КОМП'ЮТЕРНІ НАУКИ Й ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 005.8:316.422
DOI https://doi.org/10.17721/ISTS.2020.4.3-10

V. V. Morozov, orcid.org/0000-0001-7946-0832,
knumvv@gmail.com
O. O. Mezentseva, orcid.org/0000-0002-8430-4022,
olga.mezentseva.fit@gmail.com
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

USE TRAINING NEURAL NETWORKS FOR PREDICTING PRODUCT DEVELOPMENT OF IT PROJECT

The state of development of innovations in Ukraine is characterized by an increase in development on the basis of start-up projects with the use as a project product of information systems of varying complexity. The article analyzes the weak survivability of the results of start-up projects. The conclusion on the need to predict the stages of development of IT project products based on the analysis of the processes of interaction of users (customers) with the information system (product). In this article, components of the model of forecasting of IT products development of innovative start-up projects are considered based on the analysis of formed datasets of the interactions of prospective clients. We offered the algorithm of formation of initial datasets based on Customer Journey Map (CJM), which are the tool of fixing of events of the interaction of clients with the system. Examples of models of analogues of clients' travel maps are given, which are the basis for recording and analyzing interactions. This fact is the basis for the formation of appropriate data sets of large dimension. As a mechanism for processing big data sets and building strategies for IT products development, it is proposed to use a learning neural network. Mathematical models for further modeling and analysis of the obtained results are built. We used a simple linear regression analysis to model the relationship between a single explanatory variable and a continuous response variable (dependent variable). An exploratory data analysis method was applied to the available data to find repetitive patterns and anomalies. In the course of the research, we constructed a model of linear regression implementation using the gradient optimisation approach. The linear models of the scikit-learn library for the regression task were also applied, and the stabilisation regression method was implemented. Modelling and analysis of the obtained results were carried out, which showed greater efficiency over the extended life cycle of IT project products.

Keywords: *start-up; information interaction; customer journey map; forecasting.*

1. INTRODUCTION

At the current stage of development of information systems engineering technologies, innovations in the form of start-up projects are becoming increasingly important [1]. Such projects are often formalised as separate commercial enterprises to obtain funding and financial profitability. Often, such companies are represented with the implementation of SaaS (Software as a Service) distribution model [2] or B2B (Business to Business) transactions [3]. Such business models have problems with long sales cycles, as the decision of the client is collective and depends on many factors. The success of the product development and, accordingly, the efficiency of the enterprise depends on the indicator of the quality of customer service at information interaction with the IT system. However, to define directions of development of such products, it is not simple. For this purpose, it is necessary to form datasets during some period, characterising points of preferable information interaction of different kinds of clients, and also to spend analytics and to predict directions of such development.

The method of information interaction that has proved to be effective in many projects can be used to solve these problems. It is based on the analysis of the "journey" (interaction with IT product) of the prospective client (Customer Journey Map – CJM) [4, 5]. Also, it is necessary to combine millions of events, which will provide the necessary analysis of the impact on the customers' journeys and will determine the future content of projects to develop such IT products.

By analysing [6] millions of real-time interaction data points, it will be possible to find the most critical customer movement events in the system [7] and prioritise those opportunities that have a significant impact on business objectives, such as increasing revenue, reducing customer churn, improving customer service [8] and developing innovations in the company.

However, sufficient constructive and technological complexity of construction of programs of development of complex IT products demands the use of the project approach [9–12] wherein processes of creation and development of such systems methods and information technologies of project management are applied.

Experience shows that for analytical processing of considerable volumes of the information of interaction of clients with IT system use of methods of artificial intelligence (AI) will have a significant influence on the efficiency of development programs (of projects) creation.

The search for optimal variants of distribution of resources at the preparation of development programs in innovative projects can reduce terms of performance of project tasks and, as a result, decrease their cost. At the same time, predicting the impact of interaction with clients [13] on the variants of development programs in such projects is a multidimensional task that can be solved using technologies of modern artificial neural networks [14]. Besides, it is necessary to take into account numerous changes [15] that affect various parameters of innovative projects and significantly affect the negative results of their implementation.

Thus, consideration of the possibilities of experimental use of the Customer Journey Map and artificial neural networks in the research of improving the quality of interaction of numerous clients based on optimal development programs of start-up projects is an actual challenge.

2. ANALYSIS OF RECENT RESEARCH AND PUBLICATIONS

Application of the project approach to start-up development and effective use of products of these projects at the creation of modern IT were considered in research papers [9–16]. In [9], new models of management of innovations in projects based on the use of Markov processes are offered. This approach can be used for further development of start-up projects. However, there is no assessment of the value of products for customers of such projects. The [10] contains a description of the competency models that can be applied to Start-up project teams by the customer. However, there are no methods of interaction between users and the customer's team here.

