Expert Systems With Applications 134 (2019) 28–35

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Direct marketing campaigns in retail banking with the use of deep learning and random forests

Piotr Ładyżyński[a,*], Kamil Żbikowski[b], Piotr Gawrysiak[b]

[a] Faculty of Mathematics and Computer Science, Warsaw University of Technology, Koszykowa 75, Warsaw 00-662, Poland
[b] Institute of Computer Science, Warsaw University of Technology Nowowiejska 15/19, Warsaw 00-665, Poland

ARTICLE INFO

Article history:
Received 27 September 2018
Revised 3 April 2019
Accepted 14 May 2019
Available online 15 May 2019

Keywords:
Consumer credit
Retail banking
Direct marketing
Marketing campaigns
Database marketing
Random forest
Deep learning
Deep belief networks
Data mining
Time series
Feature selection
Boruta algorithm

ABSTRACT

Credit products are a crucial part of business of banks and other financial institutions. A novel approach based on time series of customer's data representation for predicting willingness to take a personal loan is shown. Proposed testing procedure based on moving window allows detection of complex, sequential, time based dependencies between particular transactions. Moreover, this approach reduces noise by eliminating irrelevant dependencies that would occur due to the lack of time dimension analysis.

The system for identifying customers interested in credit products, based on classification with random forests and deep neural networks is proposed. The promising results of empirical studies prove that the system is able to extract significant patterns from customers historical transfer and transactional data and predict credit purchase likelihood. Our approach, including the testing method, is not limited to banking sector and can be easily transferred and implemented as a general purpose direct marketing campaign system.

1. Introduction

For contemporary financial institutions, such as large retail banks, marketing is a key part of business operations. Most of the products - such as credits - that can be offered by such an organization present relatively small possibility of differentiation from those available from competitors. Therefore a success - or a failure - of a modern retail bank is greatly dependent on a quality and effectiveness of its marketing campaigns. Launching a successful marketing campaign in a very competitive market segment, such as retail banking, is certainly a challenging task. Subsequent campaigns in such market conditions are less and less effective, as customers, constantly exposed to different kinds of advertisements, learned to ignore it. One of the solutions, that is increasingly gaining popularity in recent years, is personalized, direct marketing. Such approach allows to customize the product properties per customer basis, thus increasing campaign efficiency and reducing costs (see e.g. Hossein Javaheri, 2008). Retail banks are gathering tremendous amounts of information related to their clients' activity (in a form of clients' transaction data), which makes such personalization an intuitive extension of mass campaigns.

In our approach, described in this paper, we propose an innovative, time series based method of generating personalized credit product campaigns tailored for individual customers. As we describe it in the following section, the current state of the art approach to backtesting marketing campaigns is to simply launch an A/B test for a particular period and gather test's results afterwards. This approach is very sensitive to seasonal effects. Moreover, it does not provide any built-in mechanism for attributing particular product purchases to certain marketing efforts. Another weakness of current methods is treating each example from a training set equally without identifying and making use of the time dimension.

In our approach, applying a moving window procedure allows to detect the best moment for approaching a client with an appropriate offer. A vivid example (being an extension to traditional retail banking services) that explains the difference would be analysis of purchasing flight ticket by a client. A situation in which the purchase has been made yesterday is quite different versus one made two weeks ago. In the first scenario, one could be very inter-

* Corresponding author.
E-mail addresses: p.ladyzynski@mini.pw.edu.pl (P. Ładyżyński), k.zbikowski@ii.pw.edu.pl (K. Żbikowski), p.gawrysiak@ii.pw.edu.pl (P. Gawrysiak).

https://doi.org/10.1016/j.eswa.2019.05.020
0957-4174/

ested in accommodations' options, while in the second, the appropriate product would be an insurance or on-site car rental services.

Another example that intuitively explains the importance of considering the relative time intervals between transactions, comprises two sets of the same transactional events spread differently over time for two different customers. The clients may have quite a different willingness to use credit products. We can imagine a situation in which the same income for two clients is followed by purchasing e.g. electronics producs with the only difference being a time span between income and spending. The obvious difference here, that can be implicitly detected, is propensity to save connected with well planned purchase that is uncorrelated with income for the wider time span and quite the opposite situation for a shorter one.

In this paper we describe a novel system for approaching customers with direct marketing campaigns based on features extracted from a large database of transactional data from one of the largest retail banks in Poland. In our approach each client's data is mapped into multidimensional time series. Each observation $X_t = (X_t^{(1)}, \ldots, X_t^{(k)})$ in such time series contains information describing client's behavior in the certain time window $(t - l, t)$. Such representation provides us with an intuitive training set construction strategy, as we can classify each observation in the time series into two classes: client is or is not interested in a credit product purchase.

Moreover, due to the proposed time series data representation we are able to precisely define performance metrics of the system, as the marketing campaigns' success ratio also includes time factor.

2. Previous works

The use of machine learning in direct marketing campaigns has been widely discussed in literature (Bose & Chen, 2009), however most of it relates to the retail industry and e-commerce and only few to banking products (Sing'oei & Wang, 2013). The common architecture of such systems is well stated in Hossein Javaheri (2008) and in general consists of the following: Data extraction from customers' databases, features extraction, response variable construction and, finally, training the classifier. However, most of works use "static" approach and perform model training and validation only for customer's features calculated for one specified moment. As the customers preferences vary in time and change seasonally, such an approach may result in an information loss or even in an improper performance metrics for the model.

A good inspiration for analyzing time-variant patterns in customers data can be found in Min and Han (2005). Authors tried to find patterns for improving recommender system. This task is to some extent similar to detecting the best moment for a direct marketing camapaign call.

