Models of Fraud Detection and Analysis of Payment Transactions Using Machine Learning

Viktor Shpyrko[0000-0003-4955-2685] and Bohdan Koval

Taras Shevchenko National University of Kyiv, 60, Volodymyrska Str., Kyiv, 01033, Ukraine
viksh@bigmir.net, thekovalbohdan@gmail.com

Abstract. The work's aim is to research a set of selected mathematical models and algorithms that examine the data of a single payment transaction to classify it as fraud or verified. Described models are implemented in the form of a computer code and algorithms, and therefore can be executed in real-time. The main objective is to apply different methods of machine learning to find the most accurate, in other words, the one in which the cross-validation score is maximal. Thus, the main problem to resolve is the creation of a model that could instantly detect and block a given fraudulent transaction in order to provide better security and user experience. At first, we determine the classification problem: which initial data we have, how we can interpreter it to find the solution. The next part is dedicated to presenting the methods for solving the classification problem. In particular, we describe such approaches as Logistic Regression, Support Vectors Method (SVM), K-Nearest neighbours, Decision Tree Classifier and Artificial Neural Networks; provide the notion of how these methods operate the data and yield the result. At the end, we apply these methods to the provided data using Python programming language and analyze the results.

Keywords: machine learning, classification problem, fraud, payment systems, neural networks.

1 The Notion of Classification Problem and its Characteristics

1.1 The Definition of Classification Problem

The classification problem is a formalized task, which contains a set of objects (situations), divided in a certain way into classes. There is specified a finite set of objects, and we know to which classes each of them belongs. This set is called a sample. There is no info about other objects, so we do not know to what class they belong. The aim is to create an algorithm that will be able to classify an arbitrary object from the initial set.

To classify an object means to indicate the number (or name) of the class to which this object belongs.

The classification of an object is the number or class name, issued by the classification algorithm because of its application to this particular object [1].

In mathematical statistics, the classification problems are also called as the problems of discrete analysis. In machine learning, the classification problems can be solved with the help of artificial neural network methods, particularly by staging an experiment in the form of training with a teacher.

Let X be a set of object descriptions, Y is a plurality of numbers (or names) of classes. There is an unknown target dependence – mapping (1) – whose values are known only for elements of a finite learning sample (2). The aim is to construct an algorithm (3) capable of classifying any arbitrary object $x \in X$ [2].

$$y^* : X \to Y \tag{1}$$

$$X^m = \{(x_1, y_1), \ldots , (x_m, y_m)\} \tag{2}$$

$$a : X \to Y \tag{3}$$

The probabilistic definition of the problem is more general. It assumes that the set of pairs "object-class" $X \times Y$ is a probabilistic space with an unknown probabilistic degree P. There is a finite study sample of observations (2) generated in accordance with the probabilistic degree P. The aim is to construct an algorithm (3), capable of classifying arbitrary object $x \in X$.

1.2 The Concept of Characteristics in the Tasks of Classification

The characteristic is the mapping (4), where D_f – the set of permissible values of the characteristic.

$$f : X \to D_f \tag{4}$$

If the characteristics $f_1, \ldots, f_n$ are given, then the vector (5) is called the characteristic description of the object $x \in X$.

$$\boldsymbol{x} = (f_1(x), \ldots, f_n(x)) \tag{5}$$

Characteristics can be identified with the objects themselves. In this case, the set (6) is called the space of characteristics.

$$X = D_{f1} \times \ldots \times D_{fn} \tag{6}$$

Depending on the D_f set, the characteristics are divided into the following types:

- Binary characteristics: $D_f = \{ 0, 1 \}$;
- Nominal characteristics: D_f – finite set;
- Sequence characteristics: D_f – finite ordered set;
- Quantitative characteristics: D_f – the set of real numbers.

And into the following classes:

- Two-class classification, which technically is the easiest case, and serves as the basis for solving more complex tasks;

- Multiclass classification. The number of classes reaches thousands (for example, when recognizing hieroglyphs or fused speech), the task of classification becomes significantly more difficult;
- Non-overlapping classes;
- Ordinary classes. An object may belong to several classes at a time;
- Fuzzy classes. It is necessary to determine the degree of belonging of the object to each of the classes, usually it is a valid number from 0 to 1 [2].

In our case, we are interested in the binary characteristic of the set with a two-class specification.

1.3 Publications Dedicated to the Fraud Detection Problem

Bertrand Lebichot and Yann-Ael Le Borgne have researched the problem in the "Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection" publication [3].

They worked on the design of automatic Fraud Detection Systems (FDS) able to detect fraudulent transactions with high precision and deal with the heterogeneous nature of the fraudster behavior. Indeed, the nature of the fraud behavior may strongly differ according to the payment system (e.g. e-commerce or shop terminal), the country and the population segment.

The another publication is "Improving Card Fraud Detection Through Suspicious Pattern Discovery" by Olivier Caelen and Evgueni N. Smirnov [4]. They proposed a new approach to detect credit card fraud based on suspicious payment patterns. According to their hypothesis fraudsters use stolen credit card data at specific, recurring sets of shops. They exploited this behavior to identify fraudulent transactions.

Also the problem was mentioned in "Calibrating Probability with Undersampling for Unbalanced Classification" article by Andrea Dal Pozzolo, Olivier Caelen, Gianluca Bontempi [5]. In this paper, they study analytically and experimentally how undersampling affects the posterior probability of a machine learning model. They formalize the problem of undersampling and explore the relationship between conditional probability in the presence and absence of undersampling. They use Bayes Minimum Risk theory to find the correct classification threshold and show how to adjust it after undersampling.

2 Methods of Solving the Classification Problem

2.1 Regression Methods in Solving Classification Problems

Logistic regression is suitable for solving the classification problem. This is a statistical regression method used in the case when the dependent variable is categorical, so it can acquire only two values (or, more generally, a finite set of values) [6].

Let some set Y have only two values, which are usually indicated by numbers 0 and 1. Let this value depend on some set of explanatory variables (7).

$$x = (1, x_1, x_2, \ldots, x_k) \tag{7}$$

The dependence of Y on $x_1, x_2, \ldots, x_k$ can be determined by adding an additional variable y^*, where (8).

$$y^* = \theta_0 + \theta_1 x_1 + \cdots + \theta_k x_k + u \tag{8}$$

Then (9):

$$Y = \begin{cases} 0, y^* \leq 0 \\ 1, y^* > 0 \end{cases} \tag{9}$$

The next tool is the method of support vectors – a data analysis method for classification and regression using models with controlled training with associated learning algorithms, which are called support vector machines.

For a given set of training samples, each of which is marked as belonging to one or other of the two categories, the training algorithm of the SVM builds a model that relates new samples to one or another category, making it an incredible binary linear classifier. The SVM model is the representation of samples as points in space, displayed in such a way that samples from individual categories are separated by a blank space that is most extensive. New samples then appear to the same space, and predictions about their belonging to the category are based on which side of the gaps they fall.

In addition to performing a linear classification, the SVM can effectively perform a nonlinear classification in the application of the so-called core trick, implicitly displaying its inputs to the spaces of attributes of high dimensionality.

Formally, the support vector machine builds a hyperplane, or a set of hyperplanes in a space of high or infinite dimensionality that can be used for classification, regression, and other tasks. Intuitively, good separation is achieved by a hyperplane that has the greatest distance to the nearest points of the training data of any of the classes (so-called functional separation) [7].

2.2 Discrete Methods in Solving Classification Problems

The next method for solving the problems of classification uses a slightly different approach. The method of *k*-nearest neighbours is a simple nonparametric classification method, where the distances (usually Euclidean) used to classify objects within the space of properties, counted among all other objects. The objects to which the distance is the smallest are selected, and they are allocated in a separate class.

The basic principle of the method of the closest neighbours is that the object is assigned to that class, which is the most common among the neighbours of this element. Neighbours are taken on the basis of a set of objects whose classes are already known, and based on the key for the given method, the value of k is calculated on which class is the most numerous among them. Each object has a finite number of attributes (dimensions). It is assumed that there is a certain set of objects with an already existing classification [7].

The next method for solving the classification tasks is the decision tree, which is used in the field of statistics and data analysis for predictive models.

The tree structure contains the following elements: “leaves” and “branches”. On the edges (“branches”) of decision trees, attributes are written, on which the target function depends, in the "leaves" the values of the target function are written, while in other nodes there are attributes that distinguish the cases. To classify a new case, we must go down the tree to the letter and give the corresponding value. Similar decision trees are widely used in intelligent data analysis. The goal is to create a model that predicts the value of the target variable based on multiple input variables [7].

Each leaf represents the value of the target variable, changed in the course of movement from the root to the letter. Each internal node corresponds to one of the input variables. A tree can also be "studied" by dividing the output sets of variables into subsets that are based on the testing of attribute values. This process is repeated on each of the received subsets. Recursion ends when the subset in the node has the same value as the target variable, so it does not add value to the predictions. The process of going from top to bottom, TDIDT, is an example of an absorbing "greedy" algorithm, and is by far the most widespread decision tree for data, but this is not the only one possible strategy.

The decision trees used in Data Mining are of two main types:

- Analysis of the classification tree when the predicted result is a class to which the data belongs;
- Regression analysis of a tree when the predicted result can be considered as a valid number (e.g. house price, or length of stay of a patient in a hospital) [8].

In the context of the current task, we are interested in the first type of decision tree for solving classification issues.

2.3 Artificial Neural Networks in Solving Classification Problems

Artificial neural networks can also be used to solve classification problems. An artificial neural network is a network of simple elements called neurons that receive input, change their internal state (activation) according to this input, and produce an output that is dependent on input and excite. The network is formed by connecting the outputs of certain neurons with inputs of other neurons with the formation of a directed weighted graph. Scales, as well as functions that calculate excitement, can change with the process called learning, which is guided by the rule of learning [7].

Components of the artificial neural network:

1. Neurons

The neuron with the label j, which receives input $p_j(t)$ from the neuronal predecessors, consists of the following components:

- Activation $a_j(t)$, which depends on the discrete time parameter;
- The threshold θ_j (for binary neuron), which remains unchanged, if it does not change the learning function;
- Activation functions f, which calculates the new activation at the given time t+1 from $a_j(t)$, θ_j and the network input $p_j(t)$, giving as a result the relation (10). The function is applied to all layers except the last one (where the output function is applied). Each intermediate connection has its own activation function.

$$a_j(t+1) = f(a_j(t), p_j(t), \theta_j) \quad (10)$$

— Output functions f_{out}, which calculates the exit activation: (11)

$$o_j(t) = f_{out}(a_j(t)) \quad (11)$$

The output function is often just the same function. The input neuron has no predecessors, but serves as the login interface for the entire network. Similarly, the exit neuron has no successors, and thus serves as an interface for the output for the entire network.

2. Connections and weights

The network consists of connections, each of which transmits the output of the neuron i to the input of the neuron j. In other words, i is the precursor (parent) of j, and the j is the successor (child) of i. Each such connection is assigned w_{ij} weight.

3. Distribution Functions

The distribution function calculates the input $p_j(t)$ to the neuron j from the outputs of $o_i(t)$ of the precursor neurons and usually has the form: (12)

$$p_j(t) = \sum_i o_i(t) w_{ij} \quad (12)$$

4. The rule of training

Training rule is a rule or algorithm that changes the parameters of the neural network so that the given input to the network produces a suitable output. This learning process usually involves changing the weights and thresholds of the network variables [7].

There are three main paradigms of learning, each of which corresponds to a particular learning objective. They are guided learning, spontaneous learning, and training with reinforcement [7]. We are interested in the first paradigm, because it is used to solve classification problems.

Guided learning uses a set of examples of pairs (x, y), $x \in X$, $y \in Y$, and has the purpose of finding a function (13) in a permitted class of functions that corresponds to these examples.

$$f : X \to Y \quad (13)$$

In other words, we want to display a reflection on which this data hints; the cost function is connected to the discrepancy between our reflection and the data, and it implicitly contains a priori knowledge of the subject domain. The tasks that fit into the guided learning paradigm are pattern recognition (also known as classification) and regression (also known as approximation of functions). A guided learning paradigm is also applicable to sequential data (for example, to the recognition of manual writing, speech and gestures). It can be seen as learning with a “teacher” in the form of a function that provides a constant feedback on the quality of the solutions obtained so far.

3 Practical Example of the Transaction Analysis and Fraud Detection Using Machine Learning

3.1 Overview and Description of the Transaction Database

To investigate this problem and find a solution, a database [9] of the payment system with transactional accounts was obtained. The database reflects transactions executed within 2 days, generally containing 284,807 transactions, of which 492 are fraud (0.172%). The dataset was gathered by Worldline and ULB (Université Libre de Bruxelles) and prepared by them using various approaches: their private software algorithms, manual testing, customers' feedback. That resulted into the merged dataset. The database consists only of numerical data. For confidentiality, the field of the database is anonymized. Because of this, it is not possible to specify a description of one or another peculiarity for which the field corresponds, and to give a more precise description of the data from an economic point of view.

All 28 parameters (V1, V2, ..., V28) were obtained using the main component method – principal component analysis method – a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing n observations) are an uncorrelated orthogonal basis set.

The only 2 fields that have not been transformed are "time" and "quantity". The "time" value shows the number of seconds that passed between this transaction and the first transaction. The "quantity" field shows the amount of money that went through the transaction.

All other fields have no marks or legend because of security and privacy reasons. The bank decided to not share what exactly these fields are, giving only their transformed numerical values.

The data set is very unbalanced, since the target class – fraudulent transactions – is only 0.17% of all transactions (Figure 1). If we use them to construct models, we will probably get a lot of false classifications due to overtraining of the model. The resulted model will assume that the transaction is likely to be a verified one, since almost all of the data set consists of such transactions.

3.2 Initial Analysis of the Transaction Database

We need to create a balanced subset of data with the same frequency of fraudulent and verified transactions, which will help further algorithms to show more accurate results.

What will be a subset of data? In our case, this will be a dataset with a ratio of 50/50 verified and fraudulent operations. The number of fraudulent and normal operations will be the same.

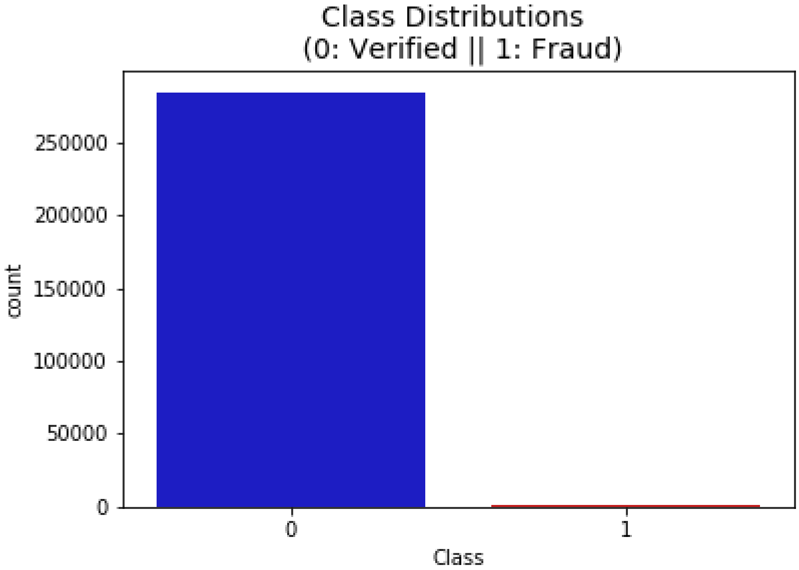

Fig. 1. Distribution of the initial transactions database by classes.

Why create a subset of data? We found that the initial set of data is very unbalanced. Its use can create the following problems:

- Overtraining. Since almost all records are verified, our model will empirically assign almost every transaction as non-fraudulent.
- Wrong correlation. Although we do not know what exactly corresponds to the “V” field, it will definitely be useful to understand how each of them affects the target function. Again, having an unbalanced set of data, the correlation matrix will be fuzzy and shifted toward non-fraud transactions [8].

Before applying random subsampling to the training set of data, we must divide the initial set of data into the training set and test set. Applying data balancing techniques (over-sampling or sub-sampling) should be done only on a training set of data in order to create a model, but the model testing should be done on the initial dataset.

In the next step, we will apply the technique of random over-sampling, which is about removing those entries from the set of data, which count is bigger. Thus, we achieve a ratio of 50/50 by excluding verified transactions (Figure 2).

Correlation matrixes are the basis for understanding the data. It is interesting for us to understand which arguments significantly affect the classification of the transaction. Particularly indicative is matrix comparison for balanced and unbalanced data sets (Figure 3).

Correlation matrix analysis:

- negative correlation: V10, V12, V14, V17. The smaller the value of these variables is, the more likely the transaction will be fraudulent.

- positive correlation: V2, V4, V11, V19. The larger the value of the variable is, the more likely the operation is fraudulent [8].

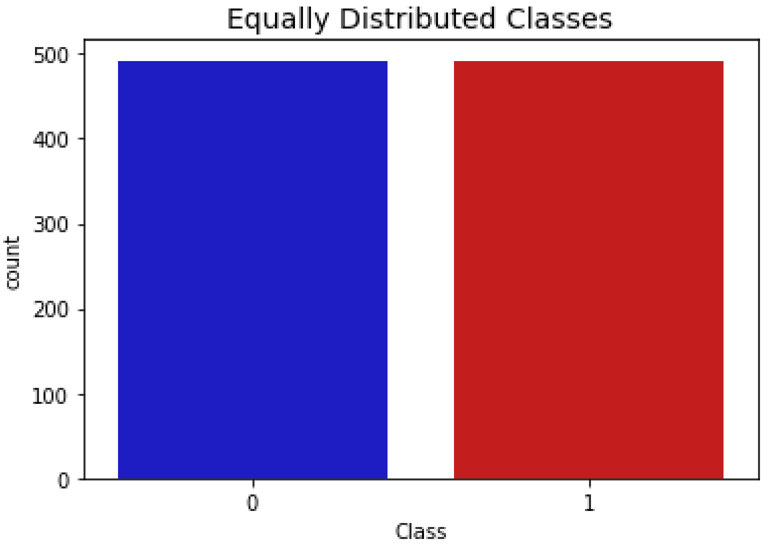

Fig. 2. Histogram of equally distributed classes after sub-sampling.

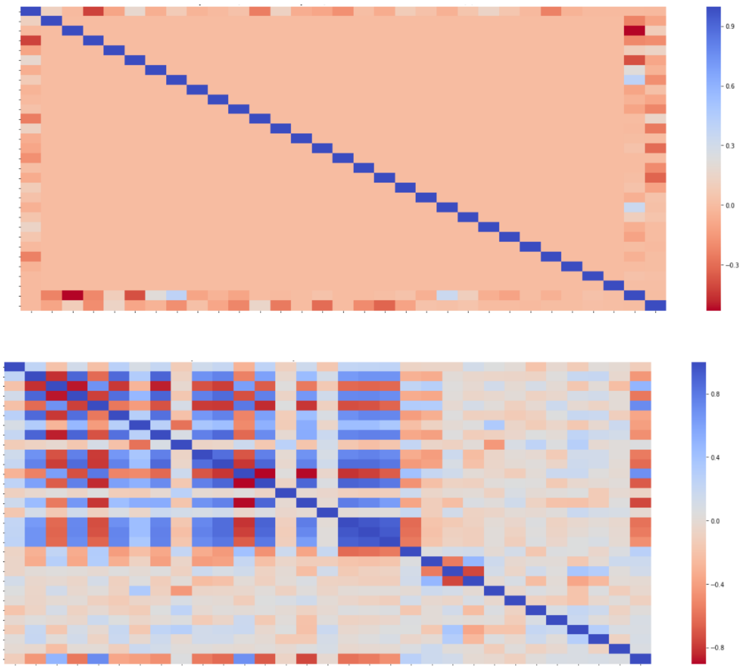

Fig. 3. Correlation matrixes of unbalanced (top) and balanced (bottom) data.

3.3 Creation and Evaluation of the Fraud Detection Classifiers

Before we begin, we need to divide our data into training and test subsets.

Of course, computing of large volumes of data and deducting the result, and, most importantly, high-speed computing, requires the use of computing machines. In practice, there are many tools and technologies for data processing, but the most popular are Python and R. What language to use is completely up to a user, the mathematical and statistical methods described above are implemented in both environments. In the given work will work in Python [10], but all the same techniques and methods are implemented in R.

We will use such libraries [11]:

- Pandas – for easier data processing;
- Matplotlib – for visualization;
- NumPy and SciPy – for scientific calculations;
- Seaborn – for visualization of statistical data;
- Sklearn – machine learning library;
- Tensorflow – machine learning library.

For each classifier, we build a model and find its accuracy [12].

After lets analyze and compare learning curves for all 4 models (Figure 4 – Figure 7):

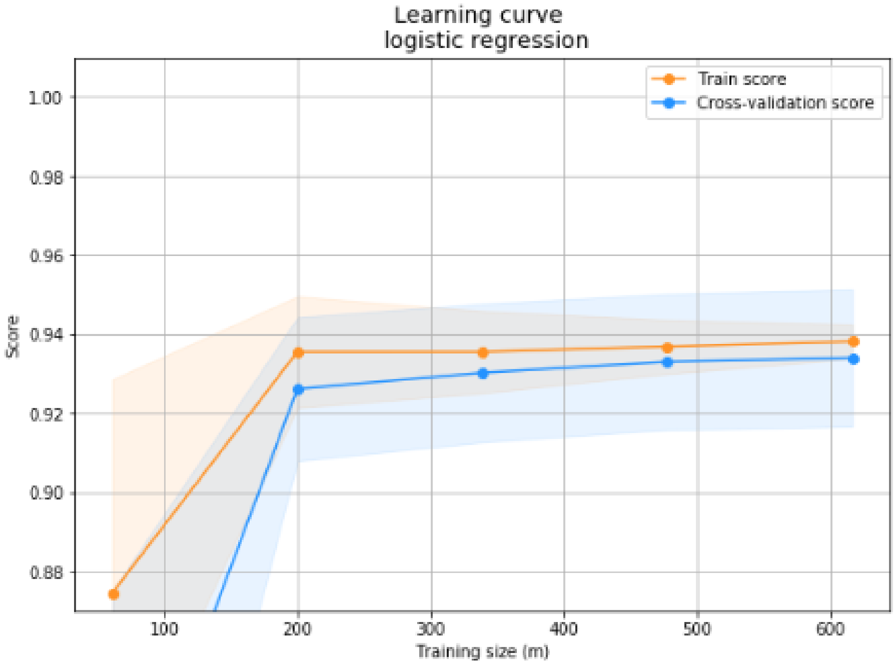

Fig. 4. Logistic regression learning curve.

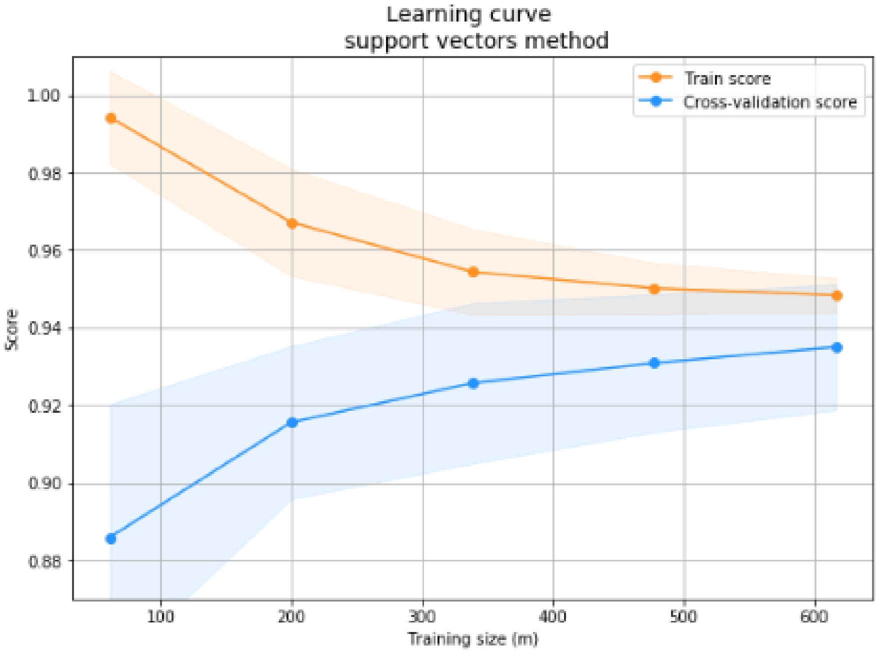

Fig. 5. Support vectors learning curve.

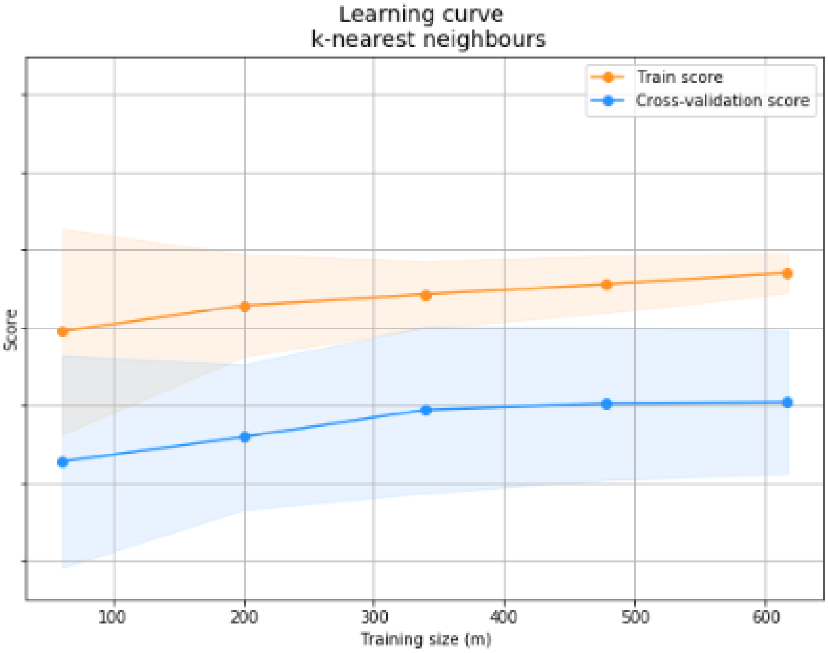

Fig. 6. K-nearest neighbours learning curve.

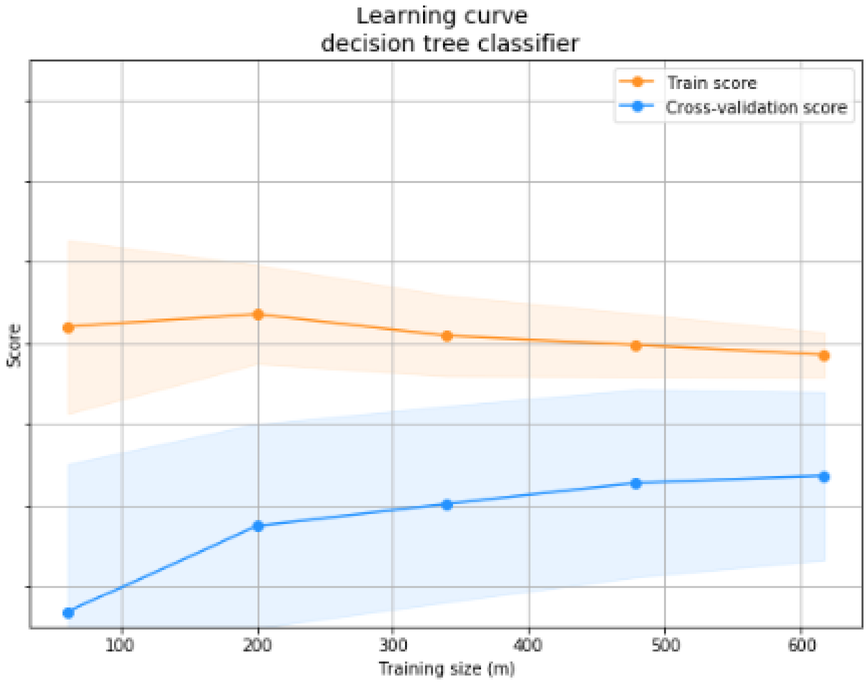

Fig. 7. Decision tree classifier learning curve.

Logistic regression showed the best accuracy with an estimate of 94%. This is a training result that was obtained from an assessment of how precisely the model determines fraud in the training sample. For a more accurate result, check the resulting models on the test sample (remember that this is still a balanced sample, so the result will still be inaccurate).

As we see from the obtained results, the logistic regression method was best demonstrated with a result of 94% on the training sample and 93.52% on the test sample (the best result was evaluated as the maximum arithmetic mean of the data of 2 indicators [13]). The method of k-nearest neighbors and the method of support vectors also showed a fairly precise result, and the support vectors method showed even better results on the test sample than the logistic regression – 93.78%.

For a more detailed demonstration of the results, we output a confusion matrix [14] for logistic regression method. In the upper left and lower right squares (yellow) the correct results are placed, in other squares (black) wrong results are places.

As we see from Figure 8, this method correctly detected 96 + 89 = 185 transactions. The other 8 transactions fell into inappropriate groups, so they were not predicted correctly. Remember, the above results was obtained on sub-sampled test dataset.

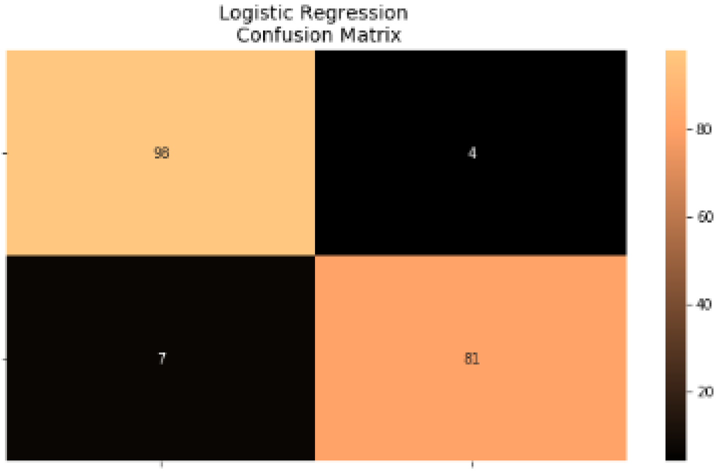

Fig. 8. Logistic regression results' confusion matrix.

3.4 Fraud Detection Using Neural Networks

To create the neural network, the same Python software package, based on the Tensorflow, was used.

The structure of the neural network: a simple model that consists of one input layer, one hidden layer of 32 nodes, and one output layer that can take one of two possible values: 0 or 1.

We will supervise two studies of the neural network: the first by means of sub-sampling, and the other by means of over-sampling. In the first case, we will narrow our data to a ratio of 50/50, so we will randomly drop a significant portion of the verified transactions. During the over-sampling, we will expand our data by adding new records of fraudulent data that will be generated basing on the existing records of the fraudulent data.

To supervise the neural network, 20 iterations were performed on the corresponding data set. After performing the neural network training, we evaluate it on the original data set and compare the results between the neural networks itself and the best classifiers.

As we see from Figure 9, the neural network on the sub-sampled data classified a significant part of the verified transactions (Y-axis) in the class of fraudulent, but only 1 fraudulent transaction passed. In general, the score of the neural network was 93.1%.

Over-sampling (data expansion) showed the best result (Figure 10) among both neural networks and all models in general, having demonstrated 99.9% of the correct classifications. However, it should be noted that 24 fraudulent transactions have passed, and therefore the percentage of blocked fraudulent transactions is lower.

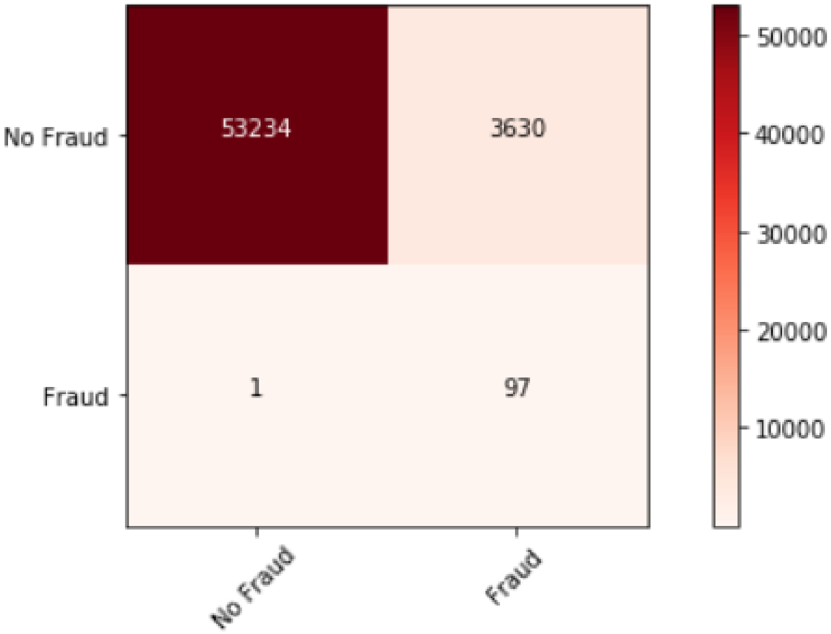

Fig. 9. Confusion matrix for the neural network, trained on sub-sampling.

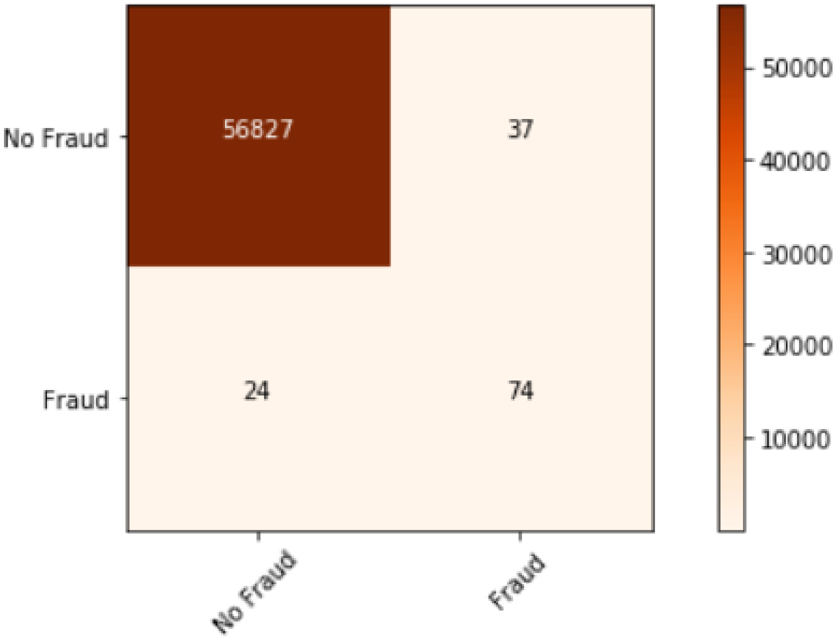

Fig. 10. Confusion matrix for the neural network, trained on over-sampling.

4 Conclusion

The logistic regression reaches up to 94% of the correct classifications, while the neural network on the sub-sampled data shows a result of 93.1%, and over-sampled data shows as much as 99.9%, but misses a significant amount of fraudulent operations.

On the one hand, the accuracy of the neural network on the over-sampling is higher, but on the other hand, it misses most of the fraudulent operations, although it better classifies the verified ones. Logistic regression showed average accuracy, but also missed a significant part of fraudulent transactions. Although the neural network on the sub-sampling showed the worst overall result of 93.1%, but it prevented the biggest amount of the fraud transactions.

In general, the use of one or another model depends on the specific situation: whether clients are ready sometimes get denial of the transaction, but to be sure that their funds will not be obtained by fraud, or they are more interested in easy of use, and security is not that important.

References

1. Ajvazyan, S.A., Buxshtaber, V.M., Enyukov, Y.S., Meshalkyn, L.D.: Prykladnaya statystyka – klassyfykacyya i snyzhenye razmernosty (Applied Statistics – Classification and Reduction of Dimensionality). Finansy i statistika, Moscow (1989)
2. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Heidelberg (2009)
3. Lebichot, B., Le Borgne, Y.-A.: Deep-Learning Domain Adaptation Techniques for Credit Cards Fraud Detection. In: Oneto, L., Navarin, N., Sperduti, A., Anguita, D. (eds.) Recent Advances in Big Data and Deep Learning, pp. 78–88. Springer, New York (2019)
4. Caelen, O., Smirnov, E.N.: Improving Card Fraud Detection Through Suspicious Pattern Discovery. In: Benferhat, S., Tabia, K., Ali, M. (eds.) Advances in Artificial Intelligence: From Theory to Practice, pp. 181–190. Springer, New York (2017)
5. Pozzolo, A.D., Caelen, O., Bontempi, G., Johnson, R.A.: Calibrating Probability with Undersampling for Unbalanced Classification. Paper presented at the 2015 IEEE Symposium Series on Computational Intelligence, Cape Town, South Africa, 7-10 December 2015
6. Chernyak, O.I., Komashko, O.V., Stavyckyj, A.V., Bazhenova, O.V.: Ekonometryka (Econometrics). Vydavnycho-polihrafichnyi tsentr "Kyivskyi universytet", Kyiv (2010)
7. Mitchell, T.: Machine Learning. McGraw Hill, New York (1997)
8. Shlezynher, M., Hlavach, V.: Desyat lekcyj po statystycheskomu y strukturnomu raspoznavanyyu (Ten lectures on statistical and structural recognition). Naukova Dumka, Kyiv (2004)
9. Transactions database. ULB, Belgium. http://mlg.ulb.ac.be/ (2016). Accessed 2 June 2018
10. McKinney, W.: Python for Data Analysis. O'Reilly Media, Sebastopol (2016)
11. Idris, I.: Python Data Analysis Cookbook. Packt Publishing, Birmingham (2016)
12. Michie, D., Spiegelhalter, D.J.: Machine Learning, Neural and Statistical Classification. University of Leeds, Leeds (1994)
13. Nilsson, N.J.: Introduction To Machine Learning. Stanford University, Stanford (1997)
14. Miller, J.D.: Big Data Visualization. Packt Publishing, Birmingham (2017)

15. Shpyrko, V., Koval, B.: Fraud detection models and payment transactions analysis using machine learning. SHS Web of Conferences. **65**, 02002 (2019). doi:10.1051/shsconf/20196502002