The [11] provides a detailed description of the process approach for managing any kind of projects, and the specifics of innovation project management are considered separately. However, the processes of management of the creation and development of a product are not given. At the same time, [12] describes integrated methodologies for organisational and product management of development projects. However, methods of an estimation of the efficiency of communication interaction are insufficiently described. In [13, 14, 16] approaches to the management of complex IT projects based on proactive (anticipatory, predictive) management with the use of methods of artificial intelligence, in particular, with the use of artificial neural networks, have been proposed. However, studies based on interaction with clients have not been conducted.

The use of methods to assess customer interactions in SaaS and B2B business models based on Customer Journey Map has been highlighted in [4, 17–19]. As noted in these papers, such developments should be based on new tools to assess the productivity (value) of customer interaction with the IT product as feedback. Here we can use well-proven customer journey models though it is not specified, how to process rather big data on numerous indicators analytically.

Then, considering aspects of application of intellectual tools for the analysis and forecasting based on the processing of a big data set from the interaction of clients, it is possible to consider papers [20–27]. In these papers, it is noted that the most acceptable for us, in terms of intellectual forecasting of the composition of programs for the development of complex IT products, is the use of trainable neural networks. Furthermore, it will be useful in our case to build a model of linear regression implementation [23] using the gradient optimisation approach [27].

The goal of the article is to justify and develop a model for predicting the composition of the program for the development of complex products of start-up projects using the analysis of customer interaction data based on the Customer Journey Map and the application of trainable neural networks.

The main objectives of this study are as follows:

1. Identify the interaction elements that form the basis of the Customer Journey Map to create survey datasets within one year.

2. Development of a model for forecasting the composition of development programs, taking into

account maximum customer loyalty and retention in their interaction with the IT system.

3. Development of algorithms for modelling the neural network training processes and setting up the forecasting processes.

3. MODEL DEVELOPMENT AND USE OF MODELLING METHOD

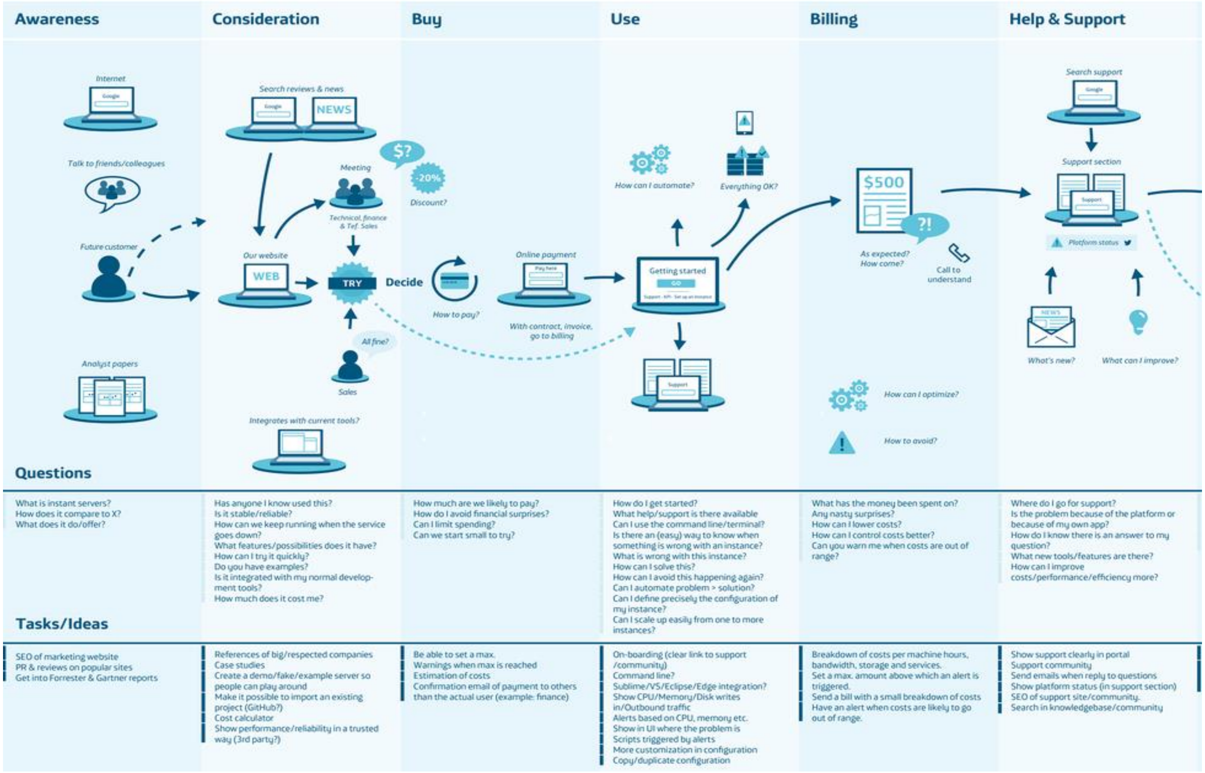
A. Constructing customer interaction models with IT product based on journey maps

As mentioned above, one of the effective tools to assess the quality of user interaction with the IT product is the customer journey map (CJM). It is a tool for visualising of the interaction of the consumer with a product or service. Creating a CJM is both a process of analysis and a method for generating ideas to improve a product or service.

