

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
FEDERAL STATE AUTONOMOUS EDUCATIONAL
INSTITUTION OF HIGHER EDUCATION
NORTH CAUCASIAN FEDERAL UNIVERSITY

GUIDELINES
for completing practical classes
by discipline

Fundamentals **of Computational Thinking and Artificial Intelligence**

for students in the field of study
11.03.02 Infocommunication technologies and communication systems
Profile
"Intelligent Data Processing in Infocommunication Systems and Networks"

Stavropol 2026

Content

Introduction

Practical Lesson #1: Fundamentals of Exploratory Data Analysis

Practical Lesson #2 Linear Regression

Practical Lesson #3 Logistic Regression

Practical Lesson #4: Quality Metrics for Classification Models

Practical Lesson #5: Nearest Neighbor and Support Vector Machines

Practical Lesson #6 Decision Trees

Practical Lesson #7: Random Forest

Practical Lesson #8: Gradient Boosting

Conclusion

Educational, methodological and informational support of the discipline

INTRODUCTION 1. The purpose and objectives of mastering the discipline (module)

The purpose of the discipline is to develop, in accordance with the educational program, the competencies of the OPK-4 of the student.

To achieve this goal, the following objectives are addressed in the course: familiarization with the basic concepts and types of machine learning problems; study of data preparation methods, including cleaning, normalization, and feature engineering; mastering basic supervised learning models, such as linear and logistic regression; mastering stochastic gradient descent as a method for optimizing models; forming an understanding of metrics for assessing the quality of classification and interpreting model results; studying methods for combating overfitting, including regularization and cross-validation; mastering classification algorithms based on the nearest neighbor method and the support vector machine; study of algorithms for constructing decision trees and ensemble models, such as random forest and gradient boosting; familiarity with unsupervised learning methods, including clustering algorithms; mastering methods of dimensionality reduction and data visualization, such as PCA and t-SNE; developing practical skills in tuning models using hyperparameter selection; developing the ability to compare and interpret the results of different models based on objective metrics.

2. The place of the discipline (module) in the structure of the educational program

Discipline refers to the part formed by the participants of educational relations.

3. List of planned learning outcomes for the discipline (module), correlated with the planned learning outcomes for the educational program

PRACTICAL LESSON #1

Topic: Fundamentals of Exploratory Data Analysis

Objective:

To learn how to apply basic methods of exploratory data analysis (EDA) using the pandas library, visualize the structure of data, and identify key patterns, gaps, and anomalies in a dataset.

Theoretical rationale:

Exploratory data analysis (EDA) is the first and most important step in working with data, aimed at understanding the structure, properties, and characteristics of a dataset. The goal of EDA is to identify relationships between features, test hypotheses, determine the level of data quality, and provide insights into further analysis or model building.

EDA utilizes a variety of tools, including data type and gap analysis, descriptive statistics, and the creation of histograms, scatterplots, boxplots, correlation heatmaps, and more. Particular attention is paid to visualization, as it allows for the visualization of hidden dependencies, outliers, and errors in the data.

Exploratory analysis tools include functions from `pandas`, `matplotlib`, `seaborn`, and other libraries. Conducting EDA improves the validity of the choice of features, methods, and models used in subsequent stages of machine learning.

Equipment and materials:

Computer or laptop with internet access; Jupyter Notebook or Google Colab; Installed libraries: `pandas`, `matplotlib`, `seaborn`, `numpy`;

CSV file with real or synthetic data (e.g., `titanic.csv` , `iris.csv`)
; Optional: built-in datasets from `seaborn` (e.g., `tips` , `penguins`)

Typical tasks:

1. Load a dataset (e.g. `titanic.csv`), determine its dimensions, feature types, presence of gaps, and statistical characteristics.
2. Create visualizations: histograms for quantitative variables, boxplots for categorical variables, and scatterplots for feature pairs.
3. Construct a heat map of feature correlation and conclude whether there are linear relationships between them.

