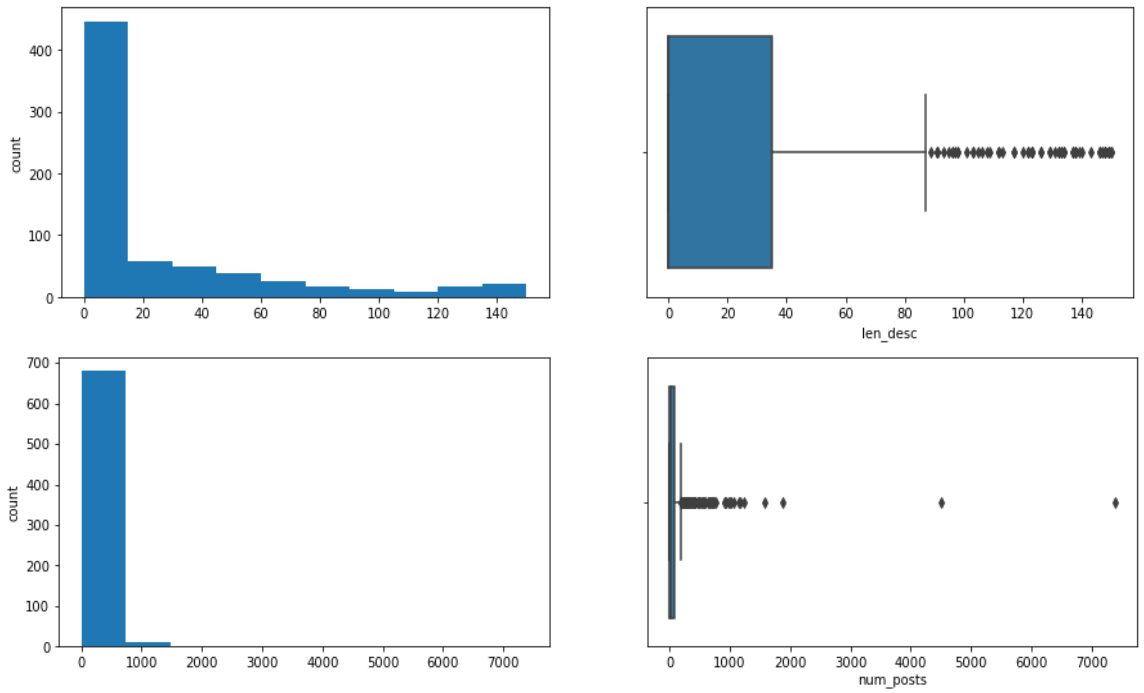

Figure 4. (Continued) EDA univariate analysis

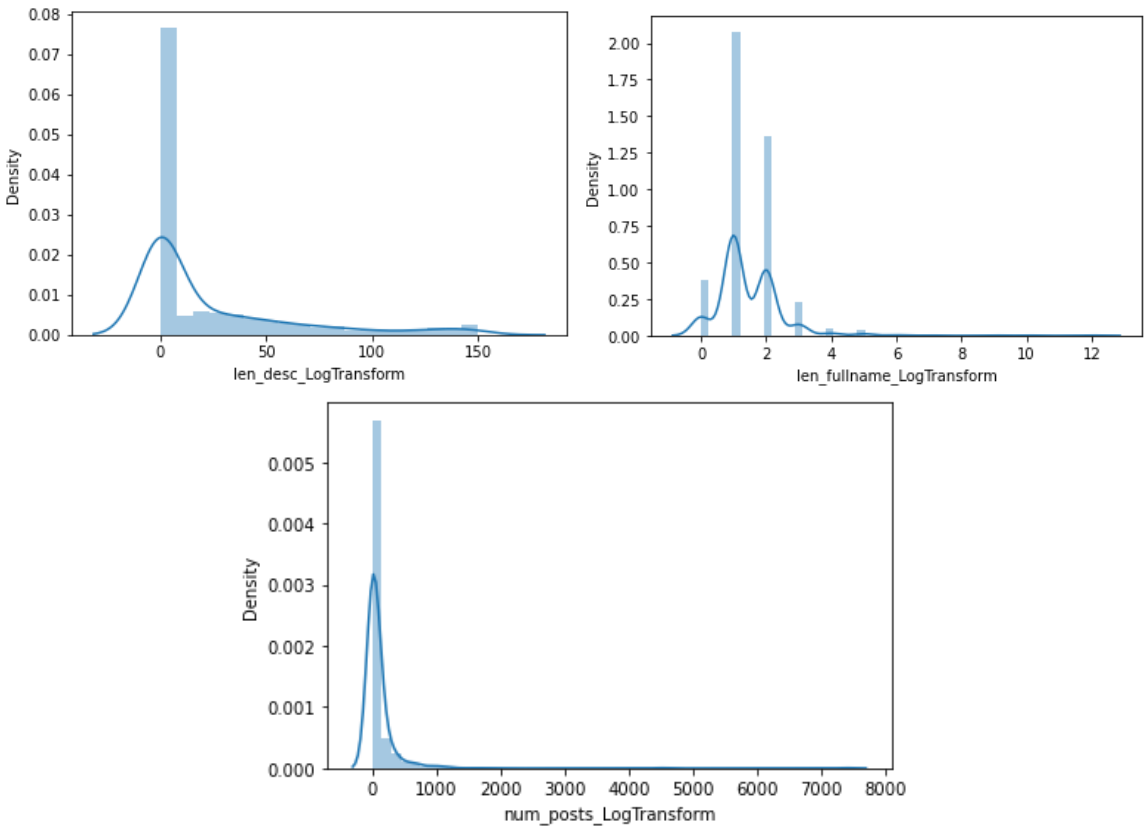

Figure 5. Distribution of “len_desc”, “len_fullname” and “num_posts” properties as a result of applying log transform

Another important step in EDA is bivariate analysis. Bivariate analysis is used to understand the relationship between two variables. This analysis is employed to determine the relationship

between variables in the dataset, examine dependencies, and uncover relational patterns. When Figure 6 is examined, it is seen that the “len_desc”, “len_fullname” and “num_posts” features of fake accounts are less than real accounts.

