СЕКЦІЯ 6
МАТЕМАТИЧНІ МЕТОДИ, МОДЕЛІ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В ЕКОНОМІЦІ

DOI: 10.32999/ksu2307-8030/2020-37-17

UDC 336:338.27:658.29

Tupko Natalia
Associate Professor of the Department of Higher and Computational Mathematics
National Aviation University
ORCID: https://orcid.org/0000-0003-0625-3271
E-mail: natupko@ukr.net

Vasil'eva Nataliia
Associate Professor of the Department of Higher Mathematics
Odessa State Academy of Civil Engineering and Architecture
ORCID: https://orcid.org/0000-0003-0211-7141
E-mail: vns02011962@gmail.com

Vasil'ev Alexander
Associate Professor of the Department of Mathematical and Computer Modeling
Odessa I.Mechnikov National University
ORCID: https://orcid.org/0000-0002-3826-4883
E-mail: av5111955@gmail.com

PREDICTION OF DATA IN THE INSURANCE INDUSTRY BASED ON NEURAL NETWORK METHODS

The paper presents a comparative analysis of the generalized linear regression model with the leading machine learning method Feed Forward Neural Network (FFNN) from the point of view of predicting counting data. These two models are described and compared from a theoretical and practical point of view. The stability of the models on the bicycle rental data set is checked, their accuracy is evaluated, the learning curves are built on test and training data sets. In order to improve the interpretability of models, the importance of input variables is evaluated. Because FFNN is often called the “black box” method, there is no direct way to evaluate the importance of variables. A new indirect method for assessing the importance of variables for deep neural networks based on the principles of information theory is proposed. It has been demonstrated that the FFNN network provides much better predictive power compared to the generalized linear regression model with a slight increase in model complexity.

Keywords: generalized linear regression model of Poisson, Feed Forward Neural Network, Poisson distribution, machine learning, bicycle rental dataset.

Тупко Н.П., Васильєва Н.С., Васильєв О.Б. ПРОГНОЗУВАННЯ ДАНИХ ПІДРАХУНКУ У СТРАХОВІЙ ГАЛУЗІ НА ОСНОВІ МЕТОДІВ НЕЙРОННИХ МЕРЕЖ

Прогнозування даних підрахунку – одна з ключових задач у страховій галузі, економіці та соціальних науках. Регресійний аналіз зазвичай відноситься до класичного підходу для вирішення цієї задачі. Однак класична регресійна модель Пуассона часто має обмежене застосування, оскільки емпіричні набори даних підрахунку зазвичай демонструють велику дисперсію та надмірну кількість нулів, а отже незбалансованість у даних. Зважаючи на це, а також на позитивні результати машинного навчання у різних галузях, розглянуто його як достойну альтернативу класичному підходу. У цій роботі проводиться порівняльний аналіз узагальненої лінійної регресійної моделі Пуассона (GLM) з нейронною мережею прямого поширення (Feed Forward Neural Network – FFNN), що є провідним методом машинного навчання, з точки зору прогнозування даних підрахунку і подальшого використання на практиці. Стаття описує дві моделі та порівнює їх з теоретичної та практичної точок зору. Протестовано їх стійкість, використовуючи набір даних про прокат велосипедів. Для кращого розуміння моделей, оцінюється їх точність та будуються криві навчання на тестових і навчальних наборах. Крім того, оцінюється важливість вхідних змінних для кращої інтерпретації алгоритмів. Оскільки FFNN є так званим методом «чорної скриньки», для нього не існує прямого способу оцінки змінних. Запропоновано нову технологію оцінки важливості вхідних даних для глибоких нейронних мереж відповідно до принципів теорії інформації. У роботі продемонстровано, що нейронна мережа прямого поширення (FFNN) у порівнянні з узагальненою лінійною регресійною моделлю Пуассона (GLM) забезпечує набагато більшу потужність при незначному збільшенні складності моделі. При побудові нейронних мереж використовувались стандартні пакети мови програмування Python, які можна швидко адаптувати до інших наборів даних. Тому підхід, запропонований у даній статті, можна успішно використовувати при вирішенні багатьох інших економічних задач. Алгоритми, побудовані за допомогою машинного навчання, точніше прогнозують дані підрахунку і можуть служити добрим орієнтиром для інших моделей.