Test questions:

1. What is EDA and what is its role in data analysis?
2. What methods can be used to determine the presence of missing values?
3. What statistical measures describe the distribution of features?
4. How to visualize the distribution of a quantitative variable?
5. How to interpret a boxplot chart?
6. What are emissions and how can they be detected visually?
7. How to investigate the relationship between two quantitative characteristics?
8. What types of correlation exist and how to interpret them?
9. Why use a heat map and how is it built?
10. How can EDA detect errors in data structure?

PRACTICAL LESSON #2

Topic: Linear Regression

Objective:

To master the basic principles of constructing a linear regression model using the `scikit-learn` library , to learn how to interpret model coefficients and to evaluate its quality using appropriate metrics.

Theoretical background:

Linear regression is one of the simplest and most common machine learning methods. It is used to model the relationship between one or more independent variables (features) and a dependent variable (target variable), assuming this relationship is linear.

The linear regression model has the form:

$$y = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n ,$$

where w are the model coefficients determined during the training process.

The main stages of model construction are: dividing the sample into training and test parts , training the model on the training data, predicting the target variable values on the test data, and assessing the prediction accuracy. The main metrics for assessing model quality are the mean square error (MSE), mean absolute error (MAE), and the coefficient of determination (R^2).

Linear regression is widely used in forecasting, trend analysis, and interpreting relationships between variables. It is important to be able to interpret the model's coefficients, evaluate its generalization ability, and identify potential problems such as multicollinearity or overfitting.

Equipment and materials:

Computer or laptop with internet access; Jupyter Notebook or Google Colab; Installed libraries:

`pandas ,numpy ,scikit-learn ,matplotlib ,seaborn;`

Dataset with numerical features and a continuous target variable (e.g., Boston , California housing , insurance.csv)

Typical tasks:

1. Load the dataset, extract features and the target variable, and split the data into training and test sets.
2. Build a linear regression model using `LinearRegression` , print the coefficients and interpret them.
3. Evaluate model quality using MSE, MAE, and R^2 metrics, and visualize predictions on test data.

Test questions:

1. What does linear regression model and what is its main assumption?
2. How are linear regression coefficients interpreted?
3. What are the steps involved in constructing a linear regression model?
4. How is data divided into training and test samples?
5. What are MSE, MAE and R^2 and how to interpret them?
6. What does the coefficient of determination R^2 show?
7. How to visually assess the quality of a regression model?
8. What are some signs that a model may be overfitting?
9. What are the advantages and limitations of linear regression?
10. How can a model be improved when there is a weak linear relationship?

PRACTICAL LESSON #3

Topic: Logistic Regression

Objective:

To study the principles of constructing a logistic regression model for binary classification problems, learn how to train a model using the `scikit-learn` library , interpret coefficients, and evaluate its quality using the accuracy metric.

Theoretical background:

Logistic regression is a basic classification algorithm used to predict the probability of an object belonging to one of two classes. The model uses a logistic (sigmoid) function, which transforms a linear combination of input features into a value between 0 and 1, which is interpreted as the probability of assigning the object to the positive class.

The model's decision rule is based on a threshold: if the probability exceeds a specified threshold (0.5 by default), the object is classified as positive; otherwise, it is classified as negative. The model's coefficients reflect the influence of the corresponding features on the logarithm of the odds ratio (log-odds).

During training, the model selects coefficients by maximizing the log-likelihood function. After training, the model can be used to predict classes of new observations. The main quality criterion in this work is accuracy, which reflects the proportion of correctly classified objects.

Logistic regression is widely used in various applications, including medical diagnostics, credit scoring, and marketing, due to its interpretability and performance on small and medium-sized samples.

Equipment and materials:

Computer or laptop with internet access

Jupyter Notebook or Google Colab
Installed libraries: `pandas` , `numpy` , `scikit-learn` , `matplotlib` , `seaborn`

Dataset for binary classification (e.g. `breast_cancer` from `sklearn.datasets` , `titanic.csv` , `diabetes.csv`)

Typical tasks:

1. Load the dataset and extract features and the target variable. Split the data into training and test sets.
2. Build a logistic regression model using `LogisticRegression` , train it, and print the coefficients.
3. Obtain predictions on the test set and calculate the model accuracy using the `accuracy_score` metric .