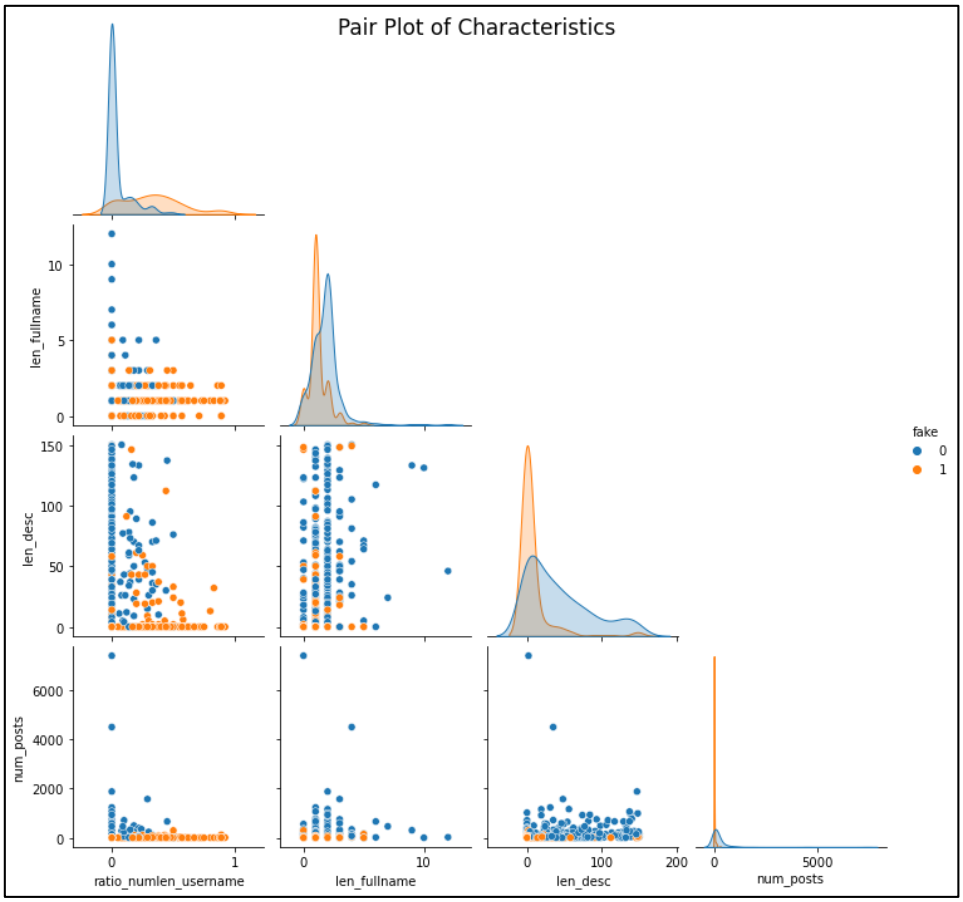

Figure 6. EDA bivariate analysis

3.3 Feature extraction and selection

The following steps in Figure 7 were followed in the feature extraction and selection phase:

Figure 7. Feature processing steps

3.3.1 Feature selection with pearson correlation

First, feature selection was performed using the Pearson correlation method. Initially, there were 12 features, and this method was used to determine the optimal subset of features. As a result, the 12 features were reduced to 5 features. These 5 features consisted of 4 input features and 1 target feature.