CJM displays a time-bound interaction broken down into small components. The components of interactions refer both to the process (consumers' goals and objectives, their actions, expected results, problems and barriers preventing the transition to the next step, touchpoints, materials, tools, equipment, KPI from the business point of view, etc.) and to the psycho-emotional state of the consumer (thoughts, feelings, emotions).

This approach to the development of products characterized by multi-channel interaction is particularly useful, that is, for those cases where the customer and the product have many "touchpoints". The customer always has a sum of impressions of the interaction with the product through all available channels, so that even one negative experience is able to vilify the entire product in the eyes of the client and force him to cancel the subscription.

As an example, we can consider the CJM depicted in Fig. 1, which shows the customer's journey from awareness, consideration, purchase and further use of the "instant server" in a particular web application (IaaS distribution model) of the multinational telecommunications company ("Telefónica").



	Awareness	Consideration	Buy	Use	Billing	Help & Support
Questions	What is instant servers? How does it compare to X? What does it do/offer?	Has anyone I know used this? Is it stable/reliable? How can we keep running when the service goes down? What features/possibilities does it have? How can I try it quickly? Do you have examples? Is it integrated with my normal development tools? How much does it cost me?	How much are we likely to pay? How do I avoid financial surprises? Can I limit spending? Can we start small to try?	How do I get started? What help/support is there available Can I use the command line/terminal? Is there an (easy) way to know when something is wrong with an instance? What is wrong with this instance? How can I solve this? How can I avoid this happening again? Can I automate problem > solution? Can I define precisely the configuration of my instance? Can I scale up easily from one to more instances?	What has the money been spent on? Any nasty surprises? How can I lower costs? How can I control costs better? Can you warn me when costs are out of range?	Where do I go for support? Is the problem because of the platform or because of my own app? How do I know there is an answer to my question? What new tools/features are there? How can I improve costs/performance/efficiency more?
Tasks/Ideas	SEO of marketing website PR & reviews on popular sites Get into Forrester & Gartner reports	References of big/respected companies Case studies Create a demo/fake/example server so people can play around Make it possible to import an existing project (GitHub?) Cost calculator Show performance/reliability in a trusted way (3rd party?)	Be able to set a max. Warnings when max is reached Estimation of costs Confirmation email of payment to others than the actual user (example: finance)	On-boarding (clear link to support /community) Command line? Sublime/VS/Eclipse/Edge integration? Show CPU/Memory/Disk writes in/Outbound traffic Alerts based on CPU, memory etc. Show in UI where the problem is Scripts triggered by alerts More customization in configuration Copy/duplicate configuration	Breakdown of costs per machine hours, bandwidth, storage and services. Set a max. amount above which an alert is triggered. Send a bill with a small breakdown of costs Have an alert when costs are likely to go out of range.	Show support clearly in portal Support community Send emails when reply to questions Show platform status (in support section) SEO of support site/community. Search in knowledgebase/community

Fig. 1. Example of Customer Journey Map

B. Building predictive models using neural networks

Regression models are used to predict target variables on a continuous scale, which makes them useful for many scientific issues and information industry applications, such as understanding relationships between variables, assessing trends, or making forecasts.

One example of their application can be the prediction of company sales in the coming months:

$$y = w_0 + w_1 x, \tag{1}$$

where w_0 – weight, that is the intersection point of the Y-axis; w_1 – the explanatory variable coefficient. The most frequently used is multiple linear regression ***y:***

$$y = w_0 x_o + w_1 x_1 + \cdots + w_m x_m =$$
$$= \sum_{i=1}^{m} w_i x_i = w^T x\ , \qquad (2)$$

where w_0 is the intersection point of the Y-axis at; m – the number of regression members; w^T– the explanatory variable coefficient.

To determine the number of linear relationships between features, we will now create a correlation matrix.

The correlation coefficients are limited to the range [-1, 1]. Two attributes have, respectively, absolutely positive correlation if r = 1, no correlation if r = 0, and absolutely negative correlation if r = -1.

As mentioned earlier, Pearson's correlation coefficient can be calculated simply as the covariance between two attributes **x** and **y** – numerator, divided by the product of their standard deviations (denominator):

$$r = \frac{\sum_{i=1}^{n}[(x^i - \mu_i)(y^i - \mu_y)]}{\sqrt{\sum_{i=1}^{n}(x^i - \mu_x)^2}\sqrt{\sum_{i=1}^{n}(y^i - \mu_y)^2}} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}, \qquad (3)$$

where *n* – the number of attributes; μ – the empirical average of these attributes; σ – covariance between attributes.

As can be seen in the resulting figure, the correlation matrix provides us with another final diagram, which can help us select attributes based on their corresponding linear correlations (Fig. 2).

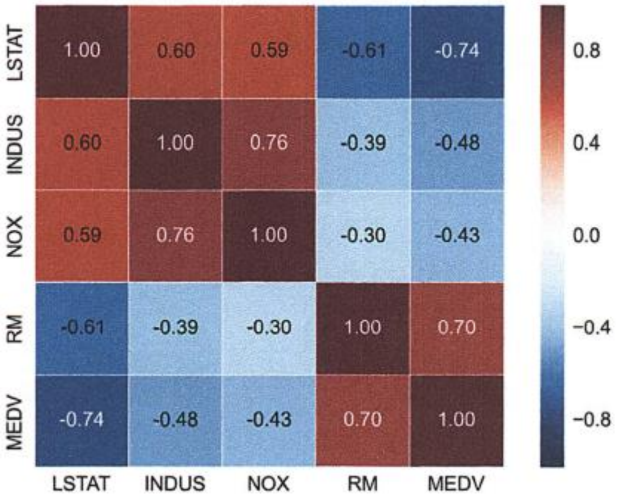


Fig. 2. An example of a correlation matrix

In order to fit a linear regression model, we are interested in those features that have a high correlation with our target variable MEDV. Looking at the above correlation matrix, we can see that our target variable MEDV shows the highest correlation with variable LSTAT (-0.74). The correlation between RM and MEDV is also relatively high (0.70). In the presence of a linear relationship between these two variables that we have observed in the dot matrix as an explanatory variable, the artificial neuron uses a linear activation function. Moreover, we have defined the JS(w) cost function that we have minimised for weight extraction thanks to optimisation algorithms such as gradient descent (GD) [23] and stochastic gradient descent (SGD) [24].

This cost function in ADALINE is the sum of squared errors (SSE). It is identical to the least squared value function (MLS), which we defined as follows:

$$JS(w) = \frac{1}{2}\sum_{i=1}^{n}\left(y^i - \hat{y}^i\right)^2, \qquad (4)$$

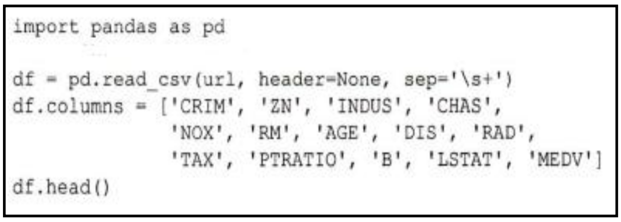
where $\hat{y}$ – it is a predicted value $\hat{y} = w^T x$; (note that the coefficient 1/2 is used simply for the convenience of obtaining an update rule for gradient descent).

In essence, linear regression on MLS can be understood as ADALINE without a single step function. As a result, we get continuous target values instead of class labels – 1 and 1.

4. EXPERIMENTATION

As our target variable, which we want to predict using one or more of these 51 explanatory variables, we will consider the tariffs for interaction with the system (a product of the start-up project) called MEDV.

Before we continue the exploration of this data set, we will put it into *DataFrame* data table of the *pandas* library [27] from the UCI repository (Fig. 3).



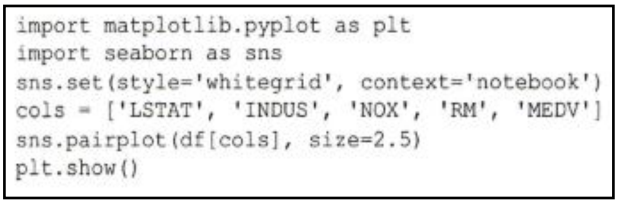
```
import pandas as pd

df = pd.read_csv(url, header=None, sep='\s+')
df.columns = ['CRIM', 'ZN', 'INDUS', 'CHAS',
              'NOX', 'RM', 'AGE', 'DIS', 'RAD',
              'TAX', 'PTRATIO', 'B', 'LSTAT', 'MEDV']
df.head()
```

Fig. 3. Downloading the source data

During the training of a linear regression model, it is not required that explanatory or target variables are distributed normally.

First, let us create a matrix of dotted charts that allows visualising pairwise correlations between different attributes in one place in this dataset. To prepare the matrix of point graphs, let's use the *pairplot* function from Python *seaborn library* [30], which is developed based on the *matplotlib* library for building statistical charts (Fig. 4).

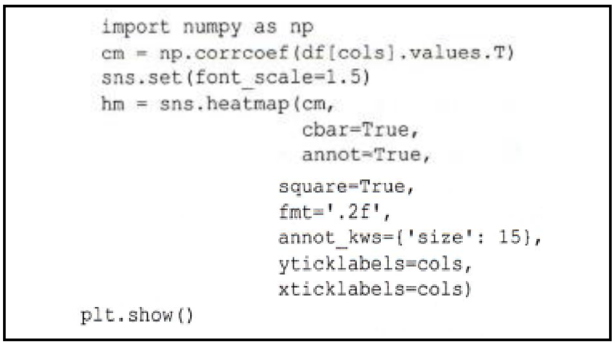
Using the *corrcoef* function of the NumPy library on five columns of attributes and the heatmap function of the seaborn library, we will get the necessary relationships between the initial indicators



```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style='whitegrid', context='notebook')
cols = ['LSTAT', 'INDUS', 'NOX', 'RM', 'MEDV']
sns.pairplot(df[cols], size=2.5)
plt.show()
```

Fig. 4. An example of a matrix of correlation links and point graphs

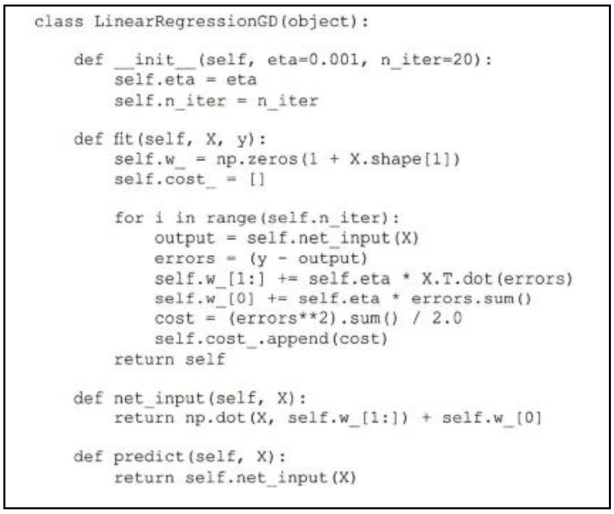
The following code is used to prepare an array of correlation matrices (Fig. 5).



```
   import numpy as np
   cm = np.corrcoef(df[cols].values.T)
   sns.set(font_scale=1.5)
   hm = sns.heatmap(cm,
                           cbar=True,
                           annot=True,
                         square=True,
                         fmt='.2f',
                         annot_kws={'size': 15},
                         yticklabels=cols,
                         xticklabels=cols)
plt.show()
```

Fig. 5. Example of preparing an array of correlation matrices in the form of a heatmap

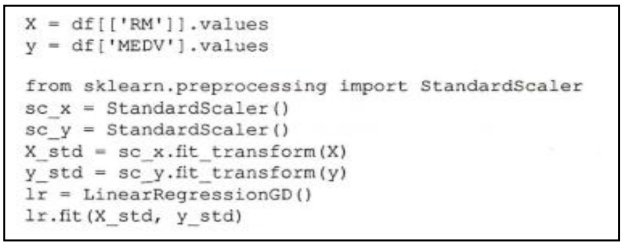
Linear regression on MLS can be understood as ADALINE without a single step function, resulting in continuous target values instead of class labels 1 and 1 (Fig. 6).



```
class LinearRegressionGD(object):

    def __init__(self, eta=0.001, n_iter=20):
        self.eta = eta
        self.n_iter = n_iter

    def fit(self, X, y):
        self.w_ = np.zeros(1 + X.shape[1])
        self.cost_ = []

        for i in range(self.n_iter):
            output = self.net_input(X)
            errors = (y - output)
            self.w_[1:] += self.eta * X.T.dot(errors)
            self.w_[0] += self.eta * errors.sum()
            cost = (errors**2).sum() / 2.0
            self.cost_.append(cost)
        return self

    def net_input(self, X):
        return np.dot(X, self.w_[1:]) + self.w_[0]

    def predict(self, X):
        return self.net_input(X)
```

Fig. 6. Part of a machine learning program for a classification task without a single step function

To see our *LinearRegressionGD* linear *regressor* in action, we will use the RM (Profit volume) variable from the IT product development prediction dataset of innovative start-up projects as an explanatory model training variable that can predict MEDV (Interaction Tariffs). Moreover, we standardise the variables for better convergence of the gradient descent algorithm. The corresponding source code looks like this (Fig. 7).



```
X = df[['RM']].values
y = df['MEDV'].values

from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
sc_y = StandardScaler()
X_std = sc_x.fit_transform(X)
y_std = sc_y.fit_transform(y)
lr = LinearRegressionGD()
lr.fit(X_std, y_std)
```

Fig. 7. Standardisation of variables for gradient descent algorithm

Let us plot the value against the number of epochs to check the convergence of the linear regression. As we can see in the chart below, the algorithm of the gradient downturn converged after the fifth epoch (Fig. 8).

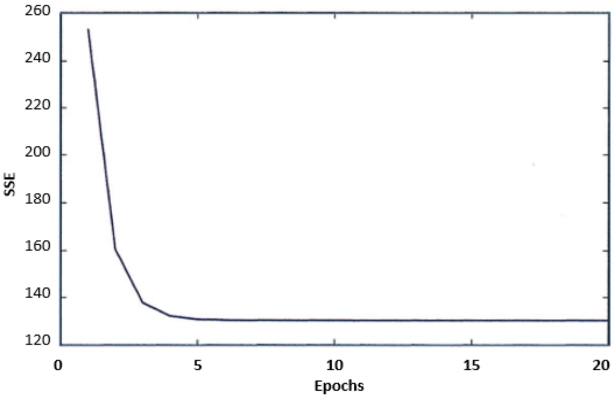
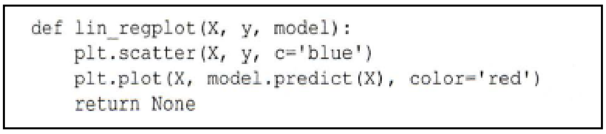


Fig. 8. Graph of cost in relation to the number of epochs

Note that it is more efficient (fewer emissions) to work with non-standardized variables in the linear regression object *LinearRegression* library *scikit-learn*, which uses the dynamic library LIBLINEAR and advanced optimisation algorithms. The RANSAC model is used in this case.

As we can see from the execution of the previous source code, the LinearRegression model of the *scikit-learn* library, fitted with non-standardized variables RM and MEDV, produced other model coefficients.

Let us compare it with our implementation based on gradient descent by constructing the MEDV chart in relation to RM (Fig. 9).



```
def lin_regplot(X, y, model):
    plt.scatter(X, y, c='blue')
    plt.plot(X, model.predict(X), color='red')
    return None
```

Fig. 9. A code fragment for plotting a point chart of linear regression

Having built a graph of training data and performed the model fitting by executing the above source code, we can now see that the overall result looks identical to our implementation based on the gradient descent (Fig. 10).

Using the RANSAC model, we have reduced the potential impact of emissions in this dataset, but we do not know if this approach has a positive impact on predictive capacity on previously unseen data.

After executing the next source code, we should see a residual graph with a line passing through the beginning of the X-axis, as shown below (Fig. 11).

In case of perfect prediction, the residues would be strictly zeros, which in real and practical applications we will probably never encounter.

However, we expect from a good regression model that the errors are distributed randomly, and the remains are randomly scattered around the midline.

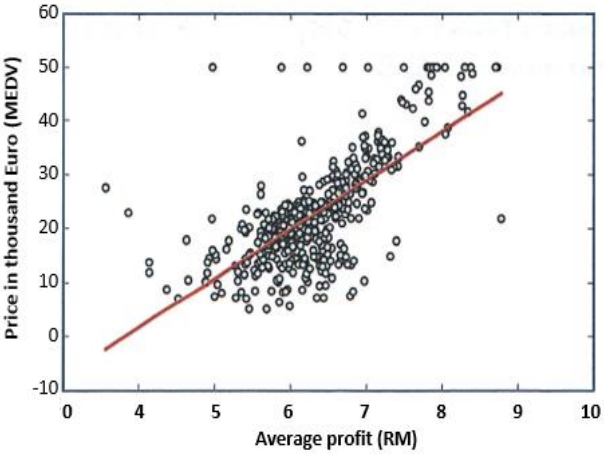


Fig. 10. The trend of IT projects development depending on the level of profit, based on training data

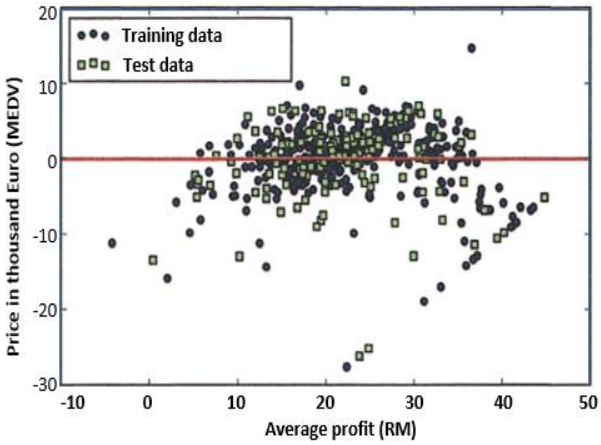


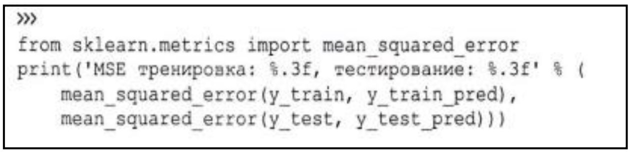
Fig. 11. Errors in fitting training data to the test data

In the case of an ideal prediction, the residuals would be strictly zeros, which we will probably never encounter in real and practical applications. However, from a good regression model, we expect errors to be randomly distributed, and residuals randomly scattered around the midline

Another useful quantitative measure for assessing the quality of a model is the so-called weighted mean square error (mean squared error, MSE), that is, simply the average value of the SSE cost function, which we minimize to fit the linear regression model. MSE is useful for comparing different regression models or for fine-tuning their parameters by searching the parameter grid and cross-checking:

$$MSE = \frac{1}{N}\sum_{i=1}^{n}\left(y^i - \hat{y}^i\right)^{2\cdot} \qquad (5)$$

We will see that the MSE on the training set is 19.96, and the MSE of the test set is much larger with a value of 27.20.



```
»»
from sklearn.metrics import mean_squared_error
print('MSE тренировка: %.3f, тестирование: %.3f' % (
    mean_squared_error(y_train, y_train_pred),
    mean_squared_error(y_test, y_test_pred)))
```

Fig. 12. Errors in fitting training data to the test data

5. CONCLUSION

The analysis of the considered approaches to the definition of effective interaction of customers with a mass information system allows defining use of modern mechanisms for the analysis on the base of Customer Journey Maps. In this case, big data sets are generated, which cannot be analysed by traditional methods. To solve this problem, the authors used machine learning algorithms of neural network types.

An exploratory data analysis method was applied to the available data to find repetitive patterns and anomalies, which proved to be effective. It has allowed constructing a model of realisation of linear regression with the use of the approach based on gradient optimisation.

This approach allowed to construct a base for prediction of processes of the development program of the start-up project product.

REFERENCES

[1] Trends in the development of the global market for information technology. [Online]. Available: http://eir.pstu.edu/handle/123456789/4299

[2] Euripidis Loukis, Marijn Janssen, Ianislav Mintchev, Determinants of software-as-a-service benefits and impact on firm performance, Decision Support Systems, Volume 117, 2019, pp. 38–47.

[3] B. Brown, K. Swani, Introduction to the special issue: B2B advertising, Industrial Marketing Management, February 2020, DOI: 10.1016/j.indmarman.2020.02.006

[4] Customer Journey Map: how to understand what the consumer needs. (2019) Available at: https://www.uplab.ru/blog/customer-journey-map/

[5] S.Gordon, M. Linoff, Data mining techniques: for marketing, sales, and customer relationship management. Published by Wiley Publishind Inc., 10475 Crosspoint Boulevard, Indianapolis, 830 p., 2011.

[6] IBM Institute for Business Value. Analytics: The real-world use of big data in consumer products, 2013.

[7] Seductive Interaction Design: Creating Playful, Fun, and Effective User Experiences (Voices That Matter) 1st Edition (2019) Available at: https://www.amazon.com/Seductive-Interaction-Design-Effective-Experiences/d p/0321725522/

[8] A. Zudov, Modeling of potential rule interactions in active databases. In journal "Modern problems of science and education" (2015). № 1–1.; URL: http://www.science-education.ru/ru/article/view?id=17745.

[9] V. Gogunskii, O. Kolesnikov, G. Oborska, S. Harelik, D. Lukianov, Representation of project systems using the Markov chain, Eastern-European Journal of Enterprise Technologies, № 1/3 (85), 2017б pp. 25–32.

[10] International Project Management Association. Individual Competence Baseline Version 4.0. International Project Management Association, 432 p., 2015.

[11] A Guide to the project management body of knowledge (PMBoK guide). Sixth Edition – USA: PMI Inc., 537 p., 2017.

[12] R. Turner, Guide to project-based management, tran. from English, Moskow, Grebennikov Publishing House, 552 p., 2007.

[13] V. Morozov, O. Kalnichenko, S. Bronin, Development Of The Model Of The Proactive Approach in Creation Of Distributed Information Systems. Eastern-European Journal of Enterprise Technologies, № 43/2 (94), pp. 6–15 (2018).

[14] A. Timinsky, O. Voitenko, I. Achkasov, Competence-based knowledge management in project oriented organisations in bi-adaptive context. Proceedings of the IEEE 14th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT–2019). – Lviv, 2019, pp. 17–20.

[15] O. Maimon, L. Rokach, Data Mining and Knowledge Discovery Handbook, 2005, [Online]. Available: http://www.bookmetrix.com/detail/book/ae1ad394-f821-4df2-9cc4-cbf8b93edf40

[16] V. Morozov, O. Kalnichenko, M. Proskurin, O. Mezentseva, Investigation of Forecasting Methods the State of Complex IT-Projects With Using Deep Learning Neural Networks, Published in the book "Lecture notes in computational intelligence and decision making" (series "Advances in intelligent systems and computing"), vol. 1020, 2020, pp. 261–280,.

[17] E. Loukis, M. Janssen, I. Mintchev, Determinants of software-as-a-service benefits and impact on firm performance, Decision Support Systems, Volume 117, 2019, pp. 38–47.

[18] K. Swani, B. Brown, Su. Mudambi, The untapped potential of B2B advertising: A literature review and future agenda, Industrial Marketing Management, 2019.