Test questions:

1. What is the basic idea of logistic regression?
2. How does the sigmoid function work and why is it used?
3. What data is suitable for logistic regression?
4. How are the coefficients of a logistic regression model interpreted?
5. What does the classification threshold of 0.5 mean and how can it be changed?
6. How is the logistic model prediction formed?
7. What is accuracy and how is it calculated?
8. Why is it important to split data into training and test sets?
9. How does feature scaling affect the performance of logistic regression?
10. What are the limitations of logistic regression when working with real data?

PRACTICAL LESSON #4

Topic: Quality metrics for classification models

Objective of the work:

To become familiar with the main metrics for assessing the quality of classification models, to learn how to calculate and interpret accuracy, recall, prediction accuracy, F1-measure and to build a confusion matrix using the `scikit-learn` library .

Theoretical rationale:

Assessing the quality of classification models is essential for understanding how well a model performs in class prediction. Using only one metric, such as accuracy , may be insufficient, especially in problems with imbalanced classes, where most objects belong to a single class. The following metrics are used for a comprehensive assessment of classification quality:

- **Accuracy** is the proportion of correct predictions from the total number of observations.
- **Recall** is the proportion of positive class objects correctly predicted by the model.

- **Prediction accuracy** is the proportion of correct positive predictions among all predictions of the positive class.
- **The F1 measure** is the harmonic mean between precision and recall and is used when it is necessary to balance both metrics.
- **A confusion matrix** is a table showing the ratio of true and false positive and negative classifications.

Understanding and correctly interpreting these metrics allows for objective comparison of models and informed choice of the appropriate algorithm.

Equipment and materials:

Computer or laptop with internet access, Jupyter Notebook or Google Colab, Installed libraries: `pandas` , `numpy` , `scikit-learn` , `matplotlib` , `seaborn`

Binary classification dataset (e.g., `breast_cancer` , `titanic.csv` , `diabetes.csv`)

Typical tasks:

1. Train a logistic regression or decision tree model on a binary dataset and obtain predictions.
2. Build a confusion matrix and visualize it using `seaborn.heatmap()` .
3. Calculate and analyze accuracy, precision, recall, and F1 score using `classification_report()` .

Test questions:

1. What is a confusion matrix and what values does it show?
2. How is accuracy calculated?
3. What is the difference between precision and recall?
4. What does the F1 score measure and when is it particularly useful?
5. Why might accuracy be a poor metric for imbalanced classes?
6. How to interpret high precision and low recall?
7. What shows `classification_report()` from `sklearn.metrics` ?
8. How to visualize a confusion matrix?
9. Which metric is more important—precision or recall—and what does it depend on?
10. How to compare two models across multiple metrics at once?

PRACTICAL LESSON #5

Topic: Nearest Neighbor and Support Vector Machines

Objective:

To study the operating principles of the k-nearest neighbors (kNN) algorithm and the support vector machine (SVM), learn how to apply them to classification problems using the `scikit-learn` library , configure model parameters and evaluate their accuracy.

Theoretical rationale:

The k-nearest neighbors method is a lazy learning algorithm: the model is not built in advance , and predictions are made based on the distance to the nearest objects in the training set. The algorithm is easy to implement, requires no training, but is sensitive to feature scaling and the choice of the number of neighbors.

The Support Vector Machine (SVM) is a more complex algorithm. It searches for a hyperplane that maximally separates classes, taking into account the maximum possible margin. Using kernel functions, the method can solve both linear and nonlinear classification problems. SVM is sensitive to the choice of parameters, especially `C` and the kernel type , and requires pre-scaling of the data.

Both methods are used in classification and recognition problems, work well on problems with a moderate number of features and observations, but require careful tuning and evaluation.

Equipment and materials:

Computer or laptop with internet access, Jupyter Notebook or Google Colab, Installed libraries: `scikit-learn` , `pandas` , `numpy` , `matplotlib` , `seaborn` ,

Datasets: `iris` , `digits` , `wine` from `sklearn.datasets` or another set for classification

Typical tasks:

1. Load the selected dataset, perform feature scaling using `StandardScaler` .
2. Build a `KNeighborsClassifier` model , train it, find the value of `k` (number of neighbors), calculate the accuracy.
3. Build an `SVC` model with different kernels (`linear` , `rbf`), tune the parameter `C` and compare the accuracy of the models.