The use of historical transactional data is widely used in previous works (Changchien, Lee, & Hsu, 2004; Liao & Chen, 2004; Rossi, McCulloch, & Allenby, 1996; Weng & Liu, 2004). Using transactional data is considered to be an advantage that allows to personalize particular campaign in order to improve cross-selling and up-selling efficiency.

The problem stated in Sing'oei and Wang (2013), is quite similar to our one: while having customer data from one of the Portuguese banks database, authors are trying to predict if a customer is interested in subscribing to the given product: a term deposit. However, the description of the proposed system is not very thorough - there is no precise feature definition in the training set (SPSS - is used only as a black box). The experimental results are also unclear - there is only a lift chart of the classifier with no legend and no business context.

Kim, Kim, and Lee (2003), focused on introducing a new algorithm for constructing classifiers' ensembles using genetic algorithms based approach. Web data from one of the leading e-commerce companies in Korea is used to build a model for the customer's purchase behavior prediction. The proposed model predicts a purchase of a given product. No time frames defining classifier success are mentioned (classifier predicts that a customer will buy certain product but it is not stated how long we should wait for the action of the customer to mark the event as the success). There is no definition of,purchase propensity" which authors try to predict.

In Kaefer, Heilman, and Ramenofsky (2005), authors try to predict chances of customer's potential loyalty to a certain brand. The trained classifier is used as a customer targeting system in a direct marketing promoting this brand. Kim and Street (2004), propose a hybrid system for customer targeting which performs feature selection with a genetic algorithm for a neural network. The system is used for predicting car insurance purchase likelihood. In Piersma and Jonker (2004), authors construct a stochastic Markov model which estimates optimal frequency of marketing contacts with a given customer.

3. Multidimensional time series representation of a customer behavior

Financial institutions collect a lot of data about their customers. The data, which describes their consumer habits is especially valuable and interesting in terms of predicting demand for certain groups of products like consumer credits, insurance etc. For example, banks are in possession of all transactional history of their customers. Mapping customer's transactions into certain predefined statistics may provide analysts with valuable insights of life habits of the customers and of their preferences. Such insights are a powerful tool for marketing campaigns, enabling targeting of certain groups of products right into groups of customers who are really interested. Having the transactional history we are able to infer various informations about the customer:

- Earnings
- Savings
- Does he or she spends a lot for certain groups of products (i.e. luxury goods, travelling)?
- What are the customer's preferences in saving? Does he/she spend everything or save the major part of earnings?
- Does the customer have a weakness for gadgets?
- Does he have children, a spouse or a beloved pet?
- Does he own a company?
- Does he enjoy nightlife?

Organizations try to construct effective marketing campaigns by extracting features like these mentioned above and then hiring analysts for discovering patterns from the extracted features. As one may imagine, the pattern leading to effective targeting of the customers interested in consumer credit purchase may look as follows:

> "*Client has been buying a new Apple gadget before Christmas for past three years but this year his income dropped by 40% in comparison to previous three years average.*"

In this paper we extract features describing customer behavior based on three main sets of data available in banks:

- Account balance history,
- money transfers history,
- credit and debit card transactions history.

For the purpose of feature definition clarity let us introduce some operators. Suppose that B_t denotes the account balance of the given customer at the end of the day t. Let (B_t) denote a dis-

crete time series for $t \in [0, T]$. In our notation (B_t) is customer account balance history.

Let

$$A^{min}_{[t_p,t_k]} = min(B_{t_p}, B_{t_p+1}, \ldots, B_{t_k-1}, B_{t_k}), \tag{1}$$

denotes minimum value of customer account balance in the interval $t \in [t_p, t_k]$ and

$$A^{max}_{[t_p,t_k]} = max(B_{t_p}, B_{t_p+1}, \ldots, B_{t_k-1}, B_{t_k}), \tag{2}$$

denotes maximum value of customer account balance in the same interval. Let

$$MA^k_t = \frac{1}{k} \sum_{i=t-k+1}^{t} B_i \tag{3}$$

denote moving average of the customer account balance for the moment t of length k, whereas exponential moving average of length k we define as

$$EMA^k_t = \alpha B_t + (1-\alpha) EMA^k_{t-1}, \tag{4}$$

where $EMA^k_0 = B_0$ and $\alpha = \frac{2}{k+1}$. Note that k affects only the smoothing factor α. Moreover let

$$MED_{[t_p,t_k]}, \tag{5}$$

denote the median of the observations: $\{B_{t_p}, B_{t_p+1}, \ldots, B_{t_k-1}, B_{t_k}\}$.

Apart from defining some characteristics of the customer balance history we can investigate his financial and consumption preferences basing on his credit or debit card transaction history and money transfers history. Both categories could be divided into certain groups. Credit or debit card transactions could be grouped on the basis of transaction MCC codes which are sent to banks by transfer managers such as Visa or MasterCard. In our work we grouped customers transactions into 19 categories. Let us s define by $\mathcal{T}$ set of transaction categories: *airlines, car hire, clothing, entertainment, petrol, food, retailers, home, department stores, household stores, health, hotel, insurance, motoring, other retailers, services, utilities, supermarket, travel, total amount of all transactions*. By

$$\mathfrak{T}^{\tau}_{[t_p,t_k]}, \tag{6}$$

we denote sum of all transactions in the category τ from the set $\mathcal{T}$ computed for the time interval $[t_p, t_k]$.