[19] D. Herhausen, K. Kleinlercher, P. Verhoef, T. Rudolph, Loyalty Formation for Different Customer Journey Segments, Journal of Retailing, 2019.

[20] V. Morozov, O. Kalnichenko, M. Proskurin, Methods of Proactive Management of Complex Projects Based on Neural Networks, Proceedings of the 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2019, pp. 964–969.

[21] D. Garaedagi, System thinking. How to manage chaos and complex processes. Platform for modeling business architecture, Grevtsov Buks (Grevtsov Publicher). 480 p., 2011.

[22] A. Novikov, Ezhov A.A., Rosenblatt's multilayer neural network and its application for solving the problem of signature recognition, Bulletin of TSU. Technical science. No. 2. pp. 188–197, 2016.

[23] D. Komashinsky, D. Smirnov, Neural networks and their use in control and communication systems, Hotline-Telecom. p. 94, 2003.

[24] Li, L., Fan, K., Zhang, Z., Xia, Z. Community detection algorithm based on local expansion K-means, Neural Network World, 26(6), 2016, pp.589–605.

[25] I. Prigogine, G.Nikolis, Knowledge of the complex. Introduction, Per. from English, M .: Lenar, 360 p., 2017.

[26] G. Carmantini, S. Rodrigues, P. Graben, M. Desroches, A modular architecture for transparent computation in recurrent neural networks, Neural Networks, #85, 2017, pp.85–105.

[27] A. Hosseini, A non-penalty recurrent neural network for solving a class of constrained optimization problems, Neural Networks, 2016, #73, pp.10–25

[28] A. Polaine, L. Løvlie, Service Design: From Insight to Implementation, Rosenfeld Media, 2013, 216 p.

[29] S. Anderson, Seductive Interaction Design: Creating Playful, Fun, and Effective User Experiences, New Riders; 240 p., 2011 https://www.amazon.com/Seductive-Interaction-Design-Effective-Experiences/dp/0321725522/

[30] Installing and getting started, [Online]. Available: https://seaborn.pydata.org/installing. html

Стаття надійшла до редколегії 18.10.2020

Використання навчальних нейронних мереж для прогнозування подій у розробленні продукції IT

Стан розвитку інновацій в Україні характеризується збільшенням розробок на основі стартап-проєктів із використанням як продукту проєкту інформаційних систем різної складності. Проведено аналіз слабкої живучості результатів виконання стартап-проєктів. Зроблено висновок щодо необхідності прогнозування етапів розвитку продуктів IT-проєктів на основі аналізу процесів взаємодії користувачів (клієнтів) з інформаційною системою (продуктом). Розглянуто складові моделі прогнозування розвитку IT-продуктів інноваційних стартап-проєктів, з урахуванням аналізу формуються набори даних взаємодії потенційних клієнтів із такими продуктами. Запропоновано алгоритм формування початкових наборів даних на основі карт подорожей клієнтів (CJM), які є інструментом фіксації подій взаємодії клієнтів із системою. Наведено приклади моделей аналогів карт подорожей клієнтів, які є базою для фіксації та аналізу взаємодій. Цей факт є основою для формування відповідних наборів даних великої розмірності. Як механізм оброблення великих масивів даних і побудови стратегій розвитку IT-продуктів запропоновано використання нейронних мереж глибокого навчання. Побудовано математичні моделі для подальшого моделювання й аналізу отриманих результатів. Використано простий лінійний регресійний аналіз для моделювання зв'язку між єдиною пояснювальної змінної і безперервної змінної відгуку (залежною змінною). Для наявних даних застосовано метод розвідувального аналізу даних для пошуку повторюваних образів і аномалій. У ході дослідження побудовано модель реалізації лінійної регресії з використанням підходу на основі градієнтної оптимізації. Також застосовано лінійні моделі бібліотеки scikit-learn для завдання регресії і реалізовано стабілізаційний регресійний метод. Проведено моделювання й аналіз отриманих результатів, який показав більшу ефективність щодо збільшеного періоду життєвого циклу продуктів IT-проєктів.

Ключові слова: *запуск; інформаційна взаємодія; карта подорожей клієнта; прогнозування.*

Viktor Morozov,
PhD, professor, Head of Dept. of Technology Management of Faculty of Informational Technology of Taras Shevchenko National University of Kyiv.

Віктор Морозов,
кандидат технічних наук, професор, завідувач кафедри менеджменту технологій факультету інформаційних технологій Київського національного університету імені Тараса Шевченка.

Olga Mezentseva,
PhD, Associate professor of Dept. of Technology Management of Faculty of Informational Technology of Taras Shevchenko National University of Kyiv.

Ольга Мезенцева,
кандидат технічних наук, доцент кафедри менеджменту технологій факультету інформаційних технологій Київського національного університету імені Тараса Шевченка.