Test questions:

1. How does the k-nearest neighbors method work?
2. What does the parameter `k` mean and how does it affect the behavior of the model?
3. Why is data scaling important for kNN and SVM?
4. How does a support vector machine work and what is a separating hyperplane?
5. What is a kernel in SVM and what types of kernels do you know?
6. How does parameter `C` affect the generalization ability of the SVM model?
7. When is it preferable to use kNN and when to use SVM?
8. How to evaluate the quality of models on a test set?
9. What are the performance benefits of kNN?
10. How to interpret class boundaries obtained from an SVM model?

PRACTICAL LESSON #6

Topic: Decision Trees

Objective:

To master the basic principles of constructing and analyzing classification and regression models based on decision trees, learn to visualize the tree, interpret the structure of the model and evaluate its accuracy.

Theoretical background:

Decision trees are interpretable and intuitive models applicable to both classification and regression problems. The basic idea of the method is to sequentially partition the feature space into regions in which a prediction is made based on the dominant class or mean.

During the tree construction process, it selects features and threshold values that provide the best split of the sample based on the target feature. The main criteria for split quality are the Gini index, entropy, and variance. The tree depth, the minimum number of objects per node, and other parameters regulate the model's complexity and prevent overfitting.

The tree model is easy to visualize and interpret: each path from root to leaf can be interpreted as a set of logical conditions that determine a decision. However, trees are prone to overfitting, especially on small samples, and are sensitive to noise.

Equipment and materials: Computer or laptop with

internet

access Jupyter Notebook or Google Colab Installed libraries: `scikit-learn` , `pandas` ,

`numpy` , `matplotlib` , `seaborn` , `graphviz` (optional)

Datasets: `iris` , `titanic` , `wine` or any other set with categorical and numerical features

Typical tasks:

1. Build a classification model using `DecisionTreeClassifier` , train it, and print the accuracy on the test set.
2. Visualize the tree structure using `plot_tree()` or `export_graphviz()` , analyze the features used and the tree levels.
3. Investigate the impact of tree depth limitation (`max_depth`) on model quality.

Test questions:

1. How does the decision tree algorithm work?
2. What criteria are used to evaluate the partitioning of data in tree nodes?
3. What is overfitting and how does it manifest itself in decision trees?
4. How to interpret tree structure?
5. What does tree depth mean and how does it affect the model?
6. What parameters allow you to control the complexity of the tree?
7. Why are decision trees sensitive to small changes in the data?
8. How is different `DecisionTreeClassifier` from `DecisionTreeRegressor` ?
9. How to visualize a tree and why is it necessary?
10. What are the advantages and limitations of decision trees compared to other models?

PRACTICAL LESSON #7

Subject: Random Forest

Objective:

To study the operating principle of the ensemble random forest method, learn to build classification and regression models using `RandomForestClassifier` and `RandomForestRegressor` , interpret the importance of features and evaluate the accuracy of the model.

Theoretical background:

Random Forest is an ensemble algorithm based on constructing multiple decision trees, each of which is trained on a random subsample of the input data and using a random subset of features. The final prediction is formed by voting (for classification) or averaging (for regression) across all trees.

The algorithm combines the simplicity and interpretability of decision trees with high robustness and generalization ability, reducing the risk of overfitting. It performs well on problems with a large number of features and observations, does not require data scaling, and can be used for both classification and regression.

One of the useful properties of random forests is the ability to evaluate the importance of features based on their participation in splits within the trees. This allows for feature selection and model simplification without significant loss of accuracy.

Equipment and materials:

Computer or laptop with internet access Jupyter Notebook or Google Colab

Installed libraries: `scikit-learn` , `pandas` , `numpy` , `matplotlib` , `seaborn`

Datasets: `breast_cancer` , `wine` , `titanic` , `diabetes` , or any CSV file with class labels

Typical tasks:

1. Train the `RandomForestClassifier` model on the selected dataset and calculate the accuracy on the test set.
2. Plot the feature importance plot using `feature_importances_` and visualize it.
3. Investigate the impact of the number of trees (`n_estimators`) and depth (`max_depth`) on the quality of the model.

