

# ORGANIZATION OF ONLINE LEARNING USING THE INTELLIGENT METASYSTEM OF OPEN SEMANTIC TECHNOLOGY FOR INTELLIGENT SYSTEMS

**Aliya Kintonova**

*Corresponding author*

Candidate of Technical Sciences, Associate Professor\*

E-mail: Aliya\_kint@mail.ru

**Amanbek Sabitov**

Doctoral Student, PhD Student\*

**Igor Povkhan**

Doctor of Technical Sciences, Professor, Dean

Department of Software Systems

Uzhhorod National University

University str., 14, Uzhhorod, Ukraine, 88000

**Dinara Khaimulina**

Master's Degree\*

**Galymzhan Gabdreshov**

Candidate of Pedagogical Sciences, Director

Research Institute "Sezual"

Tauelsizdik, str., 41, Astana, Republic of Kazakhstan, 010000

\*Department of Artificial Intelligence Technologies

L. N. Gumilyov Eurasian National University

Satpayev str., 2, Nur-Sultan, Republic of Kazakhstan, 010008

*Distance learning today allows you to create a system of mass continuous self-learning, universal exchange of information, regardless of the presence of time and space zones. For more effective online learning, it is necessary to introduce artificial intelligence technologies and methods for implementing learning systems.*

*In this paper, the objects of research were:*

*1) the capabilities of the OSTIS open semantic technology, the capabilities of the IMS.OSTIS metasystem for organizing online learning;*

*2) machine learning method – classification.*

*Results of the study:*

*1) an online course was organized using the IMS.OSTIS metasystem of the OSTIS open semantic technology;*

*2) for the analysis and visualization of training data, a machine learning method is implemented – classification.*

*The results of the implementation of the online course were obtained using the semantic technology for designing an intelligent learning system: the IMS.OSTIS metasystem using the graphical semantic code SCg. The OSTIS kernel requires a machine with the Ubuntu operating system installed, which is a GNU/Linux distribution based on Debian GNU/Linux, an operating system based on the Linux kernel.*

*The paper also shows an example of using the machine learning method – classification.*

*This method allows you to classify data. Intelligent processing and visualization of data were carried out based on the results of testing students in order to classify them into letter categories A, B, C, D according to a set of features: scores and points of the average score. The high-level Python library Pandas was used, this is a library for data analysis. To visualize the results of data processing, the Matplotlib library in Python was used*

*Keywords: online learning, online course, data visualization method, machine learning method*

Received date 01.11.2022

Accepted date 11.01.2023

Published date 28.02.2023

**How to Cite:** Kintonova, A., Sabitov, A., Povkhan, I., Khaimulina, D., Gabdreshov, G. (2023). Organization of online learning using the intelligent metasystem of open semantic technology for intelligent systems. *Eastern-European Journal of Enterprise Technologies*, 1 (2 (121)), 29–40.

doi: <https://doi.org/10.15587/1729-4061.2023.272952>

## 1. Introduction

The rapidly developing field of distance learning has contributed to the rapid growth in the popularity of online courses. Due to the numerous advantages of online learning, the number of e-learning users is growing, educational institutions offer distance learning.

The popularity of e-learning has skyrocketed. Online learning is affordable.

Online courses are part of online learning where you can gain knowledge and skills using any gadget.

Online learning is changing the traditional understanding of education, it offers learning anytime, anywhere. Popular online courses attract hundreds of thousands of students.

The process of large-scale distribution of distance learning has revealed a number of problems of training systems:

insufficient expansion of opportunities for access to scientific, scientific-methodical, information-analytical materials in various areas of the economy, science and education; insufficient development of technologies that can contribute to the automation of the formation of high-quality content in educational systems; insufficient use of technologies that can help automate the formation of training reports that reflect the success of training in training systems.

In view of the above problems, research on the organization of online learning using intelligent technologies is of scientific relevance. Intelligent technologies allow automating routine intellectual work, structuring the subject area of knowledge, including a formal description of navigation and visualization of educational material, analyzing and visualizing student learning outcomes.

---

## 2. Literature review and problem statement

---

In [1], the authors write that learning analytics methods are widely used in education to gain insight into hidden patterns from educational data. Techniques such as predictive learning analytics are used to identify and measure patterns in studying data and extrapolating future behavior. The paper reports on the use of a popular decision tree classification algorithm. Behavioral features from a public dataset were used to develop a predictive model for predicting learning achievements. But the work does not describe the method of classifying educational data and gives no analysis results on the success of mastering the knowledge of an online course.

In [2], the authors write that the Moodle learning system has some great features that would be useful during the COVID-19 pandemic. The authors describe an extended version of the technology adoption model to explore the main factors that influence students' decisions to use the e-learning system. The model was populated using data from a survey of 389 undergraduate students who used the Moodle-based e-learning system at Alazhar University. The model was evaluated using structural equation modeling (SEM). The path model was developed to analyze the relationships among factors that explain student adoption of the e-learning system. The results showed that computer anxiety, course content, hedonic motivation, perceived environment, subjective norm, and technical support significantly affected both ease of use and usefulness. The subjective norm significantly influences the intention to use. Perceived ease of use and perceived usefulness significantly affect the intention to use online learning. The paper describes a model that reflects the interrelated factors of student acceptance of the e-learning system, but does not describe technologies and methods for analyzing and processing educational data that reflect the success of learning.

The paper [3] focuses on blended learning, which combines digital online resources with traditional classroom activities and enables students to achieve higher learning outcomes through well-defined interactive strategies including online and traditional learning activities.

Learning analytics is a conceptual framework and is part of ours. Precision learning is used to analyze and predict student achievements and provide timely interventions based on student learning profiles. The study applied learning analytics and educational big data.

The results show that the final academic performance of students can be predicted if only one-third of the semester has passed. In addition, seven critical factors affecting student performance were identified, consisting of four online factors and three traditional factors. The results showed that a mixed dataset combining online and traditional critical factors had the highest predictive performance.

Using well-defined online and traditional variables, students' final academic performance was predicted. The performance was determined by a regression model to improve the performance of forecasting. The student performance prediction model is proposed in the study, and a well-defined intervention strategy should be integrated into the learning analytics framework. The full framework of learning analytics can be applied to predict student learning outcomes in the second semester of such a calculus course. The factors influencing the final performance indicators are shown. But the technology for implementing the analysis and visualization of training data is not given.

In [4], the authors write that well-designed online courses expand learning opportunities and allow students to effectively develop skills and knowledge. The authors of the work described tools for evaluating course design, with wider adoption of educational technologies. They showed how learning objectives correlate with grades. The authors presented methods for studying the effectiveness of courses, a methodology for developing and assessing the consistency between course learning objectives and evaluation activities using a combination of learning analytics and measurement theory approaches, and more specifically, research multivariate response theory models. A methodology within a professional MOOC to develop leadership skills was offered. The results of aligning existing course objectives with evaluation activities are given.