We tried to categorize money transfers in the similar way. Nevertheless, money transfers are more difficult to categorize than card transactions because there are no MCC codes in this case. One of the solutions is to perform categorization based on money transfer titles but this requires the manual construction of multiple heuristic rules or application of natural language processing algorithms. In our work, for the case of simplicity, we used categories derived from financial product type to which money transfer belongs. Lets define by $\mathcal{G}$ set of 22 categories of money transfers : *Automatic credit card repayment, capitalization of interest, premium insurance, repayment of the installment, debt early repayment, interests, express transfer charge, card owner charge, SWIFT transfer charge, sms information charge, insurance premium, interests tax, provision for international transfer, atm withdrawal provision, express transfer, incoming SEPA transfer, incoming internal transfer, transfer from the credit card, atm withdrawal, atm deposit, total amount of incoming transfers, total amount of outgoing transfers*. Similarly to the transaction aggregates $\mathfrak{T}^{\tau}_{[t_p,t_k]}$ we define

$$\mathfrak{G}^{g}_{[t_p,t_k]}, \tag{7}$$

as the sum of all money transfers in the category g from the set $\mathcal{G}$ computed for the time interval $[t_p, t_k]$.

Following the definitions above we are able to define some detailed features which describe the customer of the bank at the moment t. We analyze 90 day interval of customer history to compute the metrics for a particular moment from: $(t-89, t-88, \ldots, t)$. By

$$\begin{aligned} L_t = (&B_t, B_{t-7}, B_{t-14}, \ldots, B_{t-12\cdot 7}, \\ &A^{min}_{[t,t-7]}, A^{min}_{[t,t-14]}, A^{min}_{[t,t-21]}, A^{min}_{[t,t-28]}, \\ &A^{min}_{[t-29,t-35]}, A^{min}_{[t-29,t-42]}, A^{min}_{[t-29,t-49]}, A^{min}_{[t-29,t-56]}, \\ &A^{min}_{[t-57,t-63]}, A^{min}_{[t-57,t-70]}, A^{min}_{[t-57,t-77]}, A^{min}_{[t-57,t-84]}, \\ &A^{max}_{[t,t-7]}, A^{max}_{[t,t-14]}, A^{max}_{[t,t-21]}, A^{max}_{[t,t-28]}, \\ &A^{max}_{[t-29,t-35]}, A^{max}_{[t-29,t-42]}, A^{max}_{[t-29,t-49]}, A^{max}_{[t-29,t-56]}, \\ &A^{max}_{[t-57,t-63]}, A^{max}_{[t-57,t-70]}, A^{max}_{[t-57,t-77]}, A^{max}_{[t-57,t-84]}, \\ &EMA^{30}_t, EMA^{30}_{t-7}, EMA^{30}_{t-14}, EMA^{30}_{t-21}, \\ &EMA^{30}_{t-28}, EMA^{30}_{t-35}, EMA^{30}_{t-42}, EMA^{30}_{t-49}, \\ &MA^7_t, MA^{14}_t, MA^{21}_t, MA^{28}_t, \\ &MA^7_{t-28}, MA^{14}_{t-28}, MA^{21}_{t-28}, MA^{28}_{t-28}, \\ &MED_{[t,t-30]}, MED_{[t-30,t-60]}, MED_{[t-60,t-90]}, MED_{[t,t-89]}\big), \end{aligned} \tag{8}$$

we define aggregates obtained from account balance history (B_t). $A^{\min}_{[t_p,t_k]}$ and $A^{\max}_{[t_p,t_k]}$ captures customer's minimum and maximum balances in various time horizons whereas MA^k_t and EMA^{30}_t captures balances dynamics.

By

$$\begin{aligned} R_t = \big(&\mathfrak{T}^{\tau_1}_{[t,t-7]}, \mathfrak{T}^{\tau_1}_{[t,t-14]}, \mathfrak{T}^{\tau_1}_{[t,t-21]}, \mathfrak{T}^{\tau_1}_{[t,t-28]}, \mathfrak{T}^{\tau_1}_{[t,t-88]} \\ &\mathfrak{T}^{\tau_1}_{[t-30,t-37]}, \mathfrak{T}^{\tau_1}_{[t-30,t-44]}, \mathfrak{T}^{\tau_1}_{[t-30,t-51]}, \mathfrak{T}^{\tau_1}_{[t-30,t-58]}, \\ &\mathfrak{T}^{\tau_1}_{[t-60,t-67]}, \mathfrak{T}^{\tau_1}_{[t-60,t-74]}, \mathfrak{T}^{\tau_1}_{[t-60,t-81]}, \mathfrak{T}^{\tau_1}_{[t-60,t-88]}, \\ &\vdots \\ &\mathfrak{T}^{\tau_i}_{[t,t-7]}, \mathfrak{T}^{\tau_i}_{[t,t-14]}, \mathfrak{T}^{\tau_i}_{[t,t-21]}, \mathfrak{T}^{\tau_i}_{[t,t-28]}, \mathfrak{T}^{\tau_i}_{[t,t-88]} \\ &\mathfrak{T}^{\tau_i}_{[t-30,t-37]}, \mathfrak{T}^{\tau_i}_{[t-30,t-44]}, \mathfrak{T}^{\tau_i}_{[t-30,t-51]}, \mathfrak{T}^{\tau_i}_{[t-30,t-58]}, \\ &\mathfrak{T}^{\tau_i}_{[t-60,t-67]}, \mathfrak{T}^{\tau_i}_{[t-60,t-74]}, \mathfrak{T}^{\tau_i}_{[t-60,t-81]}, \mathfrak{T}^{\tau_i}_{[t-60,t-88]}, \\ &\vdots \\ &\mathfrak{T}^{\tau_{19}}_{[t,t-7]}, \mathfrak{T}^{\tau_{19}}_{[t,t-14]}, \mathfrak{T}^{\tau_{19}}_{[t,t-21]}, \mathfrak{T}^{\tau_{19}}_{[t,t-28]}, \mathfrak{T}^{\tau_{19}}_{[t,t-88]} \\ &\mathfrak{T}^{\tau_{19}}_{[t-30,t-37]}, \mathfrak{T}^{\tau_{19}}_{[t-30,t-44]}, \mathfrak{T}^{\tau_{19}}_{[t-30,t-51]}, \mathfrak{T}^{\tau_{19}}_{[t-30,t-58]}, \\ &\mathfrak{T}^{\tau_{19}}_{[t-60,t-67]}, \mathfrak{T}^{\tau_{19}}_{[t-60,t-74]}, \mathfrak{T}^{\tau_{19}}_{[t-60,t-81]}, \mathfrak{T}^{\tau_{19}}_{[t-60,t-88]}\big), \end{aligned} \tag{9}$$