3.3.2 RFECV feature selection

After the Pearson correlation, the Recursive Feature Elimination with Cross-Validation (RFECV) method was applied as a second feature selection method. This method is used to determine the most important features by successively eliminating features and evaluating the model's performance. It was determined that all 4 input features selected by Pearson (“ratio_numlen_username”, “len_fullname”, “len_desc”, “num_posts”) were significant. Therefore, no further feature extraction was performed at this stage. As a result, a total of 4 input features and 1 target feature were used in model training.
With these steps, the aim was to determine the most suitable subset of features and improve the performance of the model. Feature selection, which started with the Pearson correlation and then supported by the RFECV method, helped the model make better generalizations and obtain more effective results.

3.3.3 Smote over sampling

SMOTE is a sampling method used to address class imbalance in datasets. In this sampling method, it is aimed to balance the class distribution by producing synthetic samples for the minority class.

After feature extraction, the balance of the 'fake' feature representing the target variable was re-evaluated and it was seen that there was a proportional imbalance between the real and fake class. In Figure 8 this situation is illustrated. To address this, the SMOTE over-sampling method was employed to equalize the ratios of real and fake values.

With the SMOTE technique, synthetic instances were generated for the minority class (fake=1) by producing interpolated examples. This ensured an equal ratio of real and fake values and resulted in a balanced distribution between the two classes. Figure 8 demonstrates the distribution before applying SMOTE Over Sampling (after feature selection).

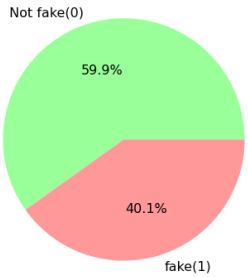

Figure 8. Imbalanced data the distribution before smote over sampling

3.4 Data scaling (Normalization)

Data scaling (normalization) is a preprocessing step used in machine learning to transform the numerical features of a dataset into a consistent range. Through normalization, potential issues that may arise from differences in the scales of features is mitigated. The general formula is in Equitation 1:

$$V' = \frac{v - min_a}{max_a - min_a}(new\_max_a - new\_min_a) + new\_min_a \quad [2](1)$$

whereas v= is an original value and v’=is the normalized value. The data was divided into 80% for training purposes and 20% for testing purposes, and the Standard Scaler was applied.

3.5 Ensemble methods

Ensemble learning methods is utilized to create a stronger model by leveraging the diversity and independence of the underlying machine learning algorithms. In this section, firstly, information about all machine learning algorithms used in ensemble methods are given, and then the ensemble methods used are explained.

3.5.1 Machine learning algorithms used in ensemble methods

Decision Tree: This algorithm creates a decision tree by using the features and target classes in the training dataset. It can handle non-linear relationships and provide effective results in classification problems.

K-Nearest Neighbors (KNN): is a machine learning algorithm used for classification or regression tasks. It predicts the class of a new instance by examining the class labels of its nearest neighbors.

Logistic Regression: is a linear regression technique used to differentiate between two or more classes. The logistic function, also known as the sigmoid function, is used to transform input values into a value between 0 and 1. These values represent the probability distribution of the classes.

SVM: is a machine learning algorithm used for classification and regression problems. It is an pattern recognition method used to classify a dataset or predict an output value.

Gaussian Naive Bayes (Gaussian NB): is a machine learning algorithm used for classification problems. The basis of this algorithm is that the distribution of features for each class follows a normal distribution. Bayes theorem is utilized to estimate the probability of a given class for the feature values in the dataset.

Extreme Gradient Boosting: is a high-performance classification algorithm on large and complex data sets. It has a strong forecasting capability thanks to its optimized tree structures and regularized error functions.

AdaBoost Classifier: this machine learning algorithm solves classification problems by bringing together different types of weak learners (usually decision trees). In this way, it can better manage the complexity of the dataset and achieve an overall high classification performance.

Gradient Boosting Classification: is used for classification problems. Gradient Boosting aims to create a strong classification model by combining a series of weak learners.

3.5.2 Bagging

Bagging (Bootstrap Aggregating) is a technique that involves generating new data sets by randomly sampling with replacement from the original data set. As seen in figure below, independent learners are trained and then each of the weak models make prediction. The resulted predictions are combined to determine the final aggregate as outcome [13]. In this study, after applying Bagging with the Decision Tree algorithm, a test accuracy of 90.441% was achieved. Figure 9 shows the Bagging process.

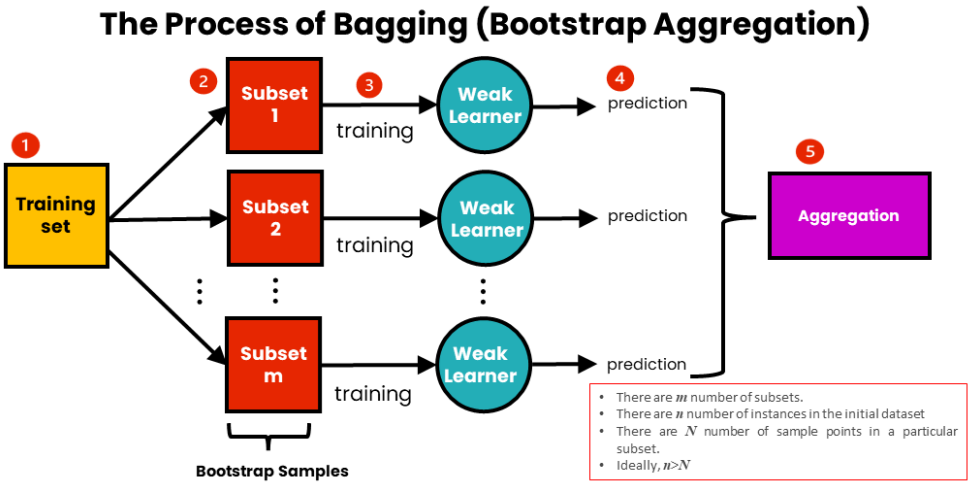

Figure 9. Bagging process [13]

3.5.3 Blending

Blending is an ensemble method that combines the predictions of different machine learning models trained on a single dataset to create a stronger prediction strategy. In contrast to other ensemble techniques, blending utilizes two layers of machine learning algorithms: the first layer constitutes of base models and the latter, the meta models [15]. Figure 10 shows the Blending process.

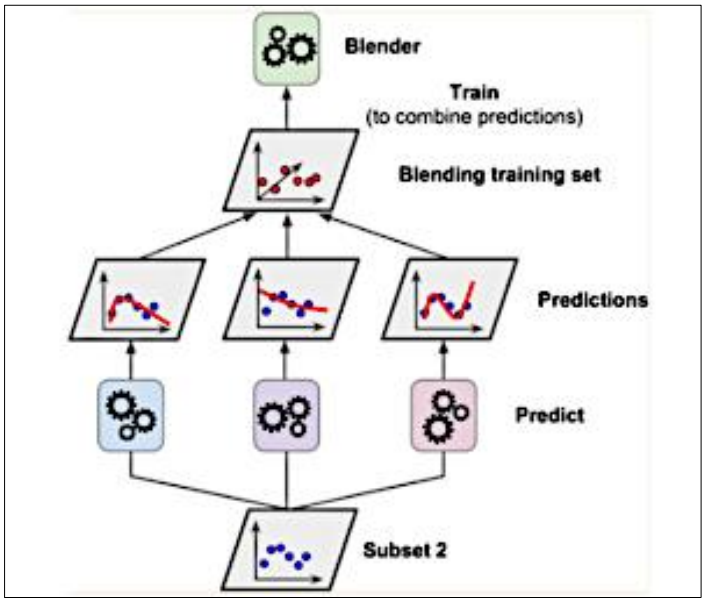

Figure 10. Blending process [15]

In this case, the Decision Tree Classifier, KNN, and Logistic Regression algorithms were used for blending. The predictions of these models were combined, and after applying blending, a test accuracy of 88.235% was achieved. By leveraging the strengths of different algorithms and combining their predictions, blending aims to improve the overall performance of the model.

3.5.4 Stacking

Stacking aims to create a more generalized and high-performance model by combining the different features and strengths of multiple heterogeneous models to make prediction [13]. Stacking is considered more robust compared to bagging and boosting as it is based on strong learners, heterogeneous models and also operationalized using meta models [13]. Figure 11 shows the Stacking process.

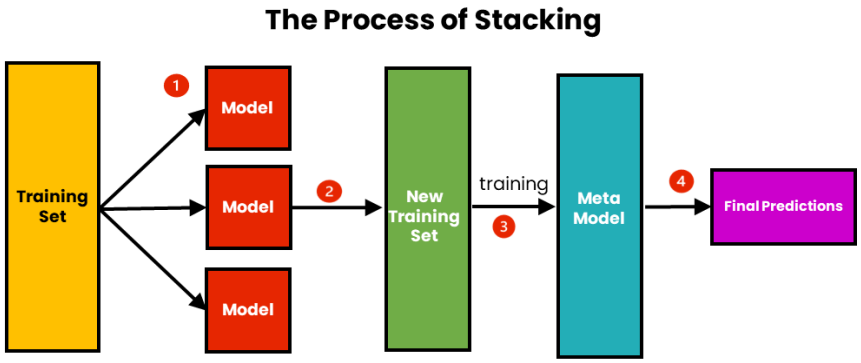

Figure 11. Stacking process [13]

During the implementation of the stacking method in the study, Logistic Regression, KNN, Decision Tree Classifier, SVM, Gaussian NB algorithms were used and a Stacking test accuracy rate of 89.706% was obtained.