Ключові слова: узагальнена лінійна регресійна модель Пуассона, нейронна мережа прямого поширення, пуассоновський розподіл, машинне навчання, набір даних про прокат велосипедів.

Тупко Н.П., Васильева Н.С., Васильев А.Б. ПРОГНОЗИРОВАНИЕ ДАННЫХ ПОДСЧЕТА В СТРАХОВОЙ ОТРАСЛИ НА ОСНОВЕ МЕТОДОВ НЕЙРОННЫХ СЕТЕЙ

В работе представлен сравнительный анализ обобщенной линейной регрессионной модели с ведущим методом машинного обучения на основе глубокой нейронной сети FFNN с точки зрения прогнозирования данных подсчета. Эти две модели описываются и сравниваются с теоретической и практической точек зрения. Проверяется устойчивость моделей на наборе данных об аренде велосипедов, оценивается их точность, строятся кривые обучения на тестовых и обучающих наборах данных. С целью улучшения интерпретируемости моделей оценивается важность входных переменных. Поскольку FFNN часто называют методом «чёрного ящика», то в этом случае не существует прямого способа оценки важности переменных. В работе предложен новый косвенный метод оценки важности входных данных для глубоких нейронных сетей, основанный на принципах теории информации. Продемонстрировано, что нейронная сеть FFNN обеспечивает намного лучшую предсказательную силу по сравнению с обобщенной линейной регрессионной моделью при небольшом увеличении сложности модели.

Ключевые слова: обобщённая линейная регрессионная модель Пуассона, нейронная сеть прямого распространения, пуассоновское распределение, машинное обучение, набор данных по прокату велосипедов.

Problem statement. Poisson distribution, that is mainly considered in this paper, serves to model occurrences of the events, such as number of phone calls at service center, arrival times of customers, occurrences of natural disasters, number of insurance claims etc. Wide application of the Poisson distribution implies its parameters estimation being an *actual problem*. Till now regression analysis, namely generalized linear regression model of Poisson (GLM), was the common approach to deal with it. The problem of this article is to design a alternative deep learning network algorithm for count data prediction using log-likelihood for Poisson distribution as a cost function and accuracy as a performance metric. Accuracy and run time of the algorithms GLM and FFNN have to be compared with the benchmark and analysed accordingly. Finally, the performance of the FFNN algorithm should be sufficiently high, and numerical measures of feature importance has to be provided.

Analysis of recent research and publications. We will make a brief review of the latest research and publications on this topic. The book [3] represents GLM as a core approach in non-life insurance and the whole insurance industry as well. The book covers Poisson, binomial and negative binomial distributions from exponential family as commonly applied to count data. Although GLM is a widely used prediction algorithm, [3] reveals a number of challenges the method brings out. In the first place, it has manual feature engineering process that is error-prone and extremely resource demanding. This way sets up a functional representation of the features dependence a priori and hence narrows down a hypothesis space drastically.On the other hand, [5] proves deep neural network to follow the opposite way. Generally speaking, its cost function is represented as a composition of weight matrices. This enables an automatic feature engineering which implies much powerful hypothesis space. Theoretical reasoning of [5] as well as [4] and [6] goes along with practical evidence. An article [1] demonstrates advantage of neural networks and the other machine learning methods to compare with GLM in scope of predicting number of spikes.

Selection of previously unsolved parts of the common problem. A major concern regarding deep neural network algorithm is a high model complexity followed by increasing computational time and low interpretability of the results. Linear model, in contrast, demonstrates simple and clear model structure together with neglectable run time. In particular, it allows for a straightforward evaluation of feature importances. On the other hand, deep neural networks can potentially cover much wider set of the hypothesis and hence explaine much more complex data structure, which means more powerful predictions in comparison to GLM. Consequently, a researcher often faces a problem of a choice between the algorithms. Thus, core open questions are: 1) does the performance of deep neural networks outweigh its complexity; 2) what quantitative metrics are there to evaluation feature importance of the input variables for deep neural networks?

The purpose of this study is application of Feed Forward Neural Networks model for count variables prediction based on given input features, as well as to compare the efficiency and accuracy of this model with the characteristics of classical regression models. On top of that, we aim to increase interpretability of FFNN and give a qualitative assessment of the predictions.

Presentation of the main research results. We turn to the presentation of the main results of this work. First of all, we will set the initial dataset for constructing the compared models.

I. DATA

Hourly bike rental data during two years 2011-2012 in Germany are provided as a dataset [7]. Generally, variables describe total number of rented bikes, weather conditions and timestamp.

A. Data description

Table 1 describes meaning of response “casual” and the other variables, their domains and transformations, which ensures correct interpretation of the variables before feeding in the models. More detailed explanation follows accordingly.

B. Data preparation

Row data are often not appropriate for analysis or misleading. For example, they could be characters while algorithm requires numerical one; continue ordinal effects being categorical; have large variance incomparable with the other numerical variables.

To get correct input data the following techniques are usually implied.

Normalization

Normalization is applied to numerical columns to use the common scale, so that no information is lost and no difference in the range of values are made. Otherwise it could lead to extremely large or extremely small weights while combining these variables by modeling.

For instance, to make variable “temp” (<=40) more influential than “humidity” (<=100) we need at least twice bigger weights.

There are multiple techniques of normalization, but here we change all values to a 0-1 scale as following:

$$x_{norm} = \left(x_{orig} - x_{max}\right) / \, x_{max} \in [0;1] \,, \tag{1}$$

x_{orig} – original variable,

– variable maximum,

x_{norm} – resulting normalized variable.

OHE

One hot encoding (OHE) or dummy encoding is a technique which represents categorical variable in numerical form without keeping ordinal effect in it.

Table 1

Description of response variable and input features

Variable name	Values	Transformation*	Description
casual	int, >=0	no transformation	number of rentals

datetime	%Y-%m-%d %H:%M:%S	ohe(Y,M,D,H)
		norm(dist_in_hrs)
			date and time of rental

season	1 [spring], 2 [summer], 3 [fall], 4 [winter]	ohe	season
holiday	0/1	ohe	if the day is identified as a holiday
workingday	0/1	ohe	if the day is identified neither a weekend nor holiday
weather	1 [clear or partly cloudy] 2 [mist and/or partly cloudy] 3 [light snow or rain] 4 [heavy rain or snow]	ohe	weather type
temp	Real	norm	temperature in Celsius
atemp	Real	norm	subjective feeling of temperature in Celsius
humidity	real, >=0	norm	relative air humidity
windspeed	real, >=0	norm	wind speed

* "ohe" and "norm" abbreviations are explained in subsection B. Data preparation.

It namely represents a feature as a set of binary vectors. Each level m out of M possible is represented as a vector that has all zero values except the index of the level i, which is marked with 1. Eventually, we get M vectors, one of them is linearly dependent on the other $M-1$ and thus redundant. Hence, we keep only $M-1$ binary vectors over the original feature.

For example, variable "weather" is categorical but encoded as numerical, 1…4.

To remove order effect (1<4) we use the following split of the column:

Weather	w_1	w_2	w_3	w_4
1	1	0	0	0
2	0	1	0	0
4		0	0	1
4	0	0	0	1
3	0	0	1	0

Figure 1. One hot encoding of feature "weather"

Date

The input sample is assumed to be mutually independent, thus "datetime" variable is proceed as following:

– hour, day and month are treated as categorical to capture circularity effects. OHE is applied;

– artificial variable "date_diff_inHRS" is created to measure distance of an instance to minimal date in hours and keep order between them. Normalization is applied.

Data split

Data are split into two subsets train, 80% of data, and test, 20% of data. The first one is used to train model and estimate its parameters. The latter one is used to get an unbiased estimation of model predictive power.

To get a uniform subsets and hence more representative training, we applied the stratified shuffle split. The folds are made in a way to preserve the distribution of the response. As its values are (theoretically) unlimited from above a shuffling that splits data by equal ratios of response classes is not possible. Thus shuffling by indices was applied.

In the paper the depth of the datasets for training and testing is 8735 and 2151 rows respectively.

II. METHODS

Neural Networks proved to be powerful in a wide scope of applications. In contrast to Recurrent Neural Networks, Feed Forward Neural Networks model data without sequential dependence. As described in chapter II, we assume the instance of the bike dataset to be mutually independent. On the other hand, GLM is treated as a baseline. It has much lower model capacity than NN but is still widely used in practice. Combining with the fact of the response variable being a positive integer, we assume the data to follow conditional Poisson distribution. Hence, loss function is logarithmic likelihood with Poisson distribution. Accuracy is an intuitive performance metric suitable for the datasets with a balanced distribution. We consider it as a suitable performance metric for our dataset.