Test questions:

1. What is the idea behind the random forest method?
2. How is a random forest different from a single decision tree?
3. How are individual trees trained in a random forest?
4. What does the `n_estimators` parameter mean ?
5. What features are selected when training each tree?
6. How is the final prediction calculated in a random forest?
7. How is feature importance determined in a model?
8. What random forest parameters affect overfitting?
9. What advantages does random forest have over other classification models?
10. For what problems is random forest particularly effective?

PRACTICAL LESSON #8

Topic: Gradient Boosting

Objective:

To study the principles of the gradient boosting method, learn how to train a model using the `GradientBoostingClassifier` from the `scikit-learn` library , evaluate accuracy, and analyze the influence of model parameters on its quality.

Theoretical background:

Gradient boosting is an ensemble method that constructs a sequence of weak models (usually decision trees), each learning from the errors of the previous one. Each subsequent tree adjusts the model's predictions, seeking to minimize a given loss function.

A distinctive feature of gradient boosting is its high accuracy and ability to detect complex nonlinear relationships. The algorithm is sensitive to the choice of parameters, such as the number of trees (`n_estimators`), learning rate (`learning_rate`), and tree depth (`max_depth`). When properly configured, it can outperform simpler models, but requires regular validation to prevent overfitting.

Gradient boosting is used in a wide range of problems, including credit scoring, customer churn analysis, medical diagnostics, and machine learning competitions.

Equipment and materials:

Computer or laptop with internet access Jupyter Notebook or Google Colab Installed libraries: `scikit-learn` , `pandas` , `numpy` , `matplotlib` , `seaborn`

Dataset: `breast_cancer` , `titanic` , `wine` , or your own CSV file for classification

Typical tasks:

1. Build a `GradientBoostingClassifier` model on the selected dataset, train it, and calculate the accuracy on the test set.
2. Change the values of the `n_estimators` , `learning_rate` , and `max_depth` parameters and analyze how they affect the accuracy of the model.
3. Plot a feature importance graph and draw conclusions about the significant variables.

Test questions:

1. How does gradient boosting work?
2. How is gradient boosting different from random forest?
3. What does the `learning_rate` parameter mean and how does it affect the model?
4. How does the `n_estimators` parameter affect overfitting and underfitting?
5. What is the role of trees in gradient boosting?
6. What loss functions can be used in gradient boosting?
7. What is the role of regularization in boosting?
8. What does `feature_importances_` show in the boosting model?
9. How to determine the optimal model parameters?
10. In what problems might gradient boosting be preferable to other methods?

CONCLUSION

The laboratory work presented in these guidelines covers key areas of the discipline *"Artificial Intelligence and Machine Learning"* and is aimed at developing students' sustainable skills in constructing, configuring, and analyzing machine learning models using modern tools and libraries of the Python language .

The coursework is structured in increasing complexity: from mastering the basic tools of the software environment (Jupyter Notebook, NumPy, pandas, matplotlib) to the practical application of supervised and unsupervised learning algorithms, including methods for classification, regression, ensemble learning, clustering, and dimensionality reduction. The final sessions focus on improving model performance, including selecting quality metrics and selecting hyperparameters.

Each laboratory work contains an objective, theoretical justification, a list of required software and data, typical tasks to complete, as well as test questions for self-assessment and ongoing monitoring.

Systematic completion of laboratory assignments contributes to: - consolidation of theoretical knowledge; - development of applied competencies in data analysis and interpretation; - development of analytical thinking and the ability to work with real-world problems; - preparation for independent professional and research activities in the field of data analysis and artificial intelligence.

The practical portion of the course focuses on the use of open and accessible software tools, ensuring learning flexibility and the applicability of acquired knowledge across a wide range of fields.