The paper [5] presents the results of the study of the media model. It becomes a guide for teachers in the implementation of technological learning using the appropriate method. The authors write that basic pedagogical skills are one of the important aspects of teachers' pedagogical abilities. They indicate the need to use multimedia technologies as a means of teaching basic pedagogical skills, for example, in the form of audiovisual or mobile applications. The authors state that their research will allow the development of teaching innovations in the form of Massive Open Online Courses (MOOCs), video content on core teaching skills to improve the pedagogical competence of Indonesian teachers, which contains multimedia content, especially videos that will help lecturers and teachers (tutors) to understand how to teach well. But the paper does not describe technologies for implementing online courses, since the work is devoted to the development of multimedia content.

In [6], the author studied the relationship between indicators of knowledge of technological pedagogical content and teachers' confidence in self-efficacy in relation to technology integration (TPACK). Within the one-group design before and after testing, correlation analysis identified several areas of expertise in the proposed TPACK model as having a significant and positive correlation with self-efficacy beliefs regarding technology integration. Multiple regression analyses of pre-test and post-test data showed a change over time in the predictive relationship between knowledge scores in TPACK domains and self-efficacy beliefs. The results of the study illustrate the changing nature of the complex relationship between knowledge and self-efficacy beliefs and highlight potential knowledge areas in TPACK areas that influence elementary school teachers' beliefs about technology integration.

In [7], the authors write that educational data mining includes the application of data mining methods to the activities of students. However, in the context of computer programming, many data mining techniques cannot be applied because they require vector input, whereas computer programs are in the form of syntax trees. The authors describe ast2vec, a neural network that converts Python syntax trees to vectors and vice versa. This allows you to use about a hundred data mining methods that were previously inapplicable. Ast2vec has been trained on almost half a million programs by novice programmers and is designed to be applied to learning tasks without retraining, meaning that users can apply it without the need for deep learning. Ast2vec can be used to make educational decisions. The authors examine the ability of ast2vec to reconstruct the original syntax tree from its vector representation on training data

and two other large-scale programming datasets, evaluate the predictive power of a linear dynamic system on top of `ast2vec`, obtaining results similar to methods that work directly with syntax trees, but much faster (processing with constant time instead of linear). `Ast2vec` will expand the educational data mining toolbox by making computer program analysis simpler, richer, and more efficient.

Analysis of literature data shows that methods of educational analytics, methods of predictive analytics of learning, technologies and methods for analyzing and processing educational data (test results, results of practical tasks, exam results) are not described in many works.

---

### 3. The aim and objectives of the study

---

The aim of the study is to implement an online course using semantic intelligent technologies and machine learning methods.

To achieve the aim, the following objectives are accomplished:

- to justify the choice of a learning system;
- to apply the method of machine learning to analyze learning outcomes (analysis of test results);
- to visualize the results of the training data analysis.

---

### 4. Materials and methods

---

The object of the study was intelligent technologies for organizing online learning.

The main hypothesis of the study is the possibility of using intelligent technologies to organize online learning.

To implement intelligent online learning, the machine learning method (classification) was used.

Simplifications adopted in the work: to implement the machine learning method, the Pandas library was used.

Studies carried out: publications on this issue have been studied by the method of analysis.

The analysis method allows you to analyze existing publications on the organization of online learning. The analysis method also made it possible to determine a method for analyzing and visualizing training data. As a result of the analysis of the methods, we settled on the machine learning method: classification (grouping) – the distribution of grades into groups according to letter characteristics.

Software and hardware and conditions for the implementation of tasks:

1. We import Pandas, Matplotlib machine learning libraries for the analysis and visualization of training data. Method of intelligent data processing and visualization. After training, students are tested. Test results can be analyzed. The results of their processing, in order to classify them into letter categories A, B, C, D according to a set of features: scores and GPA scores. Classification is the assignment of objects to a certain class according to a set of features. To do this, you can use the method of intelligent data processing – classification. The data set consists of student grades of letter types: A, B, C, D and in points: from 50 to 100 points. The target variable will be graded into groups A (excellent), B (good), C and D (satisfactory). Accordingly, our task is to use the available data to try to find the relationship between the letter grades and the GPA values: A, B, C, D.

2. Operating system. The OSTIS kernel requires a machine with an installed Ubuntu operating system version of at least 14.04 LTS. Ubuntu is a GNU/Linux distribution based on Debian GNU/Linux. To install the OSTIS core, you need an Internet connection, an installed MySQL server, a distributed version control system for GIT files. To use the Internet, you will need a registered domain name, a server with a static global IP address or VPS hosting. It will be necessary to configure a DNS record for a domain of the form: `mydomain.kz *IP address of the server*`. To start using the system, the user needs to launch the OSTIS core system, open a browser.

---

### 5. Results of the study of intelligent technologies for online learning

---

#### 5.1. Justification for the choice of a learning system

Many of the papers we have analyzed are more concerned with methodological problems in organizing online learning. Our work describes an example of the practical implementation of the online course “Algorithms and Data Structures” using the open semantic technology OSTIS – the IMS.OSTIS metasystem, using the SC (Semantic Code) graphical language:

- 1) stage of implementation. Creation of nodes. The core of the OSTIS technology is the IMS.OSTIS metasystem, using the SC language, the SCg code is one of several methods for the visual representation of SC texts and SCg graphs. Graph nodes reflect the educational component (lectures, laboratory, glossary, etc.);

- 2) stage of implementation. Creation of links between nodes. Further, the nodes are combined into a single semantic network. Node links create a single semantic network of online course learning content;

- 3) stage of implementation. Creation of online course content in HTML format. To create content, you can use a text editor and HTML;

- 4) stage of implementation. Creation of cross-references for secondary nodes;

- 5) stage of implementation. Creation of test elements with jQuery.

You can extend the functionality of elements with jQuery and JavaScript.

Such a learning system allows you to quickly find the necessary component of educational content (request-response); when performing practical tasks, it forms blocks of answers for the user, displays messages when the score threshold is passed and displays a message when the test result is unsatisfactory.

An online course was implemented based on the IMS.OSTIS intelligent metasystem using the SC graphical language. IMS.OSTIS is a semantic technology for designing an intelligent learning system. Component (modular) and platform-independent design of an intelligent system with a knowledge base and a knowledge processing model. Stages of development: creation of nodes consisting of elements of educational content (lectures, laboratory work, etc.); creation of links between nodes; creation of online course content in HTML format; creation of cross-references for secondary nodes; creation of test elements using HTML, jQuery, CSS; assembly of the “Glossary” section; assembly of the “Final test” section.

Implementation of the online course “Algorithms and Data Structures” using the SC language of the IMS.OSTIS metasytem.

Technology for designing an intelligent learning system. Technology of interpretation of logical-semantic models of intelligent systems. OSTIS technology (Open Semantic Technologies for Intelligent Systems – the IMS metasytem is an open semantic technology for designing intelligent systems [8–10].

The OSTIS technology is based on: a formal ontology of intelligent systems, or rather, a formal ontology of their logical-semantic models, a formal ontology of designing intelligent systems – a formal ontology of means and methods for designing logical-semantic models of intelligent systems, as well as project activities aimed at developing these logical-semantic models [11].

OSTIS technology is a technology of component (modular) and platform-independent design of semantically compatible intelligent systems that have knowledge bases of any complexity and implement parallel knowledge processing models [12].

The use of OSTIS technology – IMS metasytem on the example of the implementation of “Algorithms and data structures” using the SC language of the IMS metasytem. In the chosen technology, the specified complex of models, means and methods is implemented in the form of an intelligent metasytem, which is formed according to the same proposed technology for designing intelligent systems [13].

*The knowledge processing machine of the IMS.OSTIS intelligent metasytem.*

The knowledge processing machine seems to be the most important part of every intelligent system, because it is the functionality of the knowledge processing machine that determines the capabilities of the system as a whole, the ability to provide answers to complex user questions and the ability to solve various problems.

When forming the IMS knowledge processing machine, OSTIS developers from BSUIR applied a technology that has the following properties:

- generality, that is, predictable technology should provide the processing of knowledge in various subject areas;
- scalability and modularity. By this we mean that the methods should be able to increase the throughput of the system without processing the transformation of the underlying knowledge model machine;
- cross-platform. The developed technology cannot depend on the operating system or architecture of the hardware device;
- parallelism. It is designed to provide the functionality of an asynchronous application of various ways of solving problems within the framework of solving a problem, solving several problems at the same time. In this case, it is necessary to determine the consistency and integrability of the results of applying various methods for solving problems [14, 15];
- validity period. A plan is made to solve the problem in a form accessible to the student.

The technology on the basis of which the knowledge processing machine of the IMS.OSTIS metasytem is formed is part of the open semantic technology for intelligent systems OSTIS [8].

*Appearance of the user interface of the IMS.OSTIS system.*

SC technology for user interface design or what is called semantic technology for designing user interfaces, intelligent systems requires the following provisions to be observed:

- the user interface in SC technology user interface design is considered to be a specialized intelligent system that has the ability to communicate with the user through messaging. This means that the user interface is considered to be the essence of multi-agent systems based on knowledge, therefore, ontologies. However, any other system implemented on OSTIS technology has the same properties [9];

- the main goal is to translate the user interface of requests from a user defined in any language into SC code, which is the internal language of our system. The user interface must translate the system response into any external language available to the user and display the response;

- the graphical interface is implemented on the basis of Semantic Code graphical. This language is used for the visual or graphical representation of SC code texts [16].

Objects that are displayed on the screen using the SCg code are called SCg elements. All specified SCg elements represent a graphical representation of the semantic network nodes. Any object is mapped to a node in the semantic web [17].

The user interface is the IMS.OSTIS metasytem is based on the SCg code. All objects are displayed in the main window signs of some SC elements in the knowledge base.

User interaction with the system is carried out through the exchange of messages. View user forms with user commands. Custom commands are triggered using a predefined standard custom action chain. A standard custom action is an action that occurs when using some input device and cannot be decomposed (broken down into other smaller steps). Examples of such actions are pressing (disengaging) a mouse or keyboard button, moving the mouse, and so on. The following sections are distinguished by the type of user commands, called actions [18, 19]:

- release command – a class of commands that access the system;
- editing commands – a class of commands that execute the corresponding request for external RLS editing of texts;
- view command – a class of commands that perform actions set to display external SC texts.

According to the method of determining arguments, special classes of user commands are distinguished [20]:

- a class of commands with set arguments, that is, the objects for which the request is sent to the action are known in advance. For example, select the menu item View – a decomposition of the command to display an object in which the data subject is itself;
- a class of commands with additional arguments. In these commands, the objects for which the action request is being made must be specified.

There is a basic way to implement commands with optionally specified arguments. In order to specify command arguments, you need to drag them to the arguments panel with the mouse. Then you can initiate a command, just click the left mouse button on the sign of the corresponding command class (top menu).

In order to run a command with predefined arguments, just click the left mouse button on the signs corresponding to the command [21].

The following user commands are currently available in the system:

- request for a complete semantic neighborhood. This command can be initiated in two different ways. The first way: specify as an argument the SC element to which the question applies; left-click on its sign in the main menu (View KB>Semantic Neighborhood>Full Semantic

Neighborhood). The second way: initiate a command with predefined arguments – these are links within the page. Left-click on the command sign (italic text in blue, which is underlined when hovering the mouse cursor);

- command to switch the active window. To do this, left-click on the window symbol (in the left part of the main window). After activating the window, its sign turns blue, and its contents are displayed in the central area of the main window;

- identification mode change command. To run this command, left-click on the button with a triangle, which is located to the right of the current identification mode. Then, in the list that appears, left-click on one of the available identification modes;

- command to prepare the contents of the active window for printing. In order to initiate it, you need to left-click on the command with the image of the printer, which is located above the list of windows, in the left part of the main window. After initiating the command, a tab is created in the browser, the content of which is optimized for printing.

*Logical-semantic model of the user interface.*

Like any other system built using OSTIS technology, the user interface is a metasystem and is built using the component method. Let's single out the following classes of user interface components:

- translation components. The components of this class provide a translation of the SC code in an external language, and vice versa;

- visualization components. The components of this class provide output in a foreign language;

- component editing. This component class provides input of information about the user in an external language.

Each component consists of a knowledge base and a set of SC agent fragments. Of the metasystem broadcasting components in PI, it has the following components:

- a translator from the SC code to the SCsJson format (a format similar in structure to the SCs code of the 1st level, adapted for the Web);

- a translator from the SCsJson format to the SC code;

- a translator from the SCs code to the SC code;

- a translator from the SCg code to the SC code;

- a translator from the SC code to the SCg code.

The entire metaknowledge base of the user interface is divided into sections as described below:

- user description – a section of the knowledge base, which is information about all registered users of the system (preferred natural language, various settings, protocols, user activity, etc.);

- description of the syntax and semantics of all external languages – a section of the knowledge base, which contains a formal description of all external knowledge representation languages that are used in dialogue with the user [16]. Examples of such descriptions are: SCg code description, SCs code, and SCn code;

- description of the components used – a section of the knowledge base, which contains all the elements of the user interface used in the system;

- description of user commands – a section of the knowledge base, which contains descriptions of commands with which the user has the ability to interact with the system;

- description of the principles of the user interface – a section of the knowledge base, which contains all the reference information about the user interface.

The Abstract Knowledge Processing Machine consists of the following SC Agents (SC Additives that are part of the Broadcast and Display Components):

- SC command instance generation agent, in general description;

- all user interface commands are described in the knowledge base and contain any layout (template), on the basis of which the internal agent creates a copy of this command with its arguments replaced;

- translation SC agent launching responses to user requests. This SC agent waits for a response to a user request and creates a copy of the command that will start broadcasting the response in an external language;

- SC garbage collection agent. This SC agent searches for and deletes information that has been copied or used as a result of changes. This information may have already completed the broadcast command, and so on and so on;

- SC to collect agent IDs. This SC agent collects all identified SC elements in order to provide a quick search for elements by ID.

Thus, the IMS.OSTIS metasystem is an intelligent system that [16, 17]:

- 1) is implemented on the basis of OSTIS technology;

- 2) provides comprehensive support for the development of intelligent systems in accordance with OSTIS technology;

- 3) contains all the skills necessary for this (knowledge of available models, methods and design tools);

- 4) is one of the forms of the material technology of OSTIS embodiment;

- 5) manages automation projects aimed at continuous self-improvement.

In general, IMS.OSTIS can be considered as:

- a portal of knowledge necessary for creators of intelligent systems;

- an intelligent system (environment) for the automated generation of various-purpose intelligent systems;

- an intelligent system that not only provides information support for the design of intelligent systems, but also supports collective management in the design of these systems (including itself).

To describe and analyze semantic networks in OSTIS (Open Semantic Technology for Intelligent Systems) technology, a group of graphical languages is used, and the texts of which are SC constructs. Based on the SC code, a large amount of knowledge of presentation languages has been implemented, aimed at the presentation of specific subject areas.

For knowledge processing, with the help of SC (Semantic Code), programming languages are used: SCP (Semantic Code Programming) – the main procedural programming language; SCPH (Semantic Code Programming High-level) – high-level procedural programming language; functional programming language; logic programming languages based on SCL. These two programming languages are also part of the SC language family, as they are implemented on the basis of SC code. Therefore, programs and the data they process are considered SC constructs. The construction of knowledge representation languages and knowledge processing languages according to a single principle makes it possible to represent data and programs in the form of SC constructs, which makes it possible to easily introduce various types of information and accumulate declarative and procedural knowledge in

a single knowledge base. Procedural knowledge is called programs and is knowledge about algorithms for solving various types of problems. Each program has its own class of problems solved with its help. The solution of a specific problem requires the implementation of a program on predetermined initial data using an information machine. Each specific programming language is one of many ways to write programs. At the same time, each of the programming languages uniquely establishes the system of operations of the abstract machine, which ensures the smooth execution of all programs of this language. An information construct located in the memory of an abstract machine and transformed during the execution of a program is called program data or declarative knowledge.

The core of OSTIS technology is the IMS.OSTIS metasystem using the SC language.

OSTIS technology can be interpreted as a set of models, tools and methods used to implement intelligent systems, as well as to constantly update and evolve this technology itself. The OSTIS technology is based on the use of unified semantic networks with a basic set-theoretic interpretation of their elements as a way to describe knowledge. This method of knowledge representation is called SC code (Semantic Code), and the semantic networks represented in the SC code are called SC graphs.

SCP (Semantic Code Programming) is a basic programming language aimed at processing homogeneous semantic networks containing a basic set-theoretic interpretation [18]. The SC code is used to describe knowledge in the form of semantic networks. The SCP language is considered an SC language, i.e. language texts (SCP programs) are described as SC constructs. SCP programs and the information structures (SC constructs) analyzed by them can be located in one knowledge base. The features of the SCP language are [18, 19]:

- orientation to the use of structurally reconfigurable (graph-dynamic) associative memory;
- good adaptation to the analysis of non-numerical structures, which makes it possible to describe complexly structured subject areas;
- focus on processing directly the semantic networks themselves;
- the possibility of forming algorithms for solving problems of any level of complexity;
- the possibility through the concept of an SC node to integrate SCP programs with programs developed in other programming languages and implement the restructuring of information structures;
- high potential of asynchrony of information analysis processes in graphical reconfigurable associative memory.

The operational semantics of the SCP programming language is set by the corresponding abstract machine. An abstract machine is established by an abstract memory, in which the analyzed constructs and clusters of operations are located. All additional data needed by the abstract SCP machine to form SCP programs located in memory are also formed as SC constructs. Abstract memory is a dynamic information construct. The principles of memory change are set by operations, the structure of memory is given by the syntax of the language.

The tool for designing programs of the main programming language aimed at processing unified semantic networks includes the representation of SCP programs in

graphical form (SCg) and in linear form (SCs, M4SCP). The tool used at the moment includes: SCP program editor, source repository build tool, SCP interpreter, SCP debugger with the function of visual representation of the SCP program debugging process. The tool is an intelligent system and is implemented using open semantic technology.

The semantic network provides the ability to generate complex structured information in the form of a graph in order to facilitate the coding of the apparatus using binary relations in the graph.

In other words, it is the basis for such an intelligent portal, which uses the hidden semantic knowledge located on it, will “understand” users’ questions and provide them with a variety of relevant information, in addition to the necessary solution.

SCg code is one of several methods of visual representation of SC texts (text languages or textual semantic code) and SCg graphs. SCg code is an abstract language, it is a rather simple computer code of semantic networks. It can be taken as a fundamental universal element of machine coding of knowledge in the memory of future computers aimed at processing semantic networks [20].

The basic principle included in the basis of the SCg code is that any SC element is associated with an SCg element (graphical display).

So, in the course of creating a knowledge base based on this solution, as well as studying it, a list of its main advantages was compiled:

1. In the representation of the semantic network, you can form any area of knowledge.
2. Openness of the system, the ability to easily change, increase and edit the knowledge base being created.
3. The presence of the SC code functionality, which gives the advantage of regulating the visual massiveness of the semantic network, that is, performing graphical decomposition, for example, any object in the graph can be converted into an identifier, without affecting the semantics of the subject area [21].
4. The ability to store, process, receive or issue any information about the object of knowledge.

SC code is an abstract language, therefore, it is possible to implement various graphic refinements of the SC code (for example, SCg code), many options for displaying SC constructs in the form of character strings (for example, SCs code), multiple variants of machine representation of SC constructs in address memory of standard computers, as well as in a specific structurally reconfigurable associative memory of future computers aimed at processing knowledge bases. The SC code is not a complex computer code of semantic networks, which does not appear to be an “alien” representation of semantic networks, but their description, too, in the form of semantic networks, but as simple as possible – with a minimal alphabet and binary connectives.

SCg language is one of the known ways of visual representation of SC texts or SCg graphs. Most often, the SCg code is used up to the fourth level. To design a domain-specific language, you need to consider all levels of the structure in detail, to specify the entities and relationships that describe the SCg code.

The introduction of copies of the same SC element, that is, by creating multiple occurrences of images of the same character, is one of the additional measures that can

increase the visibility of the SC text, since it means that subcutaneous-like elements can be performed in the SC text [21]. Since the two are synonymous SC elements semantically representing the image of the same character, they can be logically related. More precisely, at the same time, the ability to combine the SC language with synonymous SC elements is not only logically, but also physically connected to this language improved associative properties, i.e. search for necessary information in the ribs.

#### *SCg code of the first level (SCg core).*

The SCg core is one of the most accurate ways to describe SC texts graphically. Each element of the SCg code alphabet is associated with an element of the SCg code alphabet [8]. The texts formed by means of the SCg core are applicable for illustrating the syntactic and semantic properties of the SCg code, but are not convenient for wide practical use as a visualization language for SC texts, because they are not visual. The development of the SCg code just solves this contradiction. Appendix A presents the alphabet of the SCg language.

Some of the SCg elements can be assigned string identifiers, which are the names of the displayed SC elements (how these identifiers are formatted in the knowledge base will be described later). This is necessary for the user to read SCg texts. These identifiers can be:

1) global (system), which are considered absolutely unique within the system. Basically, these identifiers are used by the system itself and its programmers;

2) the main ones, which are unique within the framework of the dialogue mode directed to the user using this natural language;

3) additional, representing a variety of synonyms.

#### *Second-level SCg code.*

The second level of the SCg code is aimed at optimizing the display on SCg screen elements in order to improve the visibility of SCg texts [9]. This essential SCg code of the second level is the fact that the image is the most commonly used accessory arcs of SC elements in standard (basic) SC sets, can be exchanged for a special characteristic for them, a graphic element.

Many SC nodes, with the exception of division based on persistence, are divided into the following subsets, on a structural basis:

– a set of SC nodes that are not binary bundles; a set of subcutaneous nodules is a structure; a set of SC nodes, role relations;

– a set of SC nodes that are non-role relations; a set of SC nodes, which are abstract objects that are not sets;

– a set of SC nodes, which are material objects.

Membership of an SC site to one of the sets described above is done with specific graphic elements that are inside the figure, depicting a constant or variable SC site.

*SCg code of the third level.* With significant volumes of SCg texts, the second level can also be low. As with the transition from level 1 to level 2, improving the readability of the SCg text can get an even more significant reduction in the depicted elements. For this, the 3rd level of the SCg code has been created. There are cases when an SCg arc (pair) leaves an SCg node (or a large number of SCg arcs it includes). In this case, the SCg text becomes difficult to read, because next to the SCg node, an SCg set of arcs is drawn, which can close the text identifier [15].

#### *SCg code of the fourth level.*

SCg code is not the only way to visually design SC texts, but is also used in the graphical user interface to implement the user's dialogue with the system. It is possible that the user will conduct the conversation using the first three levels (the user manually draws messages). But to increase the speed of the system at the SCg code fourth level there are elements of SCg control [8, 9]. Clicking on an object of this type will start the action to which it is attached.

#### *Creation of language metastructures.*

The metalanguage model is the semantic structure of the network, that is, to create a knowledge base of the technology, the OSTIS schema, unlike other technologies, not only refers to the concept, which will be the domain structure, but also the structure. The knowledge base is built in such a way that the whole set of species is in general terms, since it is one of the fundamental principles of technology, the introduction of knowledge integration in the knowledge base and knowledge base integration. In this technology, the SCg code, a visual representation of the SC code, acts as such a form. Moreover, the alphabet of all SC languages is similar to the alphabet of the SC code. The SC code is a unity of language and metalanguage. So, for example, in the form of SC constructs, one can describe the syntax, semantics and ontology of the SC code. From a formal point of view, an SC code can be defined as a metalanguage of the basic semantic specification of SC elements using a precise set of key SC code nodes. Due to the fact that this work relies on a graphical representation of data, for a clearer understanding between them, you can put the identity. Therefore, the SC language can be used as a metalanguage for decorating any external language, i.e. a language whose texts are not SC constructs. In the model, system identifiers and ordinary groups of objects are formed with such a primitive. Conference information will be displayed in the knowledge base as a system identifier containing information associated with it. All system identifiers are needed so that the system can combine identical nodes using them. System identifiers are unique. In other words, there cannot be two identical nodes with the same ID. If such nodes are present in the SCg text, this means that they are attached to the same object.

SC link is an SC element that defines a specific file that can be viewed or some computer system that can be interacted with.

SCs code is a string (linear) variant of SC code design. It is used to describe SC graphs (SC code texts) in the form of character strings that can be changed both with standard text editors and with a special SCs editor. This means that one of the requirements for the SCs code, in addition to completeness and consistency, is the availability of a set of source codes without the help of a specialized editor, using only symbols of a standard keyboard [8, 9].

The SCs code is conditionally divided into seven levels of complexity. All levels are the same in terms of the possibilities of describing knowledge, but the texts of higher levels draw up SC graphs more specifically. The introduction of levels is intended to improve the work of knowledge base programmers when creating SCs texts. When implementing knowledge bases, it is recommended to use levels up to the sixth.

To implement the online course "Algorithms and data structures", SCs code of the sixth level, HTML, jQuery, CSS is used. Conventionally, the implementation process can be divided into 6 stages:

1. Creation of nodes.
2. Creation of links between nodes.
3. Creation of online course content in HTML format.
4. Creation of cross-references for secondary nodes.
5. Creation of test elements with jQuery.
6. Assembly of the “Glossary” section.
7. Assembly of the “Final test” section.

#### *1 stage of implementation. Creation of nodes.*

The creation of nodes in the OSTIS system is implemented using the SCs code. First of all, it is required, based on the structure of the textbook, to divide the material into separate parts, which will be the nodes [15]. The main nodes of the tutorial will be the content elements. These are: abstract, introduction, lectures 1–19, laboratory work, final test, examination questions, glossary, instructions for use, list of references. Secondary nodes will be the main theses contained in the material. For example, terms like array, deque, stack, table, lookup, etc.

Each node in the OSTIS system is described in a file with the .scs extension and is described using the SCs code.

The data for creating the end nodes of the tutorial are given in appendices B-D.

#### *2 stage of implementation. Creation of links between nodes.*

After creating the nodes, you need to combine them into a single semantic network, that is, set up links between the content elements of the tutorial (lectures, terms, etc.).

#### *3 stage of implementation. Creation of online course content in HTML format.*

To optimize the display of content and eliminate the need for third-party software, the tutorial uses HTML files. To create these files, any text editor is used, and the standard HTML markup language.

In this framework, between the body tags, you need to place the content corresponding to the node and format it for acceptable display to the user using the <p>, <h1>, <img> tags, etc. Upon completion of the creation of HTML files for the corresponding nodes, the OSTIS kernel system will check each node for errors at startup and, if the result is positive, at this stage will begin to correctly display the tutorial in accordance.

#### *4 stage of implementation. Creation of cross-references for secondary nodes.*

To create cross-references to the nodes of the system in the HTML pages of the tutorial, we can use the system-defined tag of the form:

```
<sc elementsys_idtf= "System Node ID">Text</sc element>.
```

This tag is automatically read by the system and when loading the nodes of the tutorial, OSTIS gives the browser in the HTML code a link in the form:

```
<a href="#" class="sc-element" s_addr="Node Numeric Equivalent">Text</a>.
```

#### *5 stage of implementation. Creation of test elements with jQuery.*

Since the tutorial uses HTML pages, you can extend their functionality with jQuery and JavaScript. To do this, you need to connect a file containing the jQuery library to the page using the construction of the form:

```
<script type="text/javascript" src="jquery-2.1.1.min.js"></script>.
```

And place the library itself in the same directory where the page we need is located.

The next step is to create a framework for the test questions. To do this, you need to create a block in order to put all the questions there, a separate block for each question of the test, headings that contain the question, and lists that contain the answer options. For each of these elements, we will set classes for further styling using CSS.

#### *6 stage of implementation. Assembly of the “Glossary” section.*

The glossary section is a node that contains all the terms included in the lecture. To do this, all secondary nodes available to the system must be added to this node using the construction:

```
<= nrel_section_decomposition.
```

#### *7 stage of implementation. Assembly of the “Final test” section.*

### **5. 2. Application of the machine learning method for the analysis of learning outcomes (analysis of test results)**

This study considers an example of the practical implementation of the method of intelligent processing and obtaining of data on the results of testing students in order to assign them to the letter categories A, B, C, D.

The method of intelligent data processing is implemented using the machine learning method: classification. To analyze the test results, the Python library was used: Pandas.

Pandas is a high-level Python library for data analysis. The dataset consists of student grades in letter grades: A, B, C, D, and in points: from 50 to 100 points.

The target variable will be evaluated in groups A (excellent), B (good), C and D (satisfactory). Accordingly, our task is to use the available data to try to find the relationship between the letter designations and average score values: A, B, C, D.

We use the Python library – Matplotlib, because it efficiently displays student test results in the form of bar charts. A bar chart is a simple way to present tabular data graphically. Method of intelligent data processing and visualization. Based on the results of training, students are tested and the test results can be analyzed and processed in order to classify them into letter categories A, B, C, D according to a set of features: scores and GPA scores. Classification is the assignment of objects to a certain class according to a set of features.

To do this, you can use the method of intelligent data processing – classification.

For analysis, let's take the test results of one group of students. Next, we create two arrays with letter grades and GPA values relative to the values in points from 50 to 100, as shown in Fig. 1.

Next, we create two arrays, which we will fill with letter grades and GPA values, as shown in Fig. 2.

Correspondence of performance indicators and the average score relative to the values in points is shown in Fig. 3.



Out[2]:

	Name	Evaluation
0	Айыпов Еркебұлан Ермұратұлы	90
1	Аргимбеков Арнур Аканович	86
2	Әди Исламхан Ғалымжанұлы	90
3	Бекқожин Дастан Ақанұлы	93
4	Бердіқанов Марлен Нұртасұлы	87
5	Ботаев Шахдат Калдыбайұлы	55
6	Ермекбай Ерсултан Серикбайұлы	88
7	Кудайберген Асылжан Бектемирұлы	91
8	Қайратқызы Гүлсән	88
9	Қанибай Мәди Фаниұлы	92
10	Қыдыров Ислам Асланбекұлы	91
11	Мағзұмов Сағадат Талғатұлы	85
12	Мырзағали Нұрсерік Әсетұлы	91
13	Мырзахмет Дәулет Жолдасбекұлы	87
14	Ниязғали Мадияр Айдынулы	93
15	Оңдасынов Нұрдаулет Фаниұлы	93
16	Сарқытбеков (Еркинулы) Айбек Еркинович	91
17	Сыздық Аңсар	89
18	Тайжанов Азамат Жанкелдіұлы	93
19	Төреханұлы Нұрдаулет	85
20	Турсынхан Аяғоз Нурланқызы	88

Fig. 1. Test results data

```

2 worded = []
3 digital = []
4 for index, row in df.iterrows():
5     if (row.Evaluation >= 0 and row.Evaluation < 25):
6         worded.append('D')
7         digital.append(0)
8     elif row.Evaluation >= 25 and row.Evaluation < 50:
9         worded.append('D')
10        digital.append(0.5)
11
12    elif row.Evaluation >= 50 and row.Evaluation < 55:
13        worded.append('D')
14        digital.append(1.0)
15
16    elif row.Evaluation >= 55 and row.Evaluation < 60:
17        worded.append('D')
18        digital.append(1.33)
19
20    elif row.Evaluation >= 60 and row.Evaluation < 65:
21        worded.append('C')
22        digital.append(1.67)
23
24    elif row.Evaluation >= 66 and row.Evaluation < 70:
25        worded.append('C')
26        digital.append(2.0)
27
28    elif row.Evaluation >= 70 and row.Evaluation < 75:
29        worded.append('B')
30        digital.append(2.33)
31
32    elif row.Evaluation >= 75 and row.Evaluation < 80:
33        worded.append('B')
34        digital.append(2.67)
35
36    elif row.Evaluation >= 80 and row.Evaluation < 85:
37        worded.append('B')
38        digital.append(3.0)
39
40    elif row.Evaluation >= 85 and row.Evaluation < 90:
41        worded.append('B')
42        digital.append(3.33)
43
44    elif row.Evaluation >= 90 and row.Evaluation < 95:
45        worded.append('A')
46        digital.append(3.67)
47
48    elif row.Evaluation >= 95 and row.Evaluation <= 100:
49        worded.append('A')
50        digital.append(4.0)

```

Fig. 2. Filling two arrays

The application of the machine learning method – classification, made it possible to analyze and classify training data into categories of letter grades A, B, C, D.

In [4]:

1	df2
---	-----

Out[4]:

	Name	Evaluation	Worded_evaluation	GPA
0	Айыпов Еркебұлан Ермұратұлы	90	A	3.67
1	Аргимбеков Арнур Аканович	86	B	3.33
2	Әди Исламхан Ғалымжанұлы	90	A	3.67
3	Бекқожин Дастан Ақанұлы	93	A	3.67
4	Бердіқанов Марлен Нұртасұлы	87	B	3.33
5	Ботаев Шахдат Калдыбайұлы	55	D	1.33
6	Ермекбай Ерсултан Серикбайұлы	88	B	3.33
7	Кудайберген Асылжан Бектемирұлы	91	A	3.67
8	Қайратқызы Гүлсән	88	B	3.33
9	Қанибай Мәди Фаниұлы	92	A	3.67
10	Қыдыров Ислам Асланбекұлы	91	A	3.67
11	Мағзұмов Сағадат Талғатұлы	85	B	3.33
12	Мырзағали Нұрсерік Әсетұлы	91	A	3.67
13	Мырзахмет Дәулет Жолдасбекұлы	87	B	3.33
14	Ниязғали Мадияр Айдынулы	93	A	3.67
15	Оңдасынов Нұрдаулет Фаниұлы	93	A	3.67
16	Сарқытбеков (Еркинулы) Айбек Еркинович	91	A	3.67
17	Сыздық Аңсар	89	B	3.33
18	Тайжанов Азамат Жанкелдіұлы	93	A	3.67
19	Төреханұлы Нұрдаулет	85	B	3.33
20	Турсынхан Аяғоз Нурланқызы	88	B	3.33

Fig. 3. Correspondence of academic performance and GPA values relative to the values in points

### 5.3. Visualization of the results of the training data analysis

The results of the training data analysis are visualized. To implement the visualization, the Python library – Matplotlib was used.

Method of intelligent data processing and visualization. Methodology for the sequence of implementation of the stages of processing and visualization of training data:

- 1) Prepare training data for processing. To do this, it is necessary to form arrays with the results (testing, exam, etc.) in numbers and in literal equivalent.
- 2) Next, we create two arrays, which we will fill with letter grades, academic performance and GPA value.
- 3) Import the Pandas, Matplotlib machine learning libraries.
- 4) Pandas is a high-level Python library for data analysis.
- 5) We use the Matplotlib library in Python to visually display the results of the analysis of test scores in the form of a bar chart.

Fig. 4 shows the results of data analysis by the number of letter values, i.e. 11 A scores, 9 B scores, 1 D score, no C scores. This method allows you to visually analyze the success of mastering the sections of the course and the course as a whole.

This paper provided examples of designing an online course in the discipline “Web programming” using OLAT. An example of the implementation of the online course “Web Programming” on the Learnme, Articulate Storyline platforms is given. An example of the implementation of an online course using Moodle is shown. The OSTIS technology is described as an IMS metasytem that allows you to develop an online course. An example of the implementation of the online course “Algorithms and Data Structures” using the SC language of the IMS.OSTIS metasytem is given.

The capabilities of the data mining method were also shown based on the results of group testing using data mining methods.

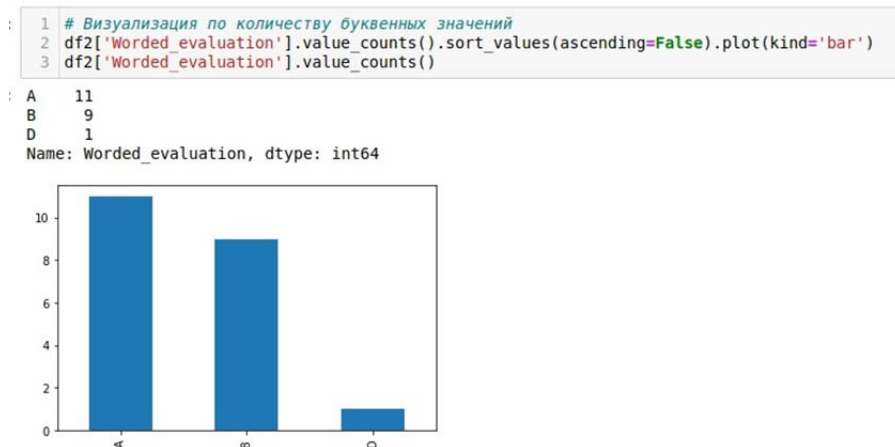


Fig. 4. Visualization by the number of letter values

### 6. Discussion of the results of the study of intelligent semantic technology

An online course based on the IMS.OSTIS intelligent metasystem using the SC graphical language was implemented, using a machine learning method for classifying test results and a visualization method for student test results.

The peculiarity of our study is that we used the machine learning method for classifying training data – classification, and for its implementation we used the Pandas library, which is used for data analysis and machine learning. For example, reference [1] reports the use of a popular decision tree classification algorithm. The authors of [2] describe a model that reflects the interrelated factors of students' acceptance of the e-learning system, but does not present technologies and methods for analyzing and processing educational data that reflect the success of education.

For example, in [3], the authors used learning analytics and educational big data, but do not describe in detail the method of classifying educational data. In [4], the authors give the results of matching existing course objectives with assessment activities, but did not provide methods for classifying grades.

In [5], the authors used multiple regression analysis of data before and after testing, but did not describe the method in detail. The paper [6] does not present the technology for implementing online courses and does not show the technology for classifying grades, since the work is devoted to the development of multimedia content. In [7], the authors describe ast2vec, a neural network that converts Python syntax trees to vectors and vice versa.

The peculiarity of the implemented online learning is that the online course was developed on the basis of the IMS.OSTIS semantic technology using a graphical language. The implementation of the online course includes the following steps: creation of nodes consisting of elements of educational content (lectures, laboratory work, etc.); creation of links between nodes; creation of online course content in HTML format; creation of cross-references for secondary nodes; creation of test elements using HTML, jQuery, CSS; assembly of the “Glossary” section; assembly of the “Final test” section. Also, the peculiarity of this study is the practical implementation of intelligent analysis and

visualization of training data (test results) using the machine learning method (classification).

Possible restrictions on the use of the IMS.OSTIS metasystem are that in order to implement the described online training, you need the OSTIS kernel, a machine with the Ubuntu operating system installed, a GNU/Linux distribution based on Debian GNU/Linux. Also, Compute Cloud – Yandex Data Science Virtual Machine (DSVM), a virtual machine with pre-installed popular libraries for data analysis and machine learning, is required.

For use on a local network, you must configure a static IP address for the machine with the OSTIS kernel installed.

To implement the machine learning method – classification, you must use the Python library – Pandas, as shown in Fig. 2, 3.

The results of the training data analysis are visualized as shown in Fig. 4. To implement visualization, you must use the Matplotlib library in Python.

The disadvantages of this study include a rather voluminous work on the construction of nodes and links between them. In the future, this can be eliminated by using an already created online course as a template for creating other courses, replacing only the content of the educational material.

When using the results of the study, one may encounter difficulties associated with changing the number of nodes and links among them for a particular discipline. Also, some difficulty can be considered the primary processing of training data for their classification and visualization.

The results of the study are based on the use of intelligent open semantic technology for the development of intelligent learning systems – the IMS.OSTIS metasystem using the SC graphical language.

Features of the proposed technology. IMS.OSTIS is a complete semantic model of an intelligent system (the SC model of the knowledge base of an intelligent system, it is designed as a connected sign construction belonging to the SC code – the basic language of the internal semantic representation of knowledge in the memory of OSTIS systems); the basic universal interpreter of the semantic model of an intelligent system stored in its memory. SC code is a universal language for the internal semantic representation of knowledge in the memory of OSTIS systems, it is the basic language of agent-oriented programming, focused on the processing of sign constructions.

Limitations of the study. The OSTIS kernel requires a machine with an installed Ubuntu operating system, a GNU/Linux distribution based on Debian GNU/Linux. It is an operating system based on the Linux kernel and is ideal for use on personal computers, laptops and servers. Also, Compute Cloud – Yandex Data Science Virtual Machine (DSVM), a virtual machine with pre-installed popular libraries for data analysis and machine learning, is required. DSVM can be used as an environment to train models and experiment with data. Managed Service for MySQL helps you deploy and maintain clusters of MySQL

servers. The service supports MySQL 5.7 and MySQL 8; Container Optimized Image 2.3.7 GPU – a virtual machine image optimized for running Docker containers, data analysis and machine learning. The image includes: Ubuntu LTS, Docker and a daemon to run Docker; GitLab.

The disadvantage of this study is rather high technical requirements for hardware and software for the development of a training system.

The development of this research: you can use this technology to develop a “shell template” for loading various contents in different disciplines into it.

The difficulty you may face in the analysis and visualization of educational data is the laboriousness of generating input educational data. They need to be formed as arrays consisting of surnames, names of students by study groups and test results, an exam in the form of points, letter equivalents and GPA.

Possible effects of the study results: mass distribution of the method of organizing online courses using the IMS.OSTIS metasytem with the help of the SC graphical language, the method of analyzing and visualizing training data using the described machine learning method – classification using high-level libraries.

---

## 7. Conclusions

---

1. An online course based on the intelligent metasytem IMS.OSTIS was implemented. We used the syntax and semantics of the SCP (Semantic Code Programming) language, which is a sublanguage of the SC code and is the main agent-oriented programming language focused on processing character structures belonging to the SC code. The SC graphical language was used, consisting of the SC, SCn, SCg codes. The user interface is based on the SCg code. The user interface is a metasytem. It is built with a bean method.

Each component consists of a knowledge base and a set of SC agent fragments. The abstract knowledge processing machine consists of the following SC agents. The SCg code is one of several methods of visual representation of SC texts and SCg graphs. We used the IMS.OSTIS.NET metasytem, which includes the following components of the SCg code: SC construct, SC element, SC constant, SC variable, SC alphabet. The nodes form educational components (lectures, laboratory, tests, etc.), and the nodes are connected in a single semantic network of the educational content of the online course. Connections between nodes are created using decomposition.

As a result, an online course based on the IMS.OSTIS metasytem was implemented, and for the OSTIS core to work, we needed a machine with the Ubuntu operating system installed, a GNU/Linux distribution based on Debian GNU/Linux.

2. A machine learning method was applied to analyze learning outcomes (analysis of test results).

For data analysis (test results of students), test results were used. The result of their processing, in order to classify them into letter categories A, B, C, D according to a set of features: scores and GPA scores. To do this, we used a machine learning method – classification. Classification is the assignment of objects to a certain class according to a set of features. The data set consists of student grades of letter types: A, B, C, D and in points: from 50 to 100 points. An array was formed from the surnames, names, patronymics of students, grades in points, grades in letter equivalent and according to GPA. These indicators are further classified. The target variable will be graded into groups A (excellent), B (good), C and D (satisfactory). Our task was to use the available data to try to find the relationship between the letter grades and the GPA value: A, B, C, D. For this, the Pandas machine learning library was used. Pandas is a high-level Python library for data analysis.

The target variable was assessed in groups A (excellent), B (good), C and D (satisfactory). As a result, we obtained a classification by grades: the number of grades (respectively, the number of students) in groups A, B, C, D. Classification result: A – 11 marks (students), B – 9 marks (students), D – 1 mark (student). There are only 21 students in the group, of which 11 students passed excellent A, 9 students passed good B, 1 student passed D (satisfactory).

3. The results of the training data analysis were visualized (test results of students). To do this, we used the Python library Matplotlib.

We visualized the classification results of students' test scores in the form of a bar chart. A bar chart is a simple way to represent tabular data graphically. Two arrays are created with letter scores and GPA values relative to score values from 50 to 100. A bar chart visualization is obtained.

---

## Conflict of interest

---

The authors declare that they have no conflict of interest in relation to this study, whether financial, personal, copyright or otherwise, that could affect the study and its results presented in this paper.

---

## Financing

---

The study was performed without financial support.

---

## Data availability

---

The manuscript has associated data in a data repository.

---

## References

1. Khor, E. T., Looi, C. K. (2022). A Learning Analytics Approach to Model and Predict Learners' Success in Digital Learning. ASCILITE Publications, 476–480. doi: <https://doi.org/10.14742/apubs.2019.315>
2. Obaid, T., Eneizan, B., Abumandil, M. S. S., Mahmoud, A. Y., Abu-Naser, S. S., Ali, A. A. A. (2022). Factors Affecting Students' Adoption of E-Learning Systems During COVID-19 Pandemic: A Structural Equation Modeling Approach. Lecture Notes in Networks and Systems, 227–242. doi: [https://doi.org/10.1007/978-3-031-16865-9\\_19](https://doi.org/10.1007/978-3-031-16865-9_19)

3. Huang, A. Y. Q., Lu, O. H. T., Yang, S. J. H. (2018). Applying Learning Analytics for the Early Prediction of Students' Academic Performance in Blended Learning. *Educational Technology & Society*, 21 (2), 220–232. Available at: <https://eric.ed.gov/?id=EJ1175301>
4. Barthakur, A., Joksimovic, S., Kovanovic, V., Richey, M., Pardo, A. (2022). Aligning objectives with assessment in online courses: Integrating learning analytics and measurement theory. *Computers & Education*, 190, 104603. doi: <https://doi.org/10.1016/j.compedu.2022.104603>
5. Sukirman, D., Setiawan, B., Riyana, C. (2022). Development of Massive Open Online Courses (MOOC) Content to Improve Indonesian Teachers' Pedagogical Competence: MOOC Technology Instructional Process. *Academic Journal of Interdisciplinary Studies*, 11 (6), 255. doi: <https://doi.org/10.36941/ajis-2022-0166>
6. Abbitt, J. T. (2011). An Investigation of the Relationship between Self-Efficacy Beliefs about Technology Integration and Technological Pedagogical Content Knowledge (TPACK) among Preservice Teachers. *Journal of Digital Learning in Teacher Education*, 27 (4), 134–143. doi: <https://doi.org/10.1080/21532974.2011.10784670>
7. Paaßen, B., McBroom, J., Jeffries, B., Yacef, K., Koprinska, I. (2021). Mapping Python Programs to Vectors using Recursive Neural Encodings. *Journal of Educational Data Mining*, 13 (3), 1–35. Available at: <https://eric.ed.gov/?id=EJ1320641>
8. Kintonova, A. Zh., Nurmukanova, Zh. (2016). Using OSTIS technology to create a knowledge base of an intelligent electronic publication. V International Scientific and Practical Conference "Informatization of Society". Kazakhstan, 226–228.
9. Kintonova, A., Povkhan, I. F., Sabitov, A., Tokkuliyeva, A., Demidchik, N. (2022). Online Learning Technologies. 2022 IEEE 7th International Energy Conference (ENERGYCON). doi: <https://doi.org/10.1109/energycon53164.2022.9830387>
10. Golenkov, V. V., Gulyakina, N. A. (2015). Semantic technology of component design of knowledge-driven systems. OSTIS-2015.
11. Raymond, E. (1999). *The Cathedral & the Bazaar*. O'Reilly Media.
12. Ghandour, A., Kintonova, A., Demidchik, N., Sverdlkova, E. (2021). Solving Tourism Management Challenges by Means of Mobile Augmented Reality Applications. *International Journal of Web-Based Learning and Teaching Technologies*, 16 (6), 1–16. doi: <https://doi.org/10.4018/ijwlts.293280>
13. Gulyakina, N. A., Pivovarchik, O. V. (2011). Integrated methodology for designing and teaching the design of intelligent reference systems. OSTIS-2011.
14. Shunkevich, D. V. (2013). Models and tools for component design of knowledge processing machines based on semantic networks. OSTIS-2013.
15. Caiko, J., Kintonova, A., Mussina, G., Kurmanbayeva, A., Nazarmatova, B. (2022). Modern Models Of Learning At The University. 2022 IEEE 7th International Energy Conference (ENERGYCON). doi: <https://doi.org/10.1109/energycon53164.2022.9830331>
16. Brooker, A., Corrin, L., De Barba, P., Lodge, J., Kennedy, G. (2018). A tale of two MOOCs: How student motivation and participation predict learning outcomes in different MOOCs. *Australasian Journal of Educational Technology*, 34 (1). doi: <https://doi.org/10.14742/ajet.3237>
17. Adomavicius, G., Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17 (6), 734–749. doi: <https://doi.org/10.1109/tkde.2005.99>
18. Fincham, E., Gasevic, D., Jovanovic, J., Pardo, A. (2019). From Study Tactics to Learning Strategies: An Analytical Method for Extracting Interpretable Representations. *IEEE Transactions on Learning Technologies*, 12 (1), 59–72. doi: <https://doi.org/10.1109/tlt.2018.2823317>
19. George, G., Lal, A. M. (2019). Review of ontology-based recommender systems in e-learning. *Computers & Education*, 142, 103642. doi: <https://doi.org/10.1016/j.compedu.2019.103642>
20. Henrie, C. R., Halverson, L. R., Graham, C. R. (2015). Measuring student engagement in technology-mediated learning: A review. *Computers & Education*, 90, 36–53. doi: <https://doi.org/10.1016/j.compedu.2015.09.005>
21. Hsiao, C. C., Huang, J. C. H., Huang, A. Y. Q., Lu, O. H. T., Yin, C. J., Yang, S. J. H. (2018). Exploring the effects of online learning behaviors on short-term and long-term learning outcomes in flipped classrooms. *Interactive Learning Environments*, 27 (8), 1160–1177. doi: <https://doi.org/10.1080/10494820.2018.1522651>