A. Generalized Linear Regression Model

The Poisson generalized linear model is a multivariate regression model, where response variable follows Poisson distribution and its expected value is modelled as a linear combination of unknown parameters (weights). Mathematically it is described as following. If $x \in \Re^n$ is a vector of independent variables $y \in \Re$ – is a response variable, which follows Poisson distribution with expected value of:

$$logE(y \mid x) = \alpha + \beta' x, \tag{2}$$

where $\alpha \in \Re, \beta \in \Re^n$,

or equivalently:

$$logE(y \mid x) = \theta' x, \tag{3}$$

where $\theta \in \Re^n, x \in \Re^{n+1}$,

and there is a dataset consisting of m vectors $x_i \in \Re^{n+1}, i \in [1,\dots,m]$ and a set of m values $y_1,\dots,y_m \in \Re$ then probability of attaining them has a closed form of:

$$P\left(y_1,\dots,y_m \middle| x_1,\dots,x_m;\theta\right) = \prod_{i=1}^{m} \frac{e^{y_i\theta' x_i} \cdot e^{-e^{\theta' x_i}}}{y_i\,!} = L(\theta \mid x, y) \tag{4}$$

and called a likelihood.

Estimation of parameter θ so that this probability is maximized is called a method of maximum likelihood. In practice minus log likelihood is used as a cost function in a minimization problem:

$$l(\theta|x,y) = \log L(\theta|x,y) = \sum_{i=1}^{m}\left(y_i\theta' x_i - e^{\theta' x_i} - \log y!\right) \qquad (5)$$

In our case $m = 8735$, $n = 74$; as an optimization method Nadam [4], which is Adam RMSprop with Nesterov momentum, is used in paper implementation.

B. Neural Network

Here, we implemented a 4-layer FFNN. It could be considered as embedding of GLM, where each layer is taking the output of previous layers as input. Mathematically speaking, we need to minimize the same loss function $l(\theta|x,y)$ but instead of scalar product of $\theta' x_i$ we use weights' matrix [5].

The nature of the response implies an output activation function. In our case, it could be *rectified linear unit* (ReLU): $x^+ = \max\{0, x\}$. However, we chose its "smoothed" (everywhere differentiable) version called *softplus*: $y = \ln(1 + e^x)$ to avoid problems in the gradient-based optimization. On the other hand, the activation function for hidden layers can have an arbitrary shape but it should eliminate vanishing gradient effect in backpropagation. Thus we prefer both ReLU and softplus over sigmoid and hyperbolic tangence as activation functions for hidden layers [6].

III. RESULTS

We explain why one-layer neural network is equivalent to GLM in chapter II. Using this reasoning, we implement both models within neural network framework using open-source Keras library, running Theano as the backend [2].

To make the results comparable, we evaluated performance of the algorithms on the same test set. The dataset was split into train and test using stratified shuffle split as explained in chapter I. The major parameters of the algorithms are summarized in Table 2.

GLM is a straightforward method that allows only linear interaction among its parameters. Thus, it is unable to describe complex feature dependencies. Formally speaking, the key limitation of GLM is fairly small hypothesis space. Based on the above, we have not expected it to work well. "Figure 2" summarizes accuracy of the algorithms:

Table 2

Parameters and results of glm and ffnn models

Parameters	GLM	FFNN
number of layers	*1*	*4*
activation function	Softplus	
loss function	Poisson	
batch size	50	
Number of epochs	50	

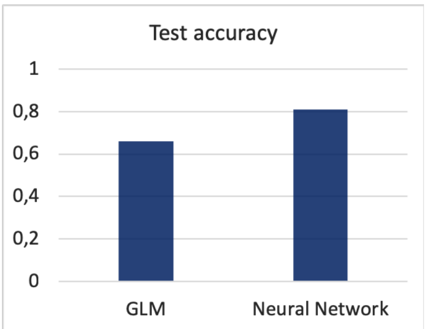

Figure 2. GML and FFNN accuracy based on test set

The accuracy of the FFNN is 0.8154 on the test set, which means that the networks algorithm outperformed GLM in accuracy by 0.1664 (16.64%). The computational time increased from 10.54 min (GLM) to 25.67 min (FFNN). Learning and validation curves of the algorithms enable analyzing learning progress in more detail. As the GLM curves are constantly decreasing (Figure 3), no stopping criteria was triggered. This means, that the model is underfitted or not "complicated enough" to capture the majority of dependency patterns, present in data. This goes along with relatively low accuracy of GLM which is just slightly better than random.

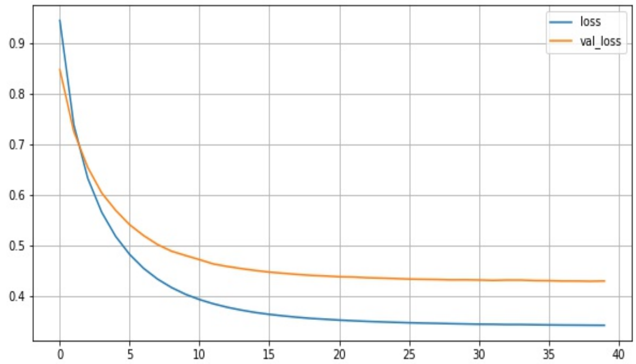

Figure 3. Learning and validation curves of GLM model (without early stopping)

To get the better insight into the modes and validity of their predictions, let us further analyze the resulting feature importance. Linear structure of GLM allows to evaluate the most influential features directly. There are plots of all weights (Figure 4) and 16 feature of the biggest weights in absolute terms (Figure 5).

Conclusions from the conducted research. GLM is a straightforward method that allows only linear interactions among its parameters, this means it is equivalent to one-layer neural network. Hence, we have not expected it to work well in comparison to multi-layer network. Our research demonstrates that FFNN provides much better predictive power indeed with a slight increase in model complexity. As far as we use a standard Python packages, FFNN can be quickly applied to other datasets. This fact ensures our results to be replicable and to be used for much wider range of tasks and data than given here. Our study demonstrates machine learning predictive models to forecast count data more accurately and to serve as a benchmark for other models. In addition, we suggest a *new technique for evaluation of feature importances* for FFNN. As far as deep neural networks are often referred to "black box" algorithms, there is no straightforward method to score importances in contrast to the one by GLM. Hence, we suggest new indirect way of the evaluation in terms of the information theory. It implies that the amount of information of variable and its variance are closely related. Thus, we could compare an original variance of the response to the one, after an increase of a feature variance by say 10%. If the response is relatively insensitive to fluctuations in the future values, then we assume it to be less informative, and hence, less important.

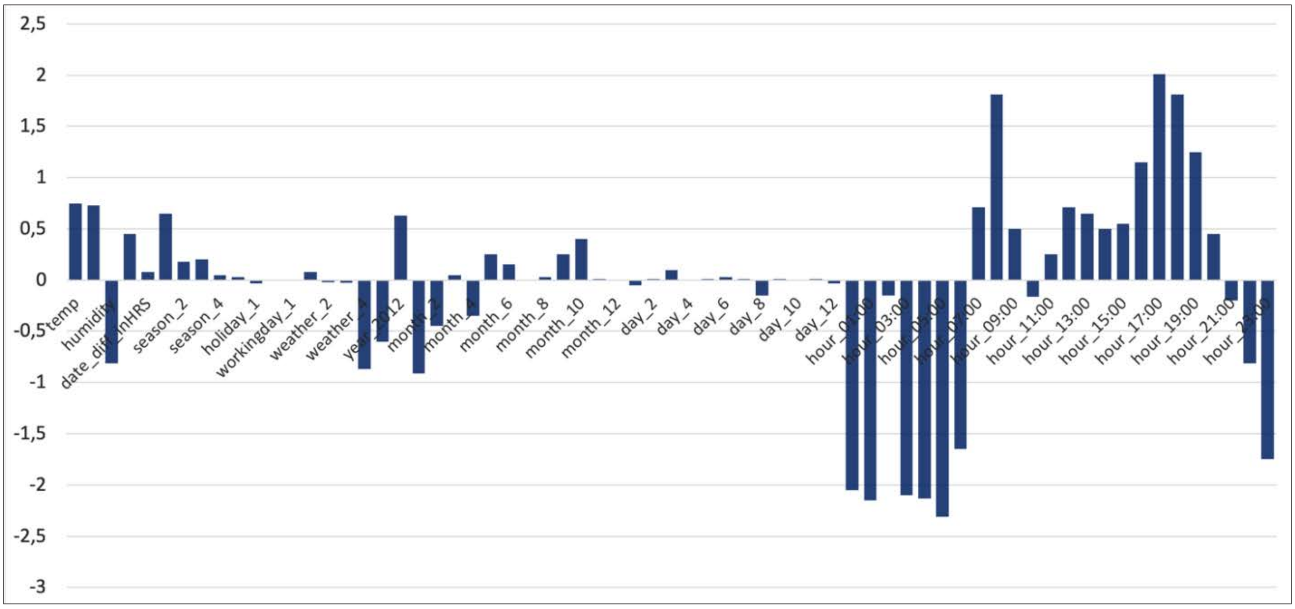

Figure 4. Weights of all input features outputted by GLM

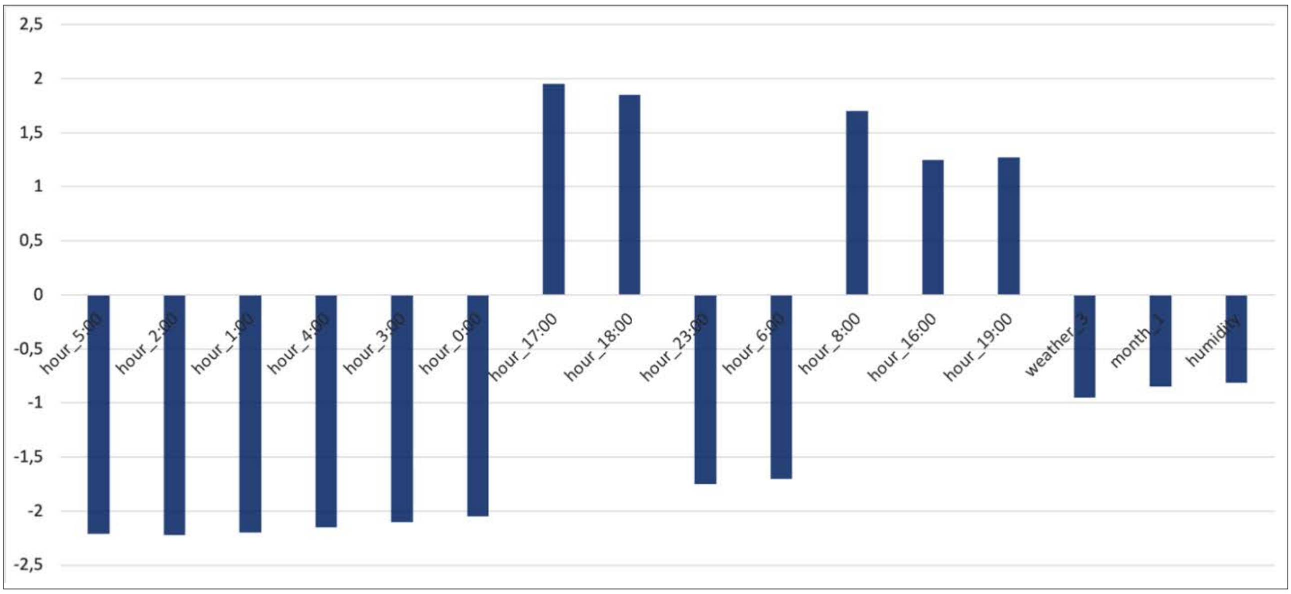

Figure 5. 16 the biggest weights of features outputted by GLM in absolute terms

REFERENCES:

1. Benjamin A., Fernandes H., Tomlinson T., Ramkumar R., Ver-Steeg C., Chowdhury R., … , Kording K. (2017). *Modern machine learning far outperforms GLMs at predicting spikes.* Retrieved from: https://www.biorxiv.org/content/10.1101/111450v2 (accessed January 29, 2020).
2. Open-source neural-network library. Retrieved from: https://keras.io/ (accessed January 21, 2020).
3. Wüthrich M.V. (2018). *Data Analytics for Non-Life Insurance Pricing*. ETH Zurich.
4. Bishop C.M. (2006). *Pattern recognition and machine learning*. Springer.
5. Goodfellow I., Bengio Y., & Courville A. (2016). *Deep learning.* MIT press.
6. Murphy K.P. (2012). *Machine learning: a probabilistic perspective*. MIT press.
7. Competitive web-based data mining platform. Retrieved from: https://www.kaggle.com (accessed January 29, 2020).

Стаття надійшла до редакції 20.02.2020.
The article was received 20 February 2020.