Development of a Probabilistic Model for Selecting a Partner Offer to a Client using Machine Learning Technologies

[1]NATALIA MAMEDOVA, [1]OLGA STAROVEROVA, [1]GEORGY EPIFANOV, [2]HUAMING ZHANG, [1]ARKADIY URINTSOV
[1]Basic Department of digital economy, Higher School of Cyber Technologies, Mathematics and Statistics, Plekhanov Russian University of Economics, RUSSIA
[2]Deputy Dean, School of Economics, Shanxi University of Finance and Economics, CHINA

Abstract: The article describes the process of developing a model, the implementation of which will change the process of the work of the company's specialists in selecting a partner offer to the client. A practical request for the development of the model was the fact that a huge amount of available information about the client, which affects decision-making to varying degrees, complicates its processing and increases the risks of making a biased decision. The theoretical significance of the research results lies in the presentation of a tool for making the right decision with a high degree of probability. We proceeded from the practice of implementing a business process, according to which the most time-consuming and risky stage of the selection process is the stage of forming the initial data sample. And they suggested using machine learning in order to simplify it significantly. The process of developing the model is presented in stages in this paper so that it can be reproduced and verified. The practical significance of this work is that the results obtained can be applied in the entire range of marketing services and can be used in companies working with large amounts of customer data.

Key-Words: selection model, machine learning, loyalty program, partner offer, client, neural network

Received: March 18, 2022. Revised: November 11, 2022. Accepted: December 9, 2022. Published: December 31, 2022.

1 Introduction

Today, loyalty programs are ubiquitous and are used in all industries. A simple, at first glance, tactic of recognizing the seller and rewarding his best customers has long gone beyond the tools of an ordinary marketer. Now it has become the policy of the organization and even the direction of its strategic development.

It is said that marketing is dynamically developing all over the world to ensure the competitiveness of business in changing conditions, [1]. However, it is more correct to talk about the dynamic development of marketing technologies. At the same time, the foundation on which new techniques and tools are emerging has not changed – it is always the involvement of the consumer, that is, his desire to perceive and respond to the information that is offered to him, [2], [3].

The bottleneck in the implementation of loyalty programs is the technological aspects of the project. Difficulties arise when it is necessary to collect and analyze data about customers of loyalty programs. This is a large-scale and time-consuming process, which is very variable and has no upper limit of reasonableness, since the object of analysis is the human nature of the client. The inability to ensure the absolute and long-term relevance of customer information due to the impermanence of human nature leads to errors in the implementation of loyalty programs. And all this happens against the background of the fact that the process of analyzing customer data is complicated by a permanent increase in the initial data – about him, his behavior, needs, consumption experience and preferences. The main problem in the field of making business decisions based on the results of such an analysis is the difficulty of allocating abstract information about the customers of loyalty programs.

The study was based on the hypothesis that the irrelevance of offers and promotions for the client reduces his brand loyalty and consumer loyalty. This causes the need to consider and propose machine learning technology as a way to solve the problem. And the question immediately arises – what to teach the car? There is no universal methodology for calculating the "consumer loyalty complex", [4].

Aspects of the formation of consumer loyalty as a behavioral line remain open for discussion and solutions. Thus, the obvious solution is to consider the factors that form customer loyalty to the brand and increase consumer loyalty.

We formulated the research question as follows – is it possible to develop a machine learning algorithm to solve the problems of forming a partnership offer to a client. The aim of the study was to develop a probabilistic model for selecting a partner offer to a client using machine learning technology. We believe that the development of a probabilistic model for the selection of a partner offer will reduce the processing time of information and promptly, and most importantly, provide an offer to the client.

2 Materials and Methods

In order to operate with real data, a company was taken as an example, the main activity of which is the implementation of targeted marketing communications with members of loyalty programs. Among the daily tasks of the company's employees is informing customers about joint actions with partners and encouraging customers to perform targeted actions.

Through the loyalty program, the company engages customers in constant interaction and motivates them to buy and use the services of their partners more often. The partners of the loyalty program are large companies, including federal and regional retail chains, online stores. Customers have the opportunity to receive bonuses when buying from partners, privileges when participating in their promotions. The company is engaged in the organization and implementation of loyalty programs in a turnkey format. The top-level business process that displays the process of launching a loyalty program promotion is shown in Figure 1.

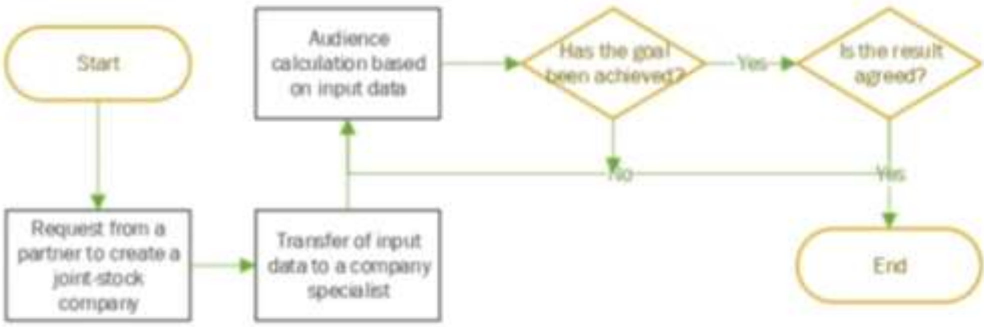

Fig. 1: Business process of launching a loyalty program promotion.

The bottleneck of the business process is the audience calculation stage – it is characterized by the greatest functionality and time load. The SLA established in the company for this stage is 3 working days, but in practice it can stretch up to a month. We see the solution to the problem in developing a model with partner-based customer preferences. Potentially, this will help to free up valuable competencies for solving non-standard tasks. A much more interesting business effect will be an increase in conversion from launching promotions, if we agree that the probabilistic machine learning model (in our case, for the selection of an affiliate offer) works more accurately than a person, [5], [6].

Our problem is a classification problem, [7], that is, it is a problem of assigning a sample to one of several pairwise non-intersecting sets. It includes the predestination of the customer's choice based on the experience of the target action (purchase, participation in the promotion), attitude to the brand, discount rate, the frequency of the target action, new products and others.

These tasks are solved by correlating statistical samples to specific classes (specific characteristics). Samples are represented by a vector, the components of which are characteristics of this sample and influence the decision of belonging to a particular class. The level of complexity of the system is determined the complexity level of the system is determined according to the recommended, [8]. The first is linear separability, when classes can be separated by straight lines. The second level is nonlinear separability, when classes cannot be separated by straight lines, but a curve can be drawn between them. The third level of complexity is probabilistic separability when classes intersect.

Ideally, after preprocessing, we should get a linearly separable problem, since after that the construction of the classifier is greatly simplified. [9]. Unfortunately, when solving a real problem, we have a limited number of samples, on the basis of which the classifier is built. At the same time, we were unable to carry out such data preprocessing in which linear separability of samples could be achieved.

Neural networks with direct communication are a universal means of approximating functions, which allows them to be used in solving classification problems. As a rule, neural networks turn out to be the most effective way of classification, because they actually generate a large number of regression models (which are used in solving classification problems by statistical methods), [10].

A data set was applied that contains information about the partners of the loyalty program. The functions (characteristics) in this data set are as follows:

- CRIM: frequency of purchase from a partner
- ZN: the share held by the partner within the program
- INDUS: share of purchases in a specific category
- CHAS: Boolean variable of purchase from a partner in the context of 1 year (it is equal to 1 if the purchase was made; 0 otherwise)
- NOX: average receipt
- RM: average number of purchases from a partner by his "client"
- AGE: target age category
- DIS: average number of regular customers
- TAX: partner's bonus privileges (calculated as the average of the transaction)
- LSTAT: percentage of involved program participants
- MEDV: average customer interest in the partner (calculated as the number of transactions the partner has relative to competitors in the category)

Below is a fragment of an overview of the original data set for the first five values (Fig. 2):

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33	36.2
5	0.02985	0.0	2.18	0.0	0.458	6.430	58.7	6.0622	3.0	222.0	18.7	394.12	5.21	28.7

Fig. 2: Overview of the source data.

We started work by creating a scattering matrix that allows us to visualize paired relationships and correlations between the various functions (Fig. 3). It was also determined how the data are distributed and whether they coincide with outliers or not.

```
import matplotlib.pyplot as plt
import seaborn as sns%matplotlib inline
# Calculate and show pairplot
sns.pairplot(data,size=2.5)
plt.tight_layout()
```

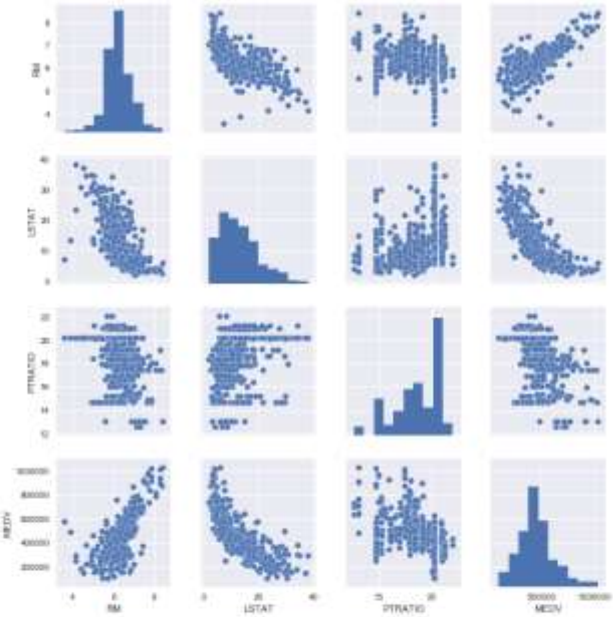

Fig. 3: Scattering matrices of characteristics.

We saw a clear linear relationship between the characteristic "RM" and "MEDV". In addition, the histogram showed that the variable "MEDV" is normally distributed, but contains several outliers.

Next, a correlation matrix was created to quantify and generalize the relationships between variables.

```
# Calculate and show correlation matrix
cm = np.corrcoef(data.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)
```

The correlation matrix is closely related to the covariance matrix, in fact, it is a modified version of the covariance matrix calculated using standardized features, [11]. This square matrix (with the same number of columns and rows), which contains the correlation coefficient of characteristics, is shown in Figure 4.

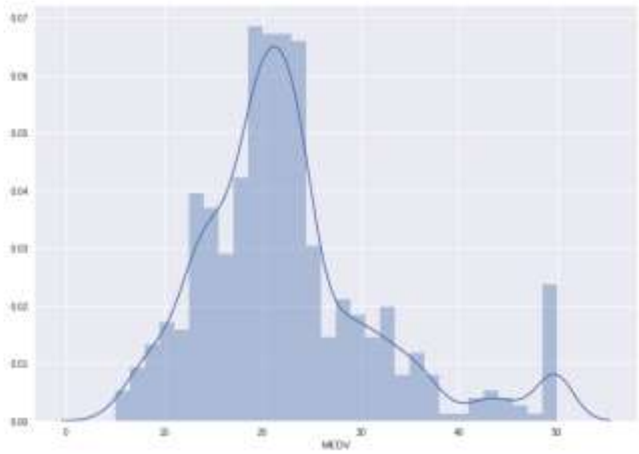

Fig. 4: Correlation matrix of characteristics.

Our target variable becomes the characteristic "MEDV", it corresponds to the regression model. Next, a distribution was built for it using the listplot function from the Seaborn library (Fig. 5).

```
sns.set(rc={'figure.figsize':(11.7,8.27)})

sns.distplot(boston['MEDV'], bins=30)

plt.show()
```

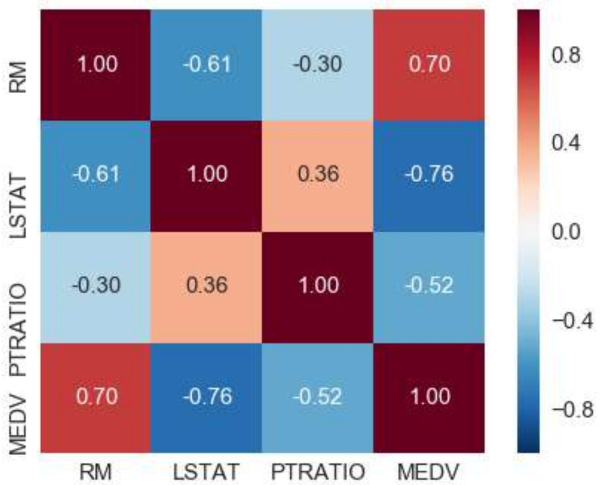

Fig. 5: Correlation matrix of all characteristics.

Obviously, MEDV values are distributed normally with a small amount of outliers.

A correlation matrix has been created that measures the linear relationships between all characteristics using the core function from the Pandas data library. We used the heatmap function from the Seaborn library to construct the correlation matrix (Fig. 6).

```
correlation_matrix = boston.corr().round(2)
# annot = True to print the values inside the square
sns.heatmap(data=correlation_matrix, annot=True)
```

Fig. 6: Correlation matrix of all characteristics.

In order to fit the linear regression model, we chose exactly those characteristics that have a high correlation with our target variable MEDV. It was determined that the variable RM has a strong positive correlation with MODS (0.7) and at the same time LSTAT has a high negative correlation with MEDV (-0.74). Based on the analysis, it was decided to use RM and LSTAT as input characteristics. Using a dot graph, you can see how these functions change with MEDV (Fig. 7).

```
plt.figure(figsize=(20, 5))

features = ['LSTAT', 'RM']

target = boston['MEDV']

for i, col in enumerate(features):

    plt.subplot(1, len(features) , i+1)

    x = boston[col]
    y = target
    plt.scatter(x, y, marker='o')

    plt.title(col)
    plt.xlabel(col)
    plt.ylabel('MEDV')
```

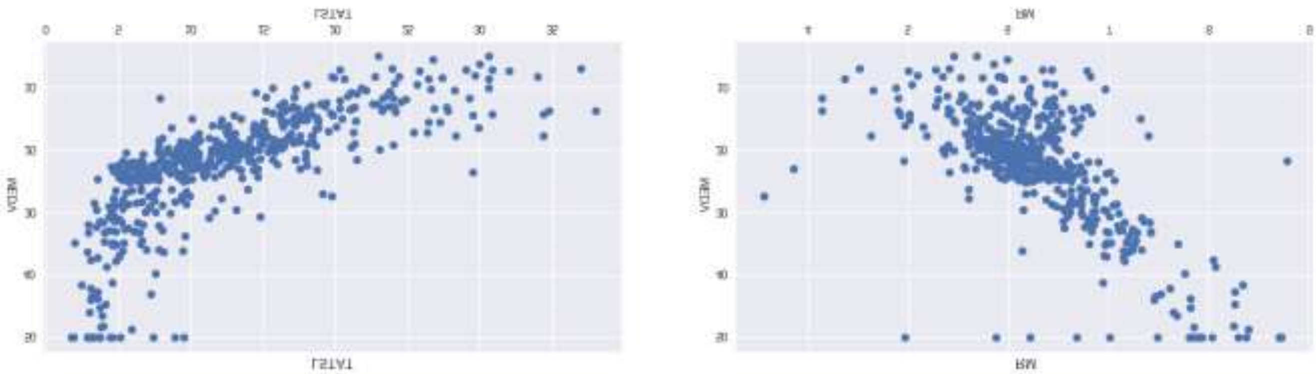

Fig. 7. RM and LSTAT graphs from MEDV.

The conclusions were made as follows. The number of repeated transactions increases as the RM increases linearly. There are several outliers, and the data is limited to the 50 bars. There is a downward trend in the growth of L STAT, although this is a very direct linear relationship. For clarity, a graph of the correlation of the variables INDUS and PTRATIO to the output function MEDV was constructed (Fig. 8).

```
plt.figure(figsize=(20, 10))

features = ['PTRATIO', 'INDUS']
target = boston['MEDV']

for i, col in enumerate(features):
    plt.subplot(1, len(features) , i+1)
    x = boston[col]
    y = target
    plt.scatter(x, y, marker='o')
    plt.title(col)
    plt.xlabel(col)
    plt.ylabel('MEDV')
```

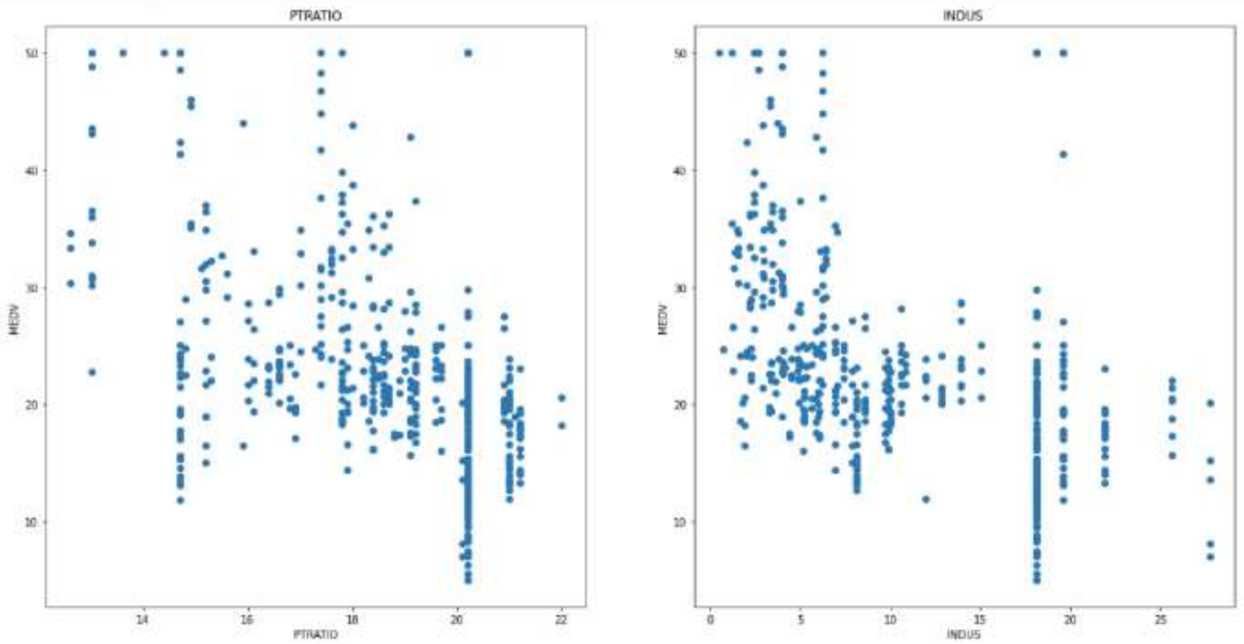

Fig. 8: Correlation of INDUS and PTRATIO values from MEDV.

Obviously, there is no correlation strong enough to include this data in the set of input characteristics.

Thus, we extracted only significant characteristics from the data set, that is, we optimized the process of creating and training our neural network. Since our goal is to develop a neural network capable of predicting the probability of a client's interest in a partner with acceptable accuracy, we have divided the data set into functions and a target variable.

The functions "RM", "LSTAT" and "PTRATIO" gave us quantitative information about each point from the dataset. We saved them in the features variable. The target variable MEDV has become the variable we are trying to predict. We have saved to store it in prices.

 # Import libraries necessary for this project
import numpy as np
import pandas as pd
from sklearn.model_selection import ShuffleSplit
 # Import supplementary visualizations code visuals.py
import visuals as vs
 # Pretty display for notebooks
%matplotlib inline

Further, the data were developed into training and test sets. We trained the model with 80% of all the examples and tested with the remaining 20% of the examples. To separate the data, the train_test_split function provided by the scikit-learn library was used. To check the success of this operation, we will print the dimensions of our training and test set (Fig. 9).

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

lin_model = LinearRegression()
lin_model.fit(X_train, Y_train)

LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Fig. 9: Dividing the data set into test and training.

We used the Scikit-Learn Linear Regression function to train our model on training and test sets (Fig. 10).

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

lin_model = LinearRegression()
lin_model.fit(X_train, Y_train)

LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Fig. 10: Neural network training.

Thus, we analyzed in detail the process of developing a model for forming a partnership offer to a client, illustrated various attribute dependencies and were able to ensure that the model fulfilled its tasks.

The final implementation requires that we put everything together and train the model using a decision tree algorithm.

To verify the fact that we created exactly the optimal model, we trained the model using various max_depth parameters for the decision tree (Fig. 11). The max_depth parameter can be thought of as the number of questions that the decision tree algorithm can ask about the data before making a prediction.

In addition, we found that our program uses ShuffleSplit() for an alternative form of cross-validation (see 'cv_sets' variable). The implementation of ShuffleSplit() below created 10 ('n_splits') different sets, and for each shuffle, 20% ('test_size') of the data was used as a test set.

```
# Import 'make_scorer', 'DecisionTreeRegressor', and 'GridSearchCV'
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import make_scorer
from sklearn.model_selection import GridSearchCV

def fit_model(X, y):
    """ Performs grid search over the 'max_depth' parameter for a
        decision tree regressor trained on the input data [X, y]. """

    # Create cross-validation sets from the training data
    cv_sets = ShuffleSplit(n_splits = 10, test_size = 0.20, random_state = 0)

    # Create a decision tree regressor object
    regressor = DecisionTreeRegressor()

    # Create a dictionary for the parameter 'max_depth' with a range from 1 to 10
    params = {'max_depth':[1,2,3,4,5,6,7,8,9,10]}

    # Transform 'performance_metric' into a scoring function using 'make_scorer'
    scoring_fnc = make_scorer(performance_metric)

    # Create the grid search cv object --> GridSearchCV()
    grid = GridSearchCV(estimator=regressor, param_grid=params, scoring=scoring_fnc, cv=cv_sets)

    # Fit the grid search object to the data to compute the optimal model
    grid = grid.fit(X, y)

    # Return the optimal model after fitting the data
    return grid.best_estimator_
```

Fig. 11: Neural network training with 10 different data sets.

Once the model is sufficiently trained, it can be used to predict new output values. We were able to use these predictions to obtain information about data for which the value of the target variable is unknown, that is, such data did not appear in our dataset. The following code fragment finds exactly the maximum depth that returns the optimal model (Fig. 12).

```
# Fit the training data to the model using grid search
reg = fit_model(X_train, y_train)

# Produce the value for 'max_depth'
print("Parameter 'max_depth' is {} for the optimal model.".format(reg.get_params()['max_depth']))

Parameter 'max_depth' is 4 for the optimal model.
```

Fig. 12: Finding the optimal depth of learning.

An optimal model is not necessarily a reliable model. We can talk about the reliability of our model due to the fact that it can be generalized to new data, as well as the fact that the model can use a learning algorithm that corresponds to the data structure. However, we did not have the opportunity to lower the noise level and increase the volume of sample data in order to strengthen the fitness of the model and its capture of the target variable.

As a result, we have received an optimally trained model for forming a partner offer to the client. At the same time, we managed to avoid retraining and get an acceptable discrepancy between the desired and actual result.

3 Results and Discussion

The result of the developmental model of forming a partnership offer to a client is a table that describes the relationship of a certain client to a specific partner with a certain probability. The client and partner are represented as numeric identifiers, which are decrypted using existing directories (Table 1).

This is the result of the model and consists of 3 attributes:

- client;
- partner;
- probability of interest.

The main link for determining the partner from the model and the directory is a pair of attributes trade_group_id and partner_id. When connecting data using these keys, we receive information from the partner's directory about its name, business category and information about retail outlets.

In the new process, the company's specialist no longer needs to develop an algorithm for selecting clients for the partner category every time. After implementing the model, it is enough to find the partner ID by name using the directory and filter the set of clients by the probability of customer interest, which is represented in the range from 0 to 1, and the partner ID. In case of a shortage of audience volume, the probability parameters change downward.

Table 1. Directory of the partner entity.

Table Name	Attribute	Attribute Description	Type	Value
partner	partner_id	partner ID	number	up to 5 digits
partner	partner_name	name of the partner	line	
partner	merchant_id	ID of the point of sale	number	from 111111111111 to 9999999999999999999
partner	merchant_nm	name of the point of sale	line	
partner	mcc_id	point of sale category	number	4 digits

Figure 13 shows the developed model for determining the probability of a client's interest in a particular loyalty program partner.

	customer_id	trade_group_id	proba
1	446 893 394	21 251	0,0081617789
2	312 435 889	21 251	0,13344645
3	312 931 070	21 251	0,0038086353
4	291 942 461	21 251	0,0104580847
5	208 202 454	21 251	0,0074165426
6	204 589 667	21 251	0,0061344454
7	206 673 121	21 251	0,003098088
8	206 447 021	21 251	0,009467814
9	252 916 707	21 251	0,0009938461
10	231 763 216	21 251	0,0031169334
11	209 008 432	21 251	0,0056372457
12	210 036 207	21 251	0,0027370905
13	250 459 689	21 251	0,002484889
14	230 005 029	21 251	0,0071378739
15	227 605 595	21 251	0,0652577289

Fig. 13: The result of the model.

With the help of such a tool, there is no need to work with a large amount of data and a diverse set of tables, the load on the database servers is reduced by reducing the number of requests for it.

As a result, we have received a tool that helps to significantly reduce the time of selecting an affiliate offer to the client. We managed to free valuable competencies from routine, repetitive work and direct them to non-standard tasks.

4 Conclusions

This study contains a description of the process of developing a model for forming a partnership offer to a client. Detailing the description allows you to repeat the development process and thereby verify the results of the study. The proof of the model's operability is illustrated by various attribute dependencies.

Thanks to the presented model, interested parties will be able to search in their databases for similar customers who are inclined to buy from a particular partner with a certain degree of probability.

We preferred to avoid the presentation of technical requirements (requirements for personal computers, for DBMS), since they are not significant. In addition, we avoided mentioning specific trade and product names. Also, we did not focus on the issue of choosing the development environment and limited ourselves to identifying essential parameters, following which will ensure similar results.

The end result of our research was an optimally trained model of forming a partnership offer to the client. At the same time, we managed to avoid retraining and get an acceptable discrepancy between the desired and actual result. The approaches used for data processing and the choice of machine learning tools are the theoretical contribution of the conducted research in the subject area of artificial intelligence.

The applied value of the research results is to save time on solving the problem of segmentation, determining the best offer to the client. As a result, there is an increased response to marketing offers and promotions. An indirect positive effect is an increase in customer loyalty to the brand, since the selection of offers received by the customer focuses on his interests.

References:

[1] Moi, L., Cabiddu, F.: An agile marketing capability maturity framework. Tour. Manag. 86, 104347 (2021). https://doi.org/10.1016/J.TOURMAN.2021.104347.

[2] Nadeem, W., Tan, T.M., Tajvidi, M., Hajli, N.: How do experiences enhance brand relationship performance and value co-creation in social commerce? The role of consumer engagement and self brand-connection. Technol. Forecast. Soc. Change. 171, 120952 (2021). https://doi.org/10.1016/J.TECHFORE.2021.120952.

[3] Hepola, J., Leppäniemi, M., Karjaluoto, H.: Is it all about consumer engagement? Explaining continuance intention for utilitarian and hedonic service consumption. J. Retail. Consum. Serv. 57, 102232 (2020). https://doi.org/10.1016/J.JRETCONSER.2020.102232.

[4] Tasci, A.D.A.: A critical review of consumer value and its complex relationships in the consumer-based brand equity network. J. Destin. Mark. Manag. 5, 171–191 (2016). https://doi.org/10.1016/J.JDMM.2015.12.010.

[5] Guo, M., Zhang, Q., Liao, X., Chen, F.Y., Zeng, D.D.: A hybrid machine learning framework for analyzing human decision-making through learning preferences. Omega. 101, 102263 (2021). https://doi.org/10.1016/J.OMEGA.2020.102263.

[6] Santos, S., Kissamitaki, M., Chiesa, M.: Should humans work? Telecomm. Policy. 44, 101910 (2020). https://doi.org/10.1016/J.TELPOL.2020.101910.

[7] Goswami, T.: Machine learning behind classification tasks in various engineering and science domains. Cogn. Informatics, Comput. Model. Cogn. Sci. 339–356 (2020). https://doi.org/10.1016/B978-0-12-819443-0.00016-7.

[8] Kim, K., Lin, H., Choi, J.Y., Choi, K.: A design framework for hierarchical ensemble of multiple feature extractors and multiple classifiers. Pattern Recognit. 52, 1–16 (2016). https://doi.org/10.1016/J.PATCOG.2015.11.006.

[9] Sadeghi Eshkevari, S., Cronin, L., Sadeghi Eshkevari, S., Pakzad, S.N.: Input estimation of nonlinear systems using probabilistic neural network. Mech. Syst. Signal Process. 166, 108368 (2022). https://doi.org/10.1016/J.YMSSP.2021.108368.

[10] Ghiassi, M., Burnley, C.: Measuring effectiveness of a dynamic artificial neural network algorithm for classification problems. Expert Syst. Appl. 37, 3118–3128 (2010). https://doi.org/10.1016/J.ESWA.2009.09.017

[11] Choi, J., Yang, X.: Asymptotic properties of correlation-based principal component analysis. J. Econom. (2021). https://doi.org/10.1016/J.JECONOM.2021.08.003.