3.5.5 Boosting

Boosting is designed to produce a model with minimum bias compared to those of separate individual models. As in bagging, the weak learners are homogeneous and in this method, weak learners are trained one after another. In this approach, each subsequent learner aims to enhance the errors made by the previous learners in the sequence [13]. The boosting method can achieve high accuracy rates and good generalization performance. Figure 12 shows the Boosting process.

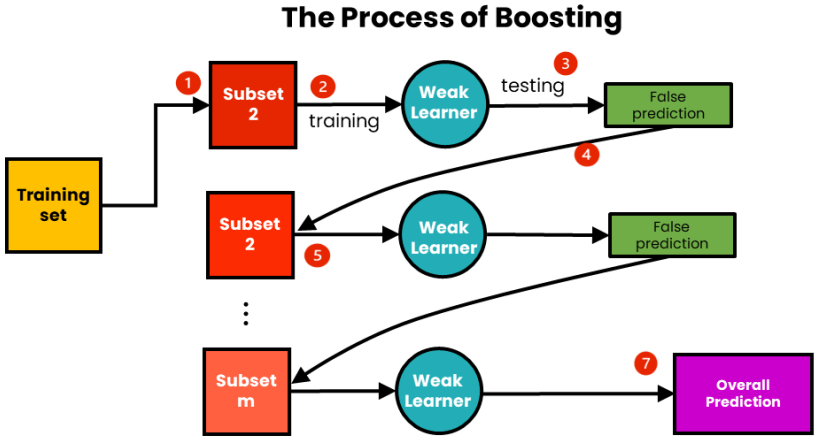

Figure 12. Boosting process [13]

For the Boosting process, XGB Classifier, AdaBoost Classifier, and Gradient Boosting Classifier algorithms were used, resulting in a XGboost test accuracy of 86.765%. Among these algorithms, AdaBoost Classifier and Gradient Boosting achieved the highest test accuracy of 91.912%.

3.5.6 Voting

As stated earlier, bagging method algorithms are trained using multiple datasets that have been bootstrapped. However, voting method algorithms are trained using an identical dataset and voting makes predictions by combining multiple models [14]. In this method, there are two approaches: hard and soft voting. Hard voting is similar to a majority vote, while soft voting involves averaging the outputs of multiple algorithms [14]. The Hard voting type was used in this study. Figure 13 shows the Voting process.

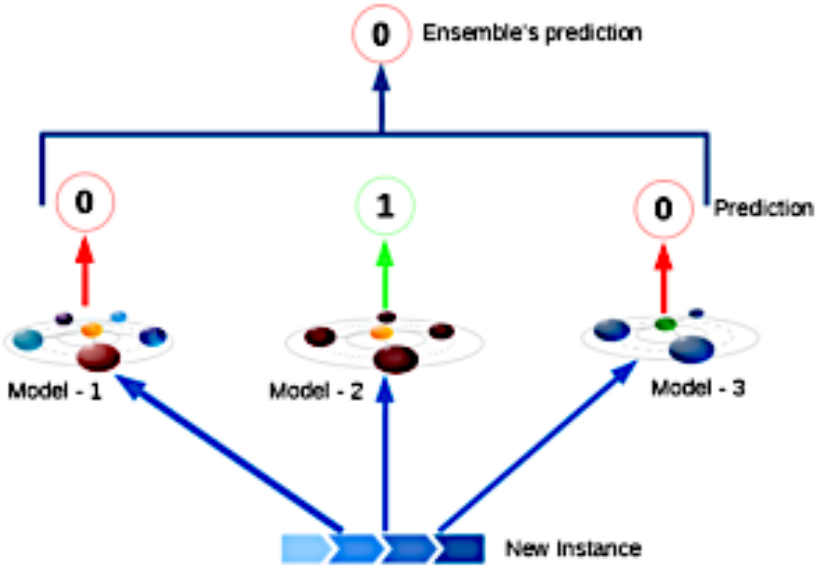

Figure 13. Voting process [14]

In this study, for the voting process, Logistic Regression, XGB Classifier, Random Forest Classifier, KNN, SVM algorithms were used, resulting in a Voting test accuracy of 88.971%.

3.6 Evaluation metrics

In the study, accuracy, F-score, precision and recall metrics were used to measure the predictive performance of the classifiers. Although the class distributions used in this study are balanced, the Recall, Precision and F-Score metrics are also used in addition to the use of the accuracy metric, as the accuracy metric can be misleading in evaluating an ensemble model with unbalanced class distributions [17]. Also, a confusion matrix is used to evaluate false detection rates of the ensemble models. Table 3 explains the confusion matrix in more detail.

Accuracy, is the ratio of the number of correct predictions to the total number of predictions. The accuracy metric is shown in equation 2.

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad (2)$$

Table 3. Confusion matrix

		Predicted Class	
		Positive	**Negative**
		TP	**FN**
Actual Class	**Positive**	The predicted value is positive and its positive in actual	The predicted value is negative and its positive in actual
	Negative	**FP**	**TN**
		The predicted value is positive and its negative in actual	The predicted value is negative and its negative in actual

Precision, is the proportion of accurate positive predictions to the total number of positive predictions. The use of the precision metric is given in Equation 3.

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

Recall, is the ratio of accurate positive predictions to the total number of positive observations. The use of the recall metric is given in Equation 4.

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

F-score, is the measure of model efficiency, a weighted average of model precision and recall. The use of the F-score metric is given in Equation 5.

$$F - score = 2x\frac{PrecisionxRecall}{Precision+Recall} \tag{5}$$

4. Results and discussion

Social media has become an important component of our lives. Therefore, detecting fake accounts on social media platforms is vital to sustaining integrity, safety, and trustworthiness of online communities. Also, such detection enables to preclude the misinformation situations. This study has performed a comparative analysis of ensemble learning methods to detect fake social media accounts. For this purpose, sub-methods of the ensemble learning method such as Bagging, Boosting, Stacking, Voting and Blending were used together with various machine learning algorithms and the success rate of each sub-method of the ensemble learning method was evaluated. According to this, the Bagging method had a test accuracy rate of 90.441%, the stacking method yielded an accuracy rate of 89.706%. The Voting and Blending method gained an accuracy rate of 88.971% and 88.235% respectively. As for the Boosting methods, the AdaBoost obtained a success rate of 91.912%, which denotes its superiority. On the other hand, XGboost achieved a success of 86.765%. These results prove that ensemble learning methods achieve a high level of performance in distinguishing between real and fake accounts.

The experiments illustrated that AdaBoost outperforms other methods by having the highest test accuracy rate. In addition, Bagging, Blending, Voting, and Stacking methods also produced

accomplished consequences. The results of ensemble learning methods in all the metrics such as accuracy, precision, recall and F1 scores are given in figures 14 to 20.

	precision	recall	f1-score	support
0	0.93	0.89	0.91	76
1	0.87	0.92	0.89	60
accuracy			0.90	136
macro avg	0.90	0.91	0.90	136
weighted avg	0.91	0.90	0.90	136

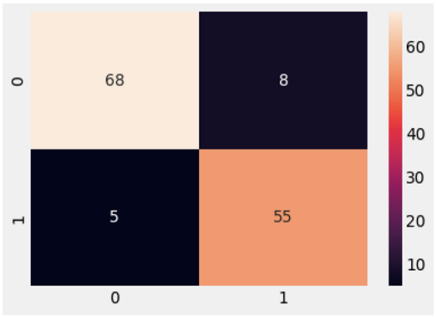

Figure 14. Results of bagging

When the results in Figure 14 are examined, it gives 68 TP, 8 FN, 5 FP, 55 TN. The success rate is also 90%.

	precision	recall	f1-score	support
0	0.91	0.88	0.89	76
1	0.85	0.88	0.87	60
accuracy			0.88	136
macro avg	0.88	0.88	0.88	136
weighted avg	0.88	0.88	0.88	136

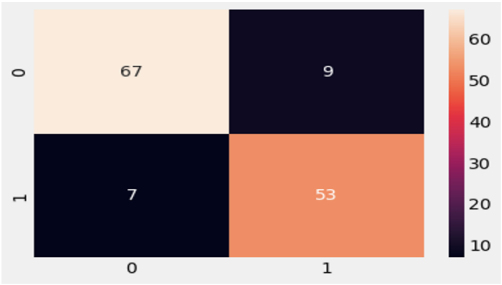

Figure 15. Results of blending

When the results in Figure 15 are examined, it gives 67 TP, 9 FN, 7 FP, 53 TN. The success rate is also 88%.

	precision	recall	f1-score	support
0	0.91	0.91	0.91	76
1	0.88	0.88	0.88	60
accuracy			0.90	136
macro avg	0.90	0.90	0.90	136
weighted avg	0.90	0.90	0.90	136

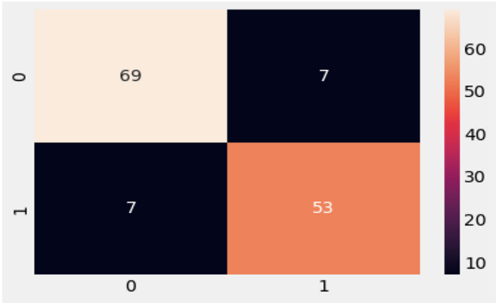

Figure 16. Results of stacking

When the results in Figure 16 are examined, it gives 69 TP, 7 FN, 7 FP, 53 TN. The success rate is also 90%.

	precision	recall	f1-score	support
0	0.90	0.86	0.88	76
1	0.83	0.88	0.85	60
accuracy			0.87	136
macro avg	0.87	0.87	0.87	136
weighted avg	0.87	0.87	0.87	136

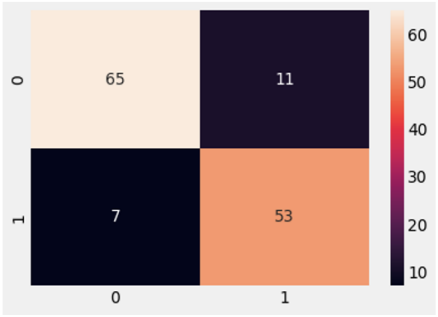

Figure 17. Results of boosting for XGboost

When the results in Figure 17 are examined, it gives 65 TP, 11 FN, 7 FP, 53 TN. The success rate is also 87%.

	precision	recall	f1-score	support
0	0.92	0.93	0.93	76
1	0.92	0.90	0.91	60
accuracy			0.92	136
macro avg	0.92	0.92	0.92	136
weighted avg	0.92	0.92	0.92	136

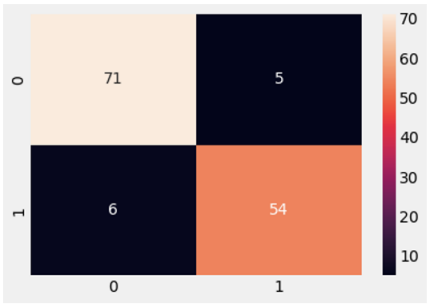

Figure 18. Results of boosting for AdaBoost classifier

When the results in Figure 18 are examined, it gives 71 TP, 5 FN, 6 FP, 54 TN. The success rate is also 92%.

	precision	recall	f1-score	support
0	0.95	0.91	0.93	76
1	0.89	0.93	0.91	60
accuracy			0.92	136
macro avg	0.92	0.92	0.92	136
weighted avg	0.92	0.92	0.92	136

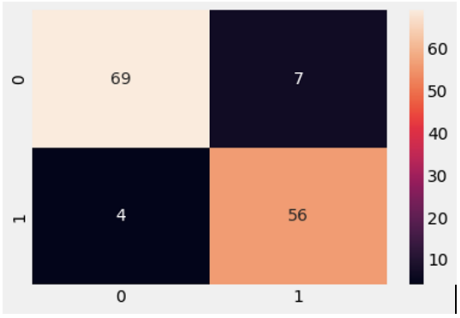

Figure 19. Results of boosting for Gradient Boosting

When the results in Figure 19 are examined, it gives 69 TP, 7 FN, 4 FP, 56 TN. The success rate is also 92%.

	precision	recall	f1-score	support
0	0.91	0.89	0.90	76
1	0.87	0.88	0.88	60
accuracy			0.89	136
macro avg	0.89	0.89	0.89	136
weighted avg	0.89	0.89	0.89	136

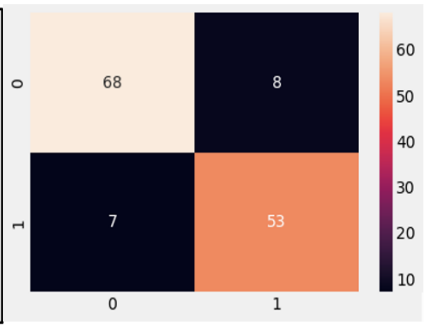

Figure 20. Results of Voting

When the results in Figure 20 are examined, it gives 68 TP, 8 FN, 7 FP, 53 TN. The success rate is also 89%. In Figure 21, the accuracy rate obtained from the test data of all the methods used in the study is given graphically. Accordingly, the ensemble method with the highest accuracy rate is Boosting.

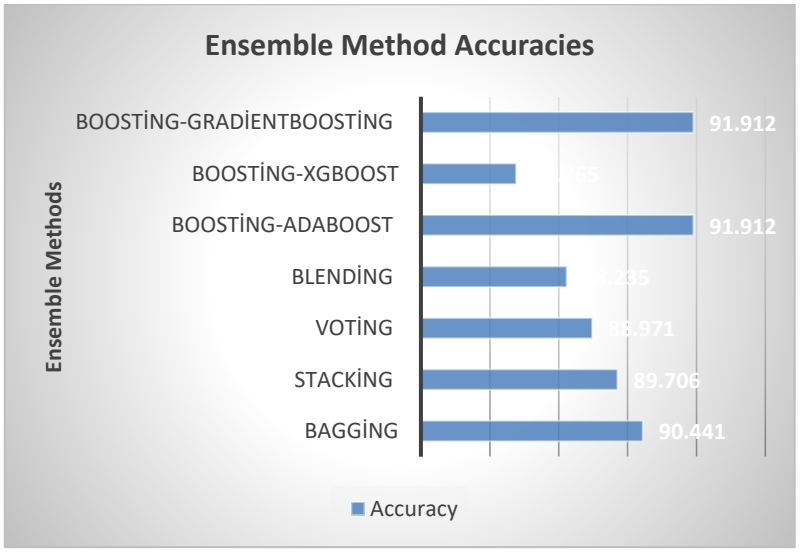

Figure 21. The success rates of each ensemble method

5. Conclusions

In this study, the process of accurately detecting fake social media accounts was carried out by using ensemble learning methods such as Bagging, Boosting, Stacking, Voting and Blending. The results indicated that AdaBoost and Gradient Boosting had the highest test accuracy rate of 91.912%, followed by Bagging and Stacking with the rate of 90.441 and 89.706, respectively. Based on these findings, it was concluded that the integration of ensemble learning methods with various machine learning algorithms, makes reliable decisions in identifying fake social media accounts. This study offers valuable guidance for future research and practical applications designed to detect fake social media accounts. As a future work, it is planned to apply the ensemble learning on a larger dataset and also adopt deep learning methods to gauge for better performance.

References

[1] Van Der Walt, E., Eloff, J., "Using machine learning to detect fake identities: bots vs humans", IEEE Access 6 (2018) : 6540-6549.

[2] Ali, A. K., Abdullah, A. M., "Fake accounts detection on social media using stack ensemble system", International Journal of Electrical and Computer Engineering (IJECE) 12(3) (2022) : 3013-3022.

[3] Al-Qurishi, M., Al-Rakhami, M., Alamri, A., Alrubaian, M., Rahman, S. M. M., Hossain, M. S., "Sybil defense techniques in online social networks: a survey", IEEE Access 5 (2017) : 1200-1219.

[4] Wang, G., Wilson, C., Zhao, X., Zhu, Y., Mohanlal, M., Zheng, H., Zhao, B. Y., "Serf and turf: crowdturfing for fun and profit", In Proceedings of the 21st international conference on World Wide Web (2012) : 679-688.

[5] Erşahin, B., Aktaş, Ö., Kılınç, D., Akyol, C., "Twitter fake account detection", In 2017 International Conference on Computer Science and Engineering (UBMK) (2017): 388-392.

[6] Adewole, K. S., Han, T., Wu, W., Song, H., Sangaiah, A. K., "Twitter spam account detection based on clustering and classification methods", The Journal of Supercomputing 76 (2020) : 4802-4837.

[7] Gayathri, A., Radhika, S., Jayalakshmi, S. L., "Detecting fake accounts in media application using machine learning", International Journal of Advanced Networking and Applications (2019) : 234-237.

[8] Mulamba, D., Ray, I., Ray, I., "Sybilradar: A graph-structure based framework for sybil detection in on-line social networks", In ICT Systems Security and Privacy Protection: 31st IFIP TC 11 International Conference 31 (2016) : 179-193.

[9] Stein, T., Chen, E., Mangla, K., "Facebook immune system", In Proceedings of the 4th workshop on social network systems (2011) : 1-8.

[10] Abokhodair, N., Yoo, D., McDonald, D. W., "Dissecting a social botnet: Growth, content and influence in Twitter", In Proceedings of the 18th ACM conference on computer supported cooperative work & social computing (2015) : 839-851.

[11] Cao, Q., Sirivianos, M., Yang, X., Pregueiro, T., "Aiding the detection of fake accounts in large scale social online services", In 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12) (2012) : 197-210.

[12] Akyon, F. C., Kalfaoglu, M. E., "Instagram fake and automated account detection", In 2019 Innovations in intelligent systems and applications conference (ASYU) (2019) : 1-7.

[13] Kalirane, M., "Ensemble Learning Methods: Bagging, Boosting and Stacking", Available from: https://www.analyticsvidhya.com/blog/2023/01/ensemble-learning-methods-bagging-boosting-and-stacking/ (Accessed June, 2023).

[14] Kim, C., "Ensemble Learning - Voting and Bagging with Python", Available from: https://medium.com/@chyun55555/ensemble-learning-voting-and-bagging-with-python-40de683b8ff0 (Accessed June, 2023).

[15] Python kitchen, "Blending Algorithms in Machine Learning", Available from: https://www.pythonkitchen.com/blending-algorithms-in-machine-learning/ (Accessed June, 2023).
[16] Kaggle,https://www.kaggle.com/code/iamamir/fake-social-media-account-detection/input (Accessed June, 2023).
[17] Kadam, V. J., Jadhav, S. M., Vijayakumar, K., "Breast cancer diagnosis using feature ensemble learning based on stacked sparse autoencoders and softmax regression", Journal of medical systems 43(8) (2019) : 263.