EDUCATIONAL, METHODOLOGICAL AND INFORMATIONAL SUPPORT OF THE DISCIPLINE

Main literature:

Bolotsky, A. V. Operations research and optimization methods
Electronic resource / Bolotsky A. V., Kochetkova O. A. - St. Petersburg: Lan, 2020. - 116 p. - ISBN 978-5-8114-4568-4

Saraev, P. V. Machine Learning Methods Electronic resource: Methodological instructions and assignments for laboratory work on the course / P. V. Saraev. - Lipetsk: Lipetsk State Technical University, EBS ASV, 2017. - 48 p. - The book is in the premium version of the EBS IPR BOOKS. - ISBN 2227-8397

Sopov, E. A. Multicriteria neuroevolutionary systems in machine learning and human-machine interaction problems. Electronic resource / Sopov E. A., Ivanov I. A.: monograph. - Krasnoyarsk: SFU, 2019. - 160 p. - ISBN 978-5-7638-3969-2

Further reading:

Guidelines for practical classes in the discipline "Theory and Practice of Machine

Translation": Field of study 45.04.02 Linguistics. Master's program "Translation and Translation Studies". Graduate qualification - Master. Mode of study - full-time. Curriculum for 2015. Studied in the 3rd semester. - Stavropol: SKFU, 2015. - 9 p.

Nedelko, V. M. **Fundamentals of Statistical Methods of Machine Learning** Electronic resource: Textbook / V. M. Nedelko. - Novosibirsk: Novosibirsk State Technical University, 2010. - 72 p. - The book is in the premium version of the IPR BOOKS electronic library system. - ISBN 978-5-7782-1385-2

Rapakov, G. G.
Methods and algorithms of machine learning in making management decisions in the regional system of medical prevention (experience of the Vologda region) Electronic resource / Rapakov G. G., Kasimov R. A.: monograph. - Vologda: Vologda State University, 2014. - 143 p. - ISBN 978-5-87851-548-1

Educational and methodological literature:

Saraev, P. V. & Machine Learning Methods Electronic resource: Methodological instructions and assignments for laboratory work on the course / P. V. Saraev. - Lipetsk: Lipetsk State Technical University, EBS ASV, 2017. - 48 p. - The book is in the premium version of the EBS IPR BOOKS. - ISBN 2227-8397

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
FEDERAL STATE AUTONOMOUS EDUCATIONAL
INSTITUTION OF HIGHER EDUCATION
NORTH CAUCASIAN FEDERAL UNIVERSITY

GUIDELINES

on organizing and conducting independent work

Fundamentals of Computational Thinking and Artificial Intelligence

for students in the field of study

11.03.02 Infocommunication technologies and communication systems
Profile

"Intelligent Data Processing in Infocommunication Systems and Networks"

Stavropol 2026

Content

Introduction

Independent work as the most important form of the educational process

Goals and main objectives of the SRS

Types of independent work

Organization of the SRS

General recommendations for organizing independent work

Educational, methodological and informational support of the discipline

INTRODUCTION 1. The purpose and objectives of mastering the discipline (module)

The purpose of the discipline is to develop, in accordance with the educational program, the competencies of the OPK-4 of the student.

To achieve this goal, the following objectives are addressed in the course: familiarization with the basic concepts and types of machine learning problems; study of data preparation methods, including cleaning, normalization, and feature engineering; mastering basic supervised learning models, such as linear and logistic regression; mastering stochastic gradient descent as a method for optimizing models; forming an understanding of metrics for assessing the quality of classification and interpreting model results; studying methods for combating overfitting, including regularization and cross-validation; mastering classification algorithms based on the nearest neighbor method and the support vector machine; study of algorithms for constructing decision trees and ensemble models, such as random forest and gradient boosting; familiarity with unsupervised learning methods, including clustering algorithms; mastering methods of dimensionality reduction and data visualization, such as PCA and t-SNE; developing practical skills in tuning models using hyperparameter selection; developing the ability to compare and interpret the results of different models based on objective metrics.

2. The place of the discipline (module) in the structure of the educational program

The discipline belongs to the compulsory part B1.O.13. of the curriculum.

3. List of planned learning outcomes for the discipline (module), correlated with the planned learning outcomes for the educational program

4. Volume of the academic discipline (module) and forms of control *

INDEPENDENT WORK AS THE MOