Вісник Тернопільського національного технічного університету
https://doi.org/10.33108/visnyk_tntu
Scientific Journal of the Ternopil National Technical University
2024, № 1 (113) https://doi.org/10.33108/visnyk_tntu2024.01
ISSN 2522-4433. Web: ***visnyk.tntu.edu.ua***

UDK 004.89: 002.53

APPLICATION OF MACHINE LEARNING METHODS FOR PREDICTING THE RISK OF STROKE OCCURRENCE

Liubomyr-Oleksii Chereshchuk; Nataliia Melnykova

Lviv Polytechnic National University, Lviv, Ukraine

Summary. *In the paper, research was carried out in the medical field, which is very important for people and is gaining more and more importance every year. The study was aimed at predicting the occurrence of a stroke, this disease is a serious threat to people's health and lives. To build machine learning models that could solve the problem of predicting the occurrence of a stroke, a very unbalanced dataset was used, which made the work difficult. The best results were shown by the Random Forest model, which reached precision, recall, and f1-score equal to 90%. The obtained results can be useful for doctors and medical workers engaged in the diagnosis and treatment of stroke.*

Key words: *machine learning, stroke, decision tree, random forest, k-neighbors, ada boost, stacking, SMOTE, grid search.*

https://doi.org/10.33108/visnyk_tntu2024.01.027 *Received 14.12.2023*

Statement of the problem. Stroke is one of the most common causes of death and disability in the world. Often, people who are at increased risk of stroke are not aware of it and therefore do not seek help from their doctors. In addition, even if a person sees a doctor, the diagnosis can be difficult, which delays the necessary treatment and can lead to serious consequences. Thus, there is a need to develop an effective stroke prediction system that would help reduce the risk of stroke and improve people's quality of life. In this context, machine learning methods can become a powerful tool for analyzing and predicting the risk of stroke. Therefore, the problem is that it is necessary to investigate the effectiveness of different machine learning methods to effectively predict the risk of stroke and to find the best methods to solve this prediction problem.

Analysis of the available investigations. The authors of the paper [1] aimed to propose a stroke prediction model using machine learning classifiers and a stacking ensemble classifier. The proposed stacking prediction model showed an accuracy rate of 97%. However, the study has some drawbacks that may limit its use. For example, the process of selecting and preparing data for analysis is not sufficiently described. In the paper [2], the authors conducted a study that proposes a machine learning approach for stroke diagnosis using unbalanced data. The randomized sampling (ROS) technique was used in this study to balance the data. The results showed that Machine Support Vector has the highest accuracy of 99.99%. Random Forest showed the second highest accuracy rate – 99.87%. However, the article has several drawbacks: the study uses a fairly limited amount of data, which may affect the accuracy of the results. In the paper [3], several models for predicting stroke risk were developed and evaluated using machine learning. The experimental results showed that the stacking classification achieved an AUC of 98.9% and an accuracy of 97.4%. The disadvantage of this article is that the study does not compare the effectiveness of the methods used with other methods of stroke risk prediction. In general, the papers [1–3] achieved very high accuracy, recall, precision, and f1-score, including solving the problem of data imbalance using various. However, rather high accuracy rates of more than 98%–99%, this tells that the models may be overtrained. In the paper [4], the

Corresponding author: Liubomyr-Oleksii Chereshchuk; e-mail: liubomyrchereshchuk@gmail.com

authors obtained the best accuracy from Random Forest, which amounted to 96.01%. However, the study has some shortcomings: the authors did not check for data imbalance, which casts doubt on the high accuracy rates. In the paper [5] the algorithm that performed best was the Naive Bayes algorithm, which yielded an accuracy of approximately 82%. However, the work has some drawbacks: relatively low accuracy achieved.

Thus, the reviewed studies have many strengths. However, these studies also have a number of shortcomings that will be addressed. For example, solving the problem of unbalanced data to achieve true results, solving the problem of underfitting models to obtain true and qualitative results.

The Objective of the work is the development of a software product – a program to predict the risk of stroke using advanced machine learning methods. The goal is driven by the need to achieve true and high-quality results for further use in medicine.

Statement of the task. To pre-process data to achieve better and more plausible results; to create fast, efficient and optimized machine learning models; achieve a good level of performance evaluation indicators for machine learning models, including accuracy; and search for and select optimal hyperparameters for the machine learning models used.

Research part and results. Now, let's move on to describe a dataset. It was obtained from the website of DataHack Analytics Vidhya [6].

The dataset consists of 11 columns, and 4981 rows. The columns have 'gender', 'age', 'hypertension', 'heart_disease', 'ever_married', 'work_type', 'Residence_type', 'avg_glucose_level', 'bmi', 'smoking_status' and 'stroke' as the main attributes. The output column is 'stroke'. Table 1 shows a description of the data columns from the dataset used:

Table 1

Stroke DataSet

Column Name	Type(Values) of the column	Description of the column
gender	String(Male, Female, Other)	Gender of the patient
age	Integer	Age of the patient
hypertension	Integer(1, 0)	Whether the patient has hypertension or not
heart_disease	Integer(1, 0)	Whether the patient has heart disease or not
ever_married	String(Yes, No)	Whether the patient is married or not
work_type	String(Govt_job, Never_worked, Private, Self-employed, children)	Categories for work of the patient
Residence_type	String(Urban, Rural)	Categories for residence type of the patient
avg_glucose_level	Float	Value of the average glucose level of the patient
bmi	Float	Value of the Body Mass Index of the patient
smoking_status	String(formerly smoked, never smoked, smokes, unknown)	Categories for smoking status of the patient
stroke	Integer(1, 0)	Whether the patient has stroke or not

The dataset is highly unbalanced, as the value of no stroke (value equal to 0) occurs 4733 times, and there was a stroke (value equal to 1) only 248 times.

Data preprocessing:

Data preprocessing plays a very important role in preparing data for training machine learning models. In our case, based on the structure and content of our data, as well as the task set in this paper, we will perform the following data preprocessing operations: Outlier removal will be done by the interquartile range method; Categorical data encoding will be done by the one-hot-encoding method; Deal with unbalanced data will be done by the SMOTE method; The dataset will be split in the ratio of 80% and 20%, where 80% is the data for training the model and 20% is the data for testing the model; Attribute scaling will be performed by using the min-max-scaler method.

Machine learning models:

Now, we can move on to selecting specific classification models for our task: Decision Tree is a machine learning algorithm that can be used for classification tasks; Random Forest is a machine learning algorithm that belongs to the category of ensemble methods and can be used for solving classification problems; K-Neighbors is a machine learning algorithm that can be used for classification tasks based on the K-Nearest Neighbors (K-NN) method; Ada Boost, or Adaptive Boosting, is a machine learning algorithm that belongs to the category of ensemble methods and can be used for classification tasks; Stacking is a machine learning algorithm that also belongs to the to the category of ensemble methods and can be used for classification tasks, but this algorithm differs from more traditional ensemble methods, such as Random Forest or Ada Boost.

Hyperparameters:

In our case, when the data volume is 4981 rows and 11 columns, which is not a very large amount of data, it is a good choice to try using the Grid Search method. Grid Search is a method of tuning model hyperparameters that involves testing all possible combinations of hyperparameters that are specified by the user in advance.

Evaluation metrics:

Performance metrics help determine how well a model performs its task. Let's move on to consider the main evaluation metrics:

Accuracy is the ratio of the number of correctly classified instances to the total number of instances in the test dataset. Accuracy formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

TP – the number of correctly classified positive examples; TN – the number of correctly classified negative examples; FP – the number of incorrectly classified positive examples; FN – the number of incorrectly classified negative examples.

Precision determines which part of the positively categorized examples are really positive. Recall determines what proportion of the truly positive examples were correctly classified. Precision and recall formulas:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

TP, FP and FN are determined as described earlier.

The F1-score is a harmonic mean between accuracy and completeness. It helps to balance accuracy and completeness in one number. F1-score formula:

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

Precision and Recall are determined as described earlier.

The confusion matrix displays the number of correctly and incorrectly classified examples for each class.

Based on the task at paper and dataset, in the work will be used f1-score, precision, and recall as the main metrics; confusion matrix as an auxiliary metric; and accuracy to compare with the main metrics to evaluate the solution to the problem of imbalance in our dataset.

Data preprocessing results:

First, let's look at the results of removing outliers in the data. Figure 1 shows the avg_clucose_level and bmi before outliers are removed:

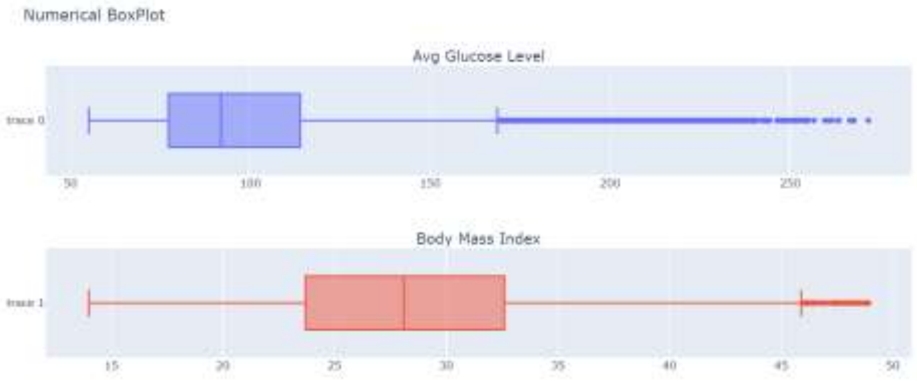


Figure 1. Avg_glucose_level and bmi data before removing outliers

Now let's look at this data after removing the outliers, as shown in Figure 2:

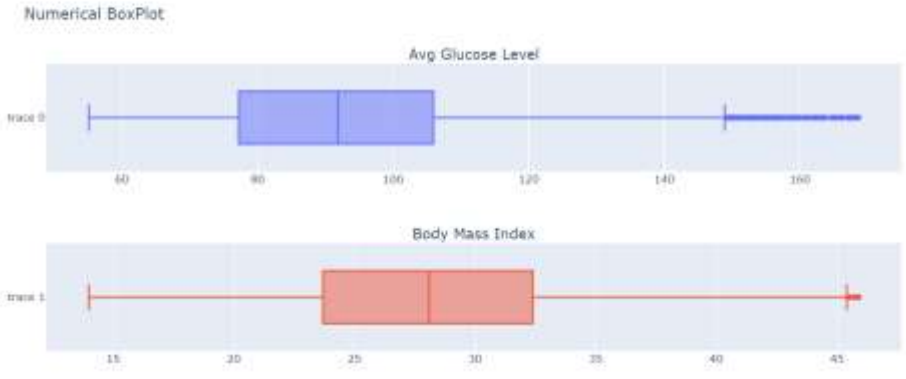
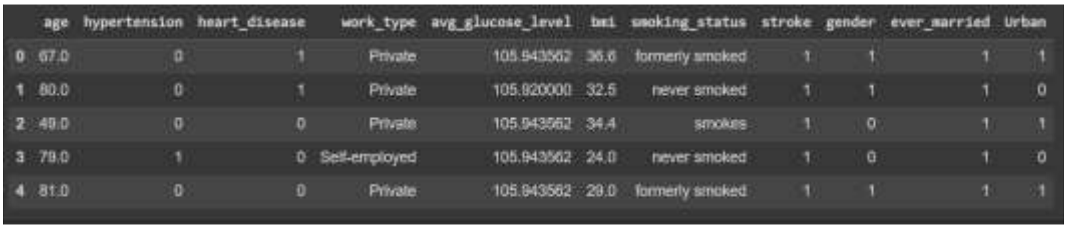


Figure 2. Avg_glucose_level and bmi data after removing outliers

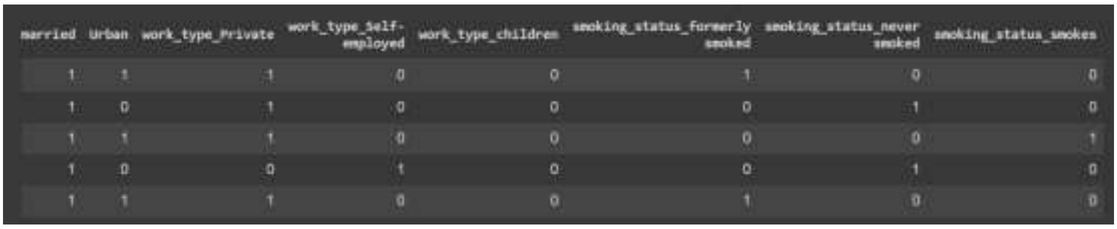
From these figures, we can conclude that the outliers removal operation was successful.

Next, let's move on to the results of encoding categorical features. These results are shown in Figures 3–4:



	age	hypertension	heart_disease	work_type	avg_glucose_level	bmi	smoking_status	stroke	gender	ever_married	Urban
0	67.0	0	1	Private	105.943562	36.6	formerly smoked	1	1	1	1
1	80.0	0	1	Private	105.920000	32.5	never smoked	1	1	1	0
2	49.0	0	0	Private	105.943562	34.4	smokes	1	0	1	1
3	79.0	1	0	Self-employed	105.943562	24.0	never smoked	1	0	1	0
4	81.0	0	0	Private	105.943562	29.0	formerly smoked	1	1	1	1

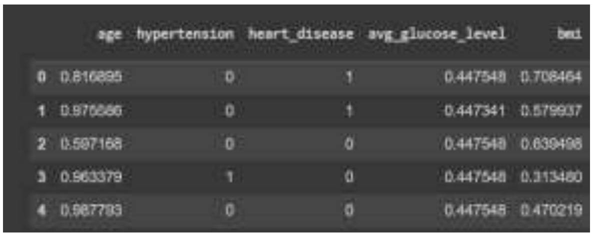
Figure 3. The results of encoding the features – gender, ever_married and Residence_type



married	Urban	work_type_Private	work_type_Self-employed	work_type_children	smoking_status_formerly smoked	smoking_status_never smoked	smoking_status_smokes
1	1	1	0	0	1	0	0
1	0	1	0	0	0	1	0
1	1	1	0	0	0	0	1
1	0	0	1	0	0	1	0
1	1	1	0	0	1	0	0

Figure 4. The results of encoding the features – work_type and smoking_status

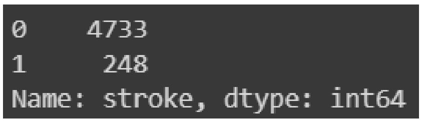
Now let's move on to the attribute scaling operation using the min max scaler method. The results of attribute scaling are shown in Figure 5:



	age	hypertension	heart_disease	avg_glucose_level	bmi
0	0.816895	0	1	0.447548	0.708464
1	0.975586	0	1	0.447341	0.579937
2	0.597168	0	0	0.447548	0.639498
3	0.963379	1	0	0.447548	0.313480
4	0.987793	0	0	0.447548	0.470219

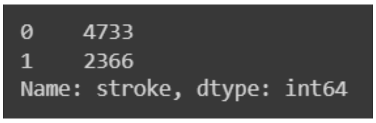
Figure 5. Results of scaling attributes – age, avg_glucose_level and bmi

Now let's move on to solving the problem of unbalanced data. The amount of data in the stroke column before and after the SMOTE method is shown in Figures 6–7:



```
0    4733
1     248
Name: stroke, dtype: int64
```

Figure 6. The amount of stroke data before applying the SMOTE method



```
0    4733
1    2366
Name: stroke, dtype: int64
```

Figure 7. The amount of stroke data after applying the SMOTE method

These figures show that after applying the SMOTE method, our data became balanced and now the number of 1 class is much higher due to the addition of synthetic instances.

Models results:

Now, let's move on to the results of training and testing our models, first looking at the performance of the base models with default hyperparameters on the testing data. The results of testing the basic Decision Tree Classifier model are shown in Table 2:

Table 2

Results of the basic Decision Tree Classifier model

class	precision	recall	f1-score	accuracy
0 (no stroke)	0.89	0.91	0.81	0.97
1 (stroke)	0.82	0.80		

The following conclusions can be drawn from this table: The model performed an almost stable and balanced classification of the data, because the precision and recall values are quite close for the two classes, which indicates that the model distinguishes between the two classes well; The f1-score is 76%, which is not a very high value.

Now let's look at the results of testing the basic Random Forest Classifier, which are shown in Table 3:

Table 3

Results of the basic Random Forest Classifier model

class	precision	recall	f1-score	accuracy
0 (no stroke)	0.93	0.94	0.86	0.9
1 (stroke)	0.87	0.86		

The following conclusions can be drawn from this table: The model performed a stable and balanced data classification, just like the baseline Decision Tree Classifier model; The precision and recall values are quite close for the two classes. The f1-score is 86%, which is quite a high value and acceptable for further improvement.

Now let's look at the results of testing the basic K-Neighbors Classifier, which are shown in Table 4:

Table 4

Results of the basic K-Neighbors Classifier model

class	precision	recall	f1-score	accuracy
0 (no stroke)	0.84	0.95	0.82	0.86
1 (stroke)	0.92	0.75		

The following conclusions can be drawn from this table: The model did not perform a very stable classification; The precision score is almost close for both classes, but the recall score is quite different; The f1-score and accuracy are not very high – 82% and 86%.

Now let's look at the results of testing the basic AdaBoost Classifier, which are shown in Table 5:

Table 5

Results of the basic AdaBoost Classifier model

class	precision	recall	f1-score	accuracy
0 (no stroke)	0.82	0.86	0.7	0.79
1 (stroke)	0.73	0.68		

The following conclusions can be drawn from this table: The model did not make a very stable classification; The precision score is closer for both classes, but the recall score is very different; The f1-score is not high – 70%.

Now let's look at the results of testing the basic Stacking Classifier, which are shown in Table 6:

Table 6

Results of the basic Stacking Classifier model

class	precision	recall	f1-score	accuracy
0 (no stroke)	0.94	0.93	0.87	0.91
1 (stroke)	0.87	0.88		

The following conclusions can be drawn from this table: The model performed a stable classification, similar to the base Random Forest Classifier model; The precision and recall scores are quite similar for both classes; The f1-score is 87%, which is the best result among the baseline models.

So, in general, we can conclude that the Random Forest Classifier model performed the best among the baseline models, as it was quite stable and did not actually get confused between classes, and its f1-score was 86%, which is a pretty good value for such a dataset.

Now let's move on to the results of finding the best hyperparameters for our models using the Grid Search method. As a result of this method, the Random Forest Classifier model was improved the most significantly. It was built a model with the following hyperparameters: n_estimators – 500; criterion – entropy; max_depth – 30. The results of this model are shown in Table 7 and Figure 8:

Table 7

Results of the improved Random Forest Classifier model

class	precision	recall	f1-score	accuracy
0 (no stroke)	0.93	0.94	0.9	0.91
1 (stroke)	0.89	0.86		

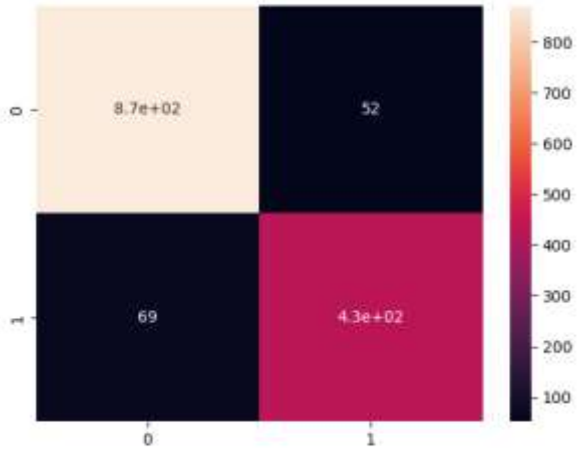


Figure 8. Confusion matrix for improved Random Forest Classifier

The following conclusions can be drawn from these results: The f1-score has increased by 4% and is equal to 90%, which is a rather high value; The model also performed a balanced classification; The number of classification errors in the 0 (no stroke) class is 69, and the number of class 1 (stroke) errors is 52.

Comparison with a trained model on unbalanced data:

To evaluate the impact of imbalanced data, we will create a random forest classifier model without using the SMOTE method and see how this model performs:

Table 8

Results of the Random Forest Classifier model without SMOTE method

class	precision	recall	f1-score	accuracy
0 (no stroke)	1	0.96	0.98	0.96
1 (stroke)	0	0	0	

The following conclusions can be drawn from this table: The model performs very poorly and is not stable; The model was not able to classify class 1 (stroke) at all; The accuracy of the model is 96%, but this metric does not correspond to the actual performance of the model; The precision, recall and f1-score for class 1(stroke) are 0, so the model has not learned to distinguish this class.

So, we can conclude that when the data is very unbalanced, it is definitely worth solving this problem, because the results of model training will not be true. The accuracy metric is a poor choice for validating classification models.

Conclusions. The paper performed research in the medical field, which is very important for people and is becoming more and more important every year. The study was aimed at predicting the occurrence of stroke, which is a serious threat to human health and life.

The best model (Random Forest Classifier) showed the following values for these indicators: Precision – 90%; Recall – 90%; f-1 score – 90%. The accuracy score was also used, which was equal to 91%, but in order to show the inappropriateness of this indicator in classification tasks, the Random Forest Classifier model was trained on data that was processed without the stage of solving the problem of data imbalance. As a result, the model showed an accuracy rate of 96%, but the precision, recall, and f1-score for the 1 (stroke) class were 0%. These results showed that the model learned poorly and was unable to classify this class at all.

The findings of this study are quite important because they can help doctors implement preventive measures more effectively and increase the chances of saving patients' lives and health from stroke.

References

1. Mostafa S. A., Elzanfaly D. S., Yakoub A. E. A Machine Learning Ensemble Classifier for Prediction of Brain Strokes. International Journal of Advanced Computer Science and Applications (IJACSA). 2022. Issue 13. No. 12. [In English]. https://doi.org/10.14569/IJACSA.2022.0131232
2. Biswas N., Uddin K. M. M., Rikta S. T., Dey S. K. A comparative analysis of machine learning classifiers for stroke prediction: A predictive analytics approach. Healthcare Analytics. 2022. Issue 2. P. 100116. [In English]. https://doi.org/10.1016/j.health.2022.100116
3. Dritsas E., Trigka M. Stroke Risk Prediction with Machine Learning Techniques. Sensors. 2022. Issue 22. No. 13. P. 4670. [In English]. https://doi.org/10.3390/s22134670
4. Khan M. K. Computer Science and Engineering. 2022. [In English].
5. Sailasya G., Kumari G. L. A. Analyzing the Performance of Stroke Prediction using ML Classification Algorithms. International Journal of Advanced Computer Science and Applications (IJACSA). 2021. Issue 12. No. 6. [In English]. https://doi.org/10.14569/IJACSA.2021.0120662
6. DataHack : Biggest Data hackathon platform for Data Scientists. Web Resource. [In English]. URL: https://datahack.analyticsvidhya.com.

Список використаних джерел

1. Mostafa S. A., Elzanfaly D. S., Yakoub A. E. A Machine Learning Ensemble Classifier for Prediction of Brain Strokes. International Journal of Advanced Computer Science and Applications (IJACSA). 2022. Issue 13. No. 12. [In English]. https://doi.org/10.14569/IJACSA.2022.0131232
2. Biswas N., Uddin K. M. M., Rikta S. T., Dey S. K. A comparative analysis of machine learning classifiers for stroke prediction: A predictive analytics approach. Healthcare Analytics. 2022. Issue 2. P. 100116. [In English]. https://doi.org/10.1016/j.health.2022.100116
3. Dritsas E., Trigka M. Stroke Risk Prediction with Machine Learning Techniques. Sensors. 2022. Issue 22. No. 13. P. 4670. [In English]. https://doi.org/10.3390/s22134670
4. Khan M. K. Computer Science and Engineering. 2022. [In English].
5. Sailasya G., Kumari G. L. A. Analyzing the Performance of Stroke Prediction using ML Classification Algorithms. International Journal of Advanced Computer Science and Applications (IJACSA). 2021. Issue 12. No. 6. [In English]. https://doi.org/10.14569/IJACSA.2021.0120662
6. DataHack : Biggest Data hackathon platform for Data Scientists. Web Resource. [In English]. URL: https://datahack.analyticsvidhya.com.

УДК 004.89: 002.53

ЗАСТОСУВАННЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ ДЛЯ ПРОГНОЗУВАННЯ ЗАГРОЗИ ВИНИКНЕННЯ ІНСУЛЬТУ

Любомир-Олексій Черещук; Наталія Мельникова

Національний університет «Львівська Політехніка», Львів, Україна

Резюме. *Проведено дослідження в медичній сфері, яка є дуже важливою для людей і з кожним роком набуває все більшого значення. Дослідження спрямоване на прогнозування виникнення інсульту. Це захворювання є серйозною загрозою для здоров'я та життя людей. Для побудови моделей машинного навчання, які б могли вирішити проблему прогнозування виникнення інсульту, використовувався дуже незбалансований набір даних, що ускладнювало роботу. Використовуючи опрацьовані дані, отримані в результаті застосування методів попереднього опрацювання даних, побудовано та порівняно різні моделі машинного навчання для вирішення поставленої задачі класифікації. Найкращі результати показала модель random forest, яка досягла precision, recall та f1-score на рівні 90%. Також використовувався показник accuracy, який дорівнював 90%. Однак для того, щоб показати недоцільність даного показника в задачах класифікації, було натреновано модель random forest classifier з найоптимальнішими гіперпараметрами, отриманими за допомогою методу grid search на даних, які опрацьовані без етапу вирішення проблеми незбалансованості даних. У результаті модель показала показник accuracy, що дорівнював 96%. Проте показники precision, recall та f1-score для 1 (буде інсульт) класу дорівнювали 0%. Такі результати показали, що модель погано навчилася та взагалі не змогла класифікувати даний клас. Отже. в результаті виконання даної роботи побудовано модель random forest classifier для вирішення проблеми прогнозування виникнення інсульту, яка показала добрі результати оцінювання показників precision, recall та f1-score – 90%. Отримані результати дослідження є досить важливими, оскільки можуть допомогти лікарям ефективніше впроваджувати профілактичні заходи і збільшити шанси на порятунок життя та здоров'я пацієнтів від інсульту.*

Ключові слова: *машинне навчання, інсульт, дерево рішень, випадковий ліс, k-сусідів, адаптивне прискорення, стекування, техніка надмірної вибірки синтетичної меншості, сітковий пошук.*

https://doi.org/10.33108/visnyk_tntu2024.01.027 *Отримано 14.12.2023*