we define aggregates derived from the sum of money amounts in debit or credit card transactions, $\tau_i \in \mathcal{T}$. Following the definition of $\mathcal{T}$ and $\mathfrak{T}^{\tau}_{[t_p,t_k]}$, $\mathfrak{T}^{\tau_1}_{[t,t-14]}$ denotes the amount of money spent by the given customer on airlines tickets in the last two weeks, whereas $\mathfrak{T}^{\tau_{19}}_{[t,t-14]}$ denotes the total amount of money spent by the customer in the last two weeks.

With R_t we are able to capture some of customer's consumption preferences. In the same way we are able to capture some information from money transfers:

$$\begin{aligned} G_t = \big(&\mathfrak{G}^{g_1}_{[t,t-7]}, \mathfrak{G}^{g_1}_{[t,t-14]}, \mathfrak{G}^{g_1}_{[t,t-21]}, \mathfrak{G}^{g_1}_{[t,t-28]}, \mathfrak{G}^{g_1}_{[t,t-88]} \\ &\mathfrak{G}^{g_1}_{[t-30,t-37]}, \mathfrak{G}^{g_1}_{[t-30,t-44]}, \mathfrak{G}^{g_1}_{[t-30,t-51]}, \mathfrak{G}^{g_1}_{[t-30,t-58]}, \\ &\mathfrak{G}^{g_1}_{[t-60,t-67]}, \mathfrak{G}^{g_1}_{[t-60,t-74]}, \mathfrak{G}^{g_1}_{[t-60,t-81]}, \mathfrak{G}^{g_1}_{[t-60,t-88]}, \\ &\vdots \\ &\mathfrak{G}^{g_i}_{[t,t-7]}, \mathfrak{G}^{g_i}_{[t,t-14]}, \mathfrak{G}^{g_i}_{[t,t-21]}, \mathfrak{G}^{g_i}_{[t,t-28]}, \mathfrak{G}^{g_i}_{[t,t-88]} \\ &\mathfrak{G}^{g_i}_{[t-30,t-37]}, \mathfrak{G}^{g_i}_{[t-30,t-44]}, \mathfrak{G}^{g_i}_{[t-30,t-51]}, \mathfrak{G}^{g_i}_{[t-30,t-58]}, \\ &\mathfrak{G}^{g_i}_{[t-60,t-67]}, \mathfrak{G}^{g_i}_{[t-60,t-74]}, \mathfrak{G}^{g_i}_{[t-60,t-81]}, \mathfrak{G}^{g_i}_{[t-60,t-88]}, \\ &\vdots \end{aligned}$$

$$\mathfrak{G}^{g_{24}}_{[t,t-7]}, \mathfrak{G}^{g_{24}}_{[t,t-14]}, \mathfrak{G}^{g_{24}}_{[t,t-21]}, \mathfrak{G}^{g_{24}}_{[t,t-28]}, \mathfrak{G}^{g_{24}}_{[t,t-88]}$$
$$\mathfrak{G}^{g_{24}}_{[t-30,t-37]}, \mathfrak{G}^{g_{24}}_{[t-30,t-44]}, \mathfrak{G}^{g_{24}}_{[t-30,t-51]}, \mathfrak{G}^{g_{24}}_{[t-30,t-58]},$$
$$\mathfrak{G}^{g_{24}}_{[t-60,t-67]}, \mathfrak{G}^{g_{24}}_{[t-60,t-74]}, \mathfrak{G}^{g_{24}}_{[t-60,t-81]}, \mathfrak{G}^{g_{24}}_{[t-60,t-88]}\Big), \tag{10}$$

where $g_i \in \mathcal{G}$.

Having all the definitions above we are able to define multidimensional time series which describes the customer of the bank at the moment t:

$$C_t = (L_t, R_t, G_t). \tag{11}$$

Having that $dim(L_t) = 57$, $dim(R_t) = 19 \cdot 13 = 247$ and $dim(L_t) = 22 \cdot 13 = 286$, the number of explanatory variables in the problem is equal to $dim(C_t) = 616$.

4. Time series approach - moving window testing vs AB testing

A common practice in validating marketing campaigns is the AB testing method. Suppose that we would like to check if after receiving a marketing phone call from a car dealer a client is more likely to buy a car. We divide clients into two sets: A - clients to be called, B - clients not to be called and compare how making a phone call affects number of sold cars. Lets explain why our time series based approach has an advantage over AB testing procedure. Let's

$$L_m = \begin{cases} 0.05, & \text{if } m \in \{Oct, Nov, Dec\} \\ 0.025, & \text{otherwise,} \end{cases} \tag{12}$$

is a probability of buying a car within 30 days after phone call where m denotes month. Similarly lets

$$T_m = \begin{cases} 0.012, & \text{if } m \in \{Oct, Nov, Dec\} \\ 0.01, & \text{otherwise,} \end{cases} \tag{13}$$

denotes a probability of buying a car within 30 days without phone call from marketer. If the company conducts AB test in April it will come to conclusion that performing a call increases sales by $L_{Apr}/T_{Apr} = 250\%$. However if we perform time series similar to that described in the next section we would get the following increase in sales: $\frac{\sum L_m/12}{\sum T_m/12} = 298\%$. The result is quite different. As we can see moving window test which averages results over time gives more accurate outcome and does not depend on seasonal factors. In direct marketing campaigns simulator described in the following section we decided to follow analogical moving window testing method.

5. Direct marketing campaigns simulator - framework for model performance evaluation

The Eq. (11) defines given customer features at the moment t. In order to construct full training set for the supervised learning we should also define the response variable. Let

$$S^j_t = \begin{cases} 1, & \text{if exists } t_0 \in M^j \text{ that } t_0 \geq t \text{ and } |t_0 - t| \leq 30 \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

where M^j is the set of moments when customer j applies for a loan. S^j_t defines credit sentiment for the customer j at the moment t. If the customer applies for a credit at the moment t_0 his credit sentiment is high (equal to 1) in the interval $[t_0 - 30, t_0]$. Supposing that J is the set of bank customers we construct complete set for training machine learning classifier:

$$F_J = (S^j_t; C^j_t), \tag{15}$$

where $j \in J$ and $t \in [0, T]$. S^j_t defines the class and C^j_t defines 616 features. Note that for the sake of simplicity we have omitted index j in the Eq. (11).

In our scenario we analyzed 800000 customers and a period of two years (the cardinality of T is equal to $2 \cdot 365 = 730$) so the full dimension of the problem is quite high - $800000 \cdot 730 = 584 \cdot 10^6$ of learning examples. This dimension is too high even for the distributed machine learning framework Apache Spark used in our problem for feature extraction and model training. Therefore we have divided J into two sets: set J_1, $card(J_1) = 5000$ and set J_2, $card(J_2) = 795000$. $J_1 \cup J_2 = J$ and $J_1 \cap J_2 = \varnothing$. By A we denote training set obtained from set

$$F_{J_1} = (S^j_t; C^j_t), \tag{16}$$

$j \in J_1$, $t \in [0, T]$, by sampling 100000 elements. By B we define test set as the whole set of customers time series realizations from set J_2:

$$B = F_{J_2} = (S^j_t; C^j_t), \tag{17}$$

where $j \in J_2$ and $t \in [0, T]$. Note that $card(A) = 100000$ and $card(B) = card(F_{J_2}) = 5000 \cdot 730 = 3650000$. Also note that testing the system with the cross-validation method performed for the set A is serious methodical error as random division of set A into training and validation sample may result in having in validation sample observation $(S^j_{t_1}; C^j_{t_1})$ and in training sample observation $(S^j_{t_2}; C^j_{t_2})$ where $t_1 < t_2$. In such approach the classifier will be able to see the future events for the same customer j as the later observation t_2 was sampled into training sample.

As our system was designed with the purpose of increasing success ratio in direct marketing campaigns we have to define performance measures to evaluate our classifier. Typically, direct marketing campaigns are scored with the following scenario: call center consultants call the defined group of k customers of which l takes some action after call (purchase credit). Action should be taken in the specified time horizon t_c after the call. For example, if the customer purchases a credit after a quite short period after the call, one may suppose that the marketers triggered his decision. The ratio $\frac{l}{k}$ defines marketing campaign success ratio. Note that after a call the customer is blocked for further calls for a specified time t_b, as being called every day would be annoying.

In order to compute campaign evaluation metrics for test set B we implemented direct marketing campaign simulator (Algorithm 1) which simulates strategy above and computes campaign success metrics.

Having sets M^{call}_j (see Algorithm 1) and M_j we are able to define marketing campaign performance metrics. Let

$$Prec_{camp} = \frac{\sum_{j \in J_2} card(\{z | z \in M^{call}_j \quad \text{and} \quad \exists_{x \in M_j} : x - z \leq 30\})}{\sum_{j \in J_2} card(M^{call}_j)}, \tag{18}$$

denote marketing campaign precision. The numerator is equal to the number of calls which triggered credit purchase (number of calls after which the customer purchased a credit within 30 days), denominator is equal to the sum of all calls made by call center consultants. Similarly,

$$Recall_{camp} = \frac{\sum_{j \in J_2} card(\{x | x \in M_j \quad \text{and} \quad \exists_{z \in M^{call}_j} : x - z \leq 30\})}{\sum_{j \in J_2} card(M_j)}, \tag{19}$$

denotes marketing campaign recall. The numerator is equal to the number of credits triggered by call center agents (number of credits from the test set purchased within 30 days after call from the call center) and denominator is equal to the sum of all credits taken by customers.

Algorithm 1 Direct marketing campaign simulator.

CAMPAIGN PARAMETERS: t_c - number of days on which a customer should take an action after a call to regard his action as a success, t_b - number of days when customer is blocked for further calling after a call.In our case $t_b = t_c = 30$.
INPUT DATA: test set B defined by (17), trained classifier $\mathcal{C}$

0: $t_j := 0 \quad$ for $\quad j \in J_2$
0: $M_j^{call} := \varnothing \quad$ for $\quad j \in J_2$
0: $\{M_j^{call}$ - is the set of moments when the customer j is called by a call center consultant $\}$
for $j \in J_2$ **do**
  while $t_j < T$ **do**
    $\{\mathcal{C}(\cdot) : \mathbb{R}^n \to \{0, 1\}$ - classification function of classifier $\mathcal{C}$. For example for the feature vector x,$\mathcal{C}(x) = 1$ if classifier assigns class 1 to the vector x.$\}$
    if $\mathcal{C}(C_t^j) = 1$ **then**
      $t_j := t_j + 30$
      $M_j^{call} := M_j^{call} \cup \{t_j\}$
    else
      $t_j := t_j + 1$
    end if
  end while
end for

6. System architecture for discovering credit purchase likelihood

While considering databases of large banks we are faced not only with the data modelling problems, but with data volume issues as well. For example 10^6 of active bank customers generates $10 \cdot 10^9$ of card transactions yearly, which is quite a high amount of data for extracting valuable features and building machine learning models. With the amount of data of such magnitude, relational databases become inefficient both in terms of hardware required and of licensing. Thus, a lot of financial institutions facing big data problems moved already towards Apache Hadoop platform. In our work we used two different frameworks for distributed machine learning: Apache Spark Framework and H2O framework.

Apache Spark is a fast and general-purpose cluster computing system (ApacheSpark, 2018; Zaharia, 2016). It introduced Resilient Distributed Datasets (RDD) as an abstraction that aggregate data and allow to efficiently run computations over cluster of nodes. Efficiency is achieved mainly by preserving results of intermediary computations in-memory and through creating direct acyclic graph for computations that significantly reduces necessity of shuffling the data.

H2O (H2O.ai, 2018) is an open source machine learning platform. It is easily deployed and provides a pile of distributed implementations of machine learning algorithms. Moreover H2O has a great integration with R platform.

The high level system architecture for credit purchase likelihood prediction is depicted in the Fig. 1. In the first stage, data is migrated from the data warehouse (relational database) to the Hadoop cluster and than Apache Spark jobs perform feature extraction and construction of both the training (16) and the test set (17). In the third stage we trained various models (Random Forests, CART with the use of Apache Spark and Deep Belief Networks with the use of H2O) and score them in the last stage with our campaign simulator. In our work we compared three different machine learning algorithms: classification trees - CART (Breiman, 2017), random forests (Friedman, Hastie, & Tibshirani, 2001) and deep belief networks - DBN (Hinton, Osindero, & Teh, 2006). Classification and regression trees are wide known machine learning algorithms based on idea of minimizing statistical dispersion while going from the root to the leaves in the tree. Random forests are extension of decision trees with the variant of bagging technique. Deep belief networks are neural networks with multiple hidden layers consisting of stochastic binary neurons (the probability of turning on a stochastic binary neuron is determined by the weighed input from other units). The main idea of training such a network is to treat the entire network as stacked restricted Boltzman machines and pre-train weights with unsupervised learning algorithm before the application of gradient descent. Such approach enables the model to extract a deep hierarchical representation of the training data and in some cases outperform other state of art classifiers (Hinton et al., 2006).

7. Feature selection with Boruta algorithm

Dimensionality problem is a well known obstacle to applying statistical learning algorithms. For instance, it has been noted that increasing number of features does not lead to better accuracy for classification problems assuming a finite training data set. It was show in Hughes (1968) and in Trunk (1979) that there exist some optimal point in terms of number of features after which accuracy decreases. It is sometimes referred as “Curse of dimensionality” or, after the author of Hughes (1968), “Hughes phenomenon”.

Dimensionality reduction basically falls into two categories: feature extraction and feature selection Tang, Alelyani, and Liu (2014). A representative example of the former is Principal Component Analysis. It transforms original feature space into new space with less dimensions. However, new features are difficult to interpret and interpretability is especially important in financial models that are under supervision of regulatory authorities. The latter is vast amount of methods that select a subset of original features. In Nilsson, Peña, Björkegren, and Tegnér (2007) the distinction between minimal-optimal and all-relevant subsets is being made. The latter subset is much harder to compute.

The minimal-optimal dimensionality reduction basically requires to select such a subset of features that would result in no or minimal decline in classification performance compared to number of dimensions reduced. It promotes features that are characterized by a high signal to noise ratio. All-relevant dimensionality reduction has its roots in gene expression data analysis (Nilsson et al., 2007). The objective there is to find all features that are relevant to the target variable, not only those supreme in terms of signal to noise ratio. It is especially important in disease detection where data is highly unbalanced and majority of dataset happen to be of a negative class. Similar, although of much less importance for humans lives, is the problem of credit scoring when number of positive cases i.e. credit defaults are a minority of the whole dataset dedicated for the purpose of learning. Due to this fact we decided to use a method that seem to be more suitable for such a case.

We used the Boruta algorithm for a feature selection purpose. The algorithm that was described in Kursa, Rudnicki et al. (2010) applies the following sequence:

1. Duplicate each of the features for training set A creating artificial shadow attributes.
2. Shuffle all the artificial variables in order to remove dependence with the response variable.
3. Select all the features that had better maximum Z score (MZSA) from all the Z scores of shadow attributes computed by running Random Forest classifier over extended dataset.
4. Remove features with importance significantly lower than importance for MZSA resulted from running two-sided test of equality with MZSA.

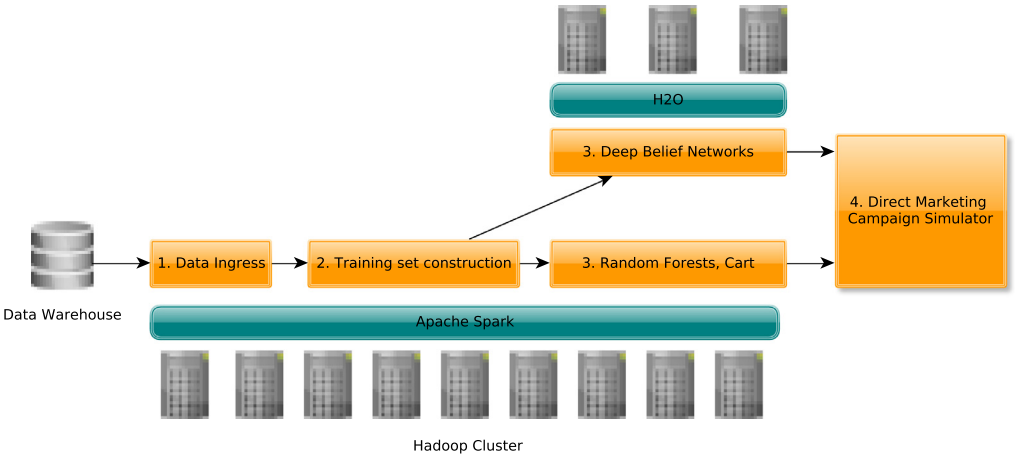

Fig. 1. The high level system architecture for credit purchase likelihood prediction.

Table 1
System performance metrics obtained with random forests classifier.

	Model	x KG	Dimension	$Prec_{camp}$	$Recall_{camp}$
1	*RandomForest Full*	1	100000 x 616	15.56%	2.37%
2	*RandomForest Full*	2	100000 x 616	13.11%	11.11%
3	*RandomForest Full*	3	100000 x 616	12.45%	18.20%
4	*RandomForest Boruta*	1	100000 x 164	14.84%	7.04%
5	*RandomForest Boruta*	3	100000 x 164	15.00%	10.63%
6	*RandomForest Boruta*	5	100000 x 164	12.80%	14.38%

Table 2
System performance metrics obtained with pruned CART classifier.

	Model	x KG	Dimension	$Prec_{camp}$	$Recall_{camp}$
1	*CART Boruta*	1	100000 x 616	9.56%	23.17%
2	*CART Boruta*	3	100000 x 616	9.01%	67.27%

5. Select features with importance significantly higher than importance for MZSA resulted from running two-sided test of equality with MZSA.
6. Remove all shadow attributes and repeat the procedure until all features are assigned or predetermined number of iterations is being reached.

From the 616 input features from training set (16) Boruta algorithm indicated 164 as important ones. The exemplary important features: total amount of transactions in the last three months, total petrol expenditure in the last month and last three months, total entertainment expenditure in the last two weeks, last month and last three months, total amount of incoming money transfers in the last two weeks, last month and last three months, maximum and minimum account balance in the last two weeks and last month, median account balance in the last three months and others.

8. Experimental results

Tables from 1 to 4 describes system performance metrics computed with various machine learning algorithms. Best models was marked with light grey colour. Models 5 (Table 1) and 13 (Table 4) won in terms of campaign precision whereas models 2 (Table 2), 7 (Table 3) and 11 (Table 4) have still quite good precision and better campaign recall. Direct marketing campaign managed by model 11 (Table 4) detected 28.62% of all credits in the test set and 13.60% of the simulated calls made by consultants resulted in credit purchase.

Results from (Table 1) clearly show that using Boruta algorithm increases precision but significantly decreases recall. Comparing various networks architectures (Table 3), (Table 4) we can deduce that adding regularization parameter $l1$, and setting fixed number of epochs equal to 20 increases precision. Moreover best deep learning model 13 (Table 4) beats best random forest model 5 (Table 1) both in term of precision and recall. However differences are not very significant.

Note that network architecture in Tables 3 and 4 follows notation form H2O R package (see *h2o.deeplearning()* method documentation)[1]. For example *c*(1500, 1400, 1300) denotes deep neural network with three hidden layers having 1500, 1400, 1300 neurons in each layer accordingly.

To choose the best classifier for given business problem one should consider the loss function based on false negatives and false positives and score all the classifiers with this function. In retail banking false negative (not calling a client who is interested in a loan) is times more expensive than false positive (calling a client who is not interested). Bearing this in mind we used the simplest model 2 (Table 2) as our production classifier. It offers us great recall, decent precision and is most efficient in terms of computing power.

9. Conclusions

Despite recent legislation, such as European GDPR directive, that aim to limit profiling of customer behaviour, analysis of pur-

[1] https://cran.r-project.org/web/packages/h2o/h2o.pdf.

Table 3
System performance metrics obtained with deep belief networks implemented in H2O framework.

	Model	x KG	Dimension	Network Architecture	$Prec_{camp}$	$Recall_{camp}$
1	*DeepLearning Boruta*	5	100000 x 164	c(200, 180, 160, 140, 120, 100)	5.30%	83%
2	*DeepLearning Boruta*	5	100000 x 164	c(200)	5.20%	93%
3	*DeepLearning Full*	5	100000 x 616	c(1500)	7.94%	78.27%
4	*DeepLearning Full*	5	100000 x 616	c(1500,1400)	7.52%	64.48%
5	*DeepLearning Full*	5	100000 x 616	c(1500, 1400, 1300)	8.05%	66.70%
6	*DeepLearning Full*	5	100000 x 616	c(1500, 1400, 1300, 1200)	7.28%	63.44%
7	*DeepLearning Full*	2	100000 x 616	c(1300)	9.27%	64.82%
8	*DeepLearning Full*	1	100000 x 616	c(1300)	7.03%	34%

Table 4
System performance metrics obtained with deep belief networks implemented in H2O framework with $l1$ regularization parameter added.

	Model	x KG	Dimension	Network Architecture	$Prec_{camp}$	$Recall_{camp}$
9	*DeepLearning Full*	2	100000 x 616	c(300), l1=1e-4	11.01%	31.38%
10	*DeepLearning Full*	2	100000 x 616	c(300), l1=1e-3, epochs=20	13.00%	25.86%
11	*DeepLearning Full*	2	100000 x 616	c(300, 280),l1=1e-3, epochs=20	13.60%	28.62%
12	*DeepLearning Full*	2	100000 x 616	c(1300),l1=1e-4, epochs=10, input_dropout_ratio=0.2	11.86%	33.10%
13	*DeepLearning Full*	2	100000 x 616	(100,95), l1=1e-3, epochs=20	16.00%	11.00%

chase patterns will remain one of the most important tools used in direct marketing. Unfortunately in most cases this is not a trivial task, as demonstrated by a relatively poor accuracy of existing solutions. In this paper we described a novel system, that incorporates temporal information to analysis of customer behaviour, in order to estimate not only the best type of personalized marketing proposal directed to a specific client, but also a most appropriate moment, when the client should be contacted.

The effectiveness of our approach was verified in a real-world financial institution. However, our findings clearly indicate that there is much room for improvement and further research. One of the possible directions of future research can be extending data used for training machine learning models. Enriching the data can lead to constructing more adequate campaigns and most importantly to designing much more “fitted” product offer that would be even better perceived by the clients. This approach, tempting from the perspective of data analysis, is unfortunately limited because in most cases in order to obtain any additional knowledge, a client must provide additional marketing permissions.

One of weaknesses of presented study is that it relies on binary classification, that do not ”dig deeper” in terms of whether a particular product is well suited for a particular customer. Is is partially justified as most banking products are very much alike. This does not apply however to products offered by many recently established fin-tech startups plenty of new finacial products are available and one can easily imagine exploring past purchases and trying to obtain knowledge from those transactions. This is are very well suited for models using collaborative filtering approach (Min & Han, 2005). We might focus on analyzing responsiveness based on specific product offers in combination with moving window testing procedure It could improve accuracy of predicting best time to call a client while being able to significantly enhance client’s experience because of better suited product offer.

Results of presented study is not only limited to a banking sector. In fact, applications in other areas may be more beneficial as banking sector is very homogeneous in terms of product offered.

From the technical perspective, we analyzed the data with a quite constrained time window, capturing only 90 days worth of customer actions. Albeit extending this window would be computationally expensive, it might be worth doing in order to capture deeper insights about individual customer behaviour. This should also allow us to analyze the customers’ sensitivity to repeated call center contacts. Finally, with the wider rollout of our system in the bank, we should be able to assess its effectiveness as part of a multichannel marketing strategy of the financial institution.

Conflict of interest

The project described in the paper was concluded at the end of the experimental phase and thus no commercial, or production deployment of the methods presented in the paper ever took place.

Credit authorship contribution statement

Piotr Ładyżyński: Conceptualization, Methodology, Software, Writing - original draft, Supervision. **Kamil Żbikowski:** Conceptualization, Methodology, Writing - review & editing, Validation. **Piotr Gawrysiak:** Writing - review & editing, Validation, Methodology.

References

ApacheSpark (2018). *Apache spark framework*. http://spark.apache.org/

Bose, I., & Chen, X. (2009). Quantitative models for direct marketing: A review from systems perspective. *European Journal of Operational Research, 195*(1), 1–16. doi:10.1016/j.ejor.2008.04.006.

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Changchien, S. W., Lee, C.-F., & Hsu, Y.-J. (2004). On-line personalized sales promotion in electronic commerce. *Expert Systems with Applications, 27*(1), 35–52.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning*: 1. Springer series in statistics New York, NY, USA:.

H2O.ai (2018). *H2O Framework*. https://www.h2o.ai/.

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554.

Hossein Javaheri, S. (2008). Response modeling in direct marketing: a data mining based approach for target selection.

Hughes, G. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory, 14*(1), 55–63.

Kaefer, F., Heilman, C. M., & Ramenofsky, S. D. (2005). A neural network application to consumer classification to improve the timing of direct marketing activities. *Computers & Operations Research, 32*(10), 2595–2615.

Kim, E., Kim, W., & Lee, Y. (2003). Combination of multiple classifiers for the customer's purchase behavior prediction. *Decision Support Systems, 34*(2), 167–175.

Kim, Y., & Street, W. N. (2004). An intelligent system for customer targeting: A data mining approach. *Decision Support Systems, 37*(2), 215–228.

Kursa, M. B., Rudnicki, W. R., et al. (2010). Feature selection with the Boruta package. *Journal Statistics Software, 36*(11), 1–13.

Liao, S.-H., & Chen, Y.-J. (2004). Mining customer knowledge for electronic catalog marketing. *Expert Systems with Applications, 27*(4), 521–532.

Min, S.-H., & Han, I. (2005). Detection of the customer time-variant pattern for improving recommender systems. *Expert Systems with Applications, 28*(2), 189–199.

Nilsson, R., Peña, J. M., Björkegren, J., & Tegnér, J. (2007). Consistent feature selection for pattern recognition in polynomial time. *Journal of Machine Learning Research, 8*(Mar), 589–612.

Piersma, N., & Jonker, J.-J. (2004). Determining the optimal direct mailing frequency. *European Journal of Operational Research, 158*(1), 173–182.
Rossi, P. E., McCulloch, R. E., & Allenby, G. M. (1996). The value of purchase history data in target marketing. *Marketing Science, 15*(4), 321–340.
Sing'oei, L., & Wang, J. (2013). Data mining framework for direct marketing: A case study of bank marketing. *International Journal of Computer Science Issues (IJCSI), 10*(2 Part 2), 198.
Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. *Data Classification: Algorithms and Applications*, 37.
Trunk, G. V. (1979). A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (3), 306–307.
Weng, S.-S., & Liu, M.-J. (2004). Feature-based recommendations for one-to-one marketing. *Expert Systems with Applications, 26*(4), 493–508.
Zaharia, M. (2016). *An architecture for fast and general data processing on large clusters*. Morgan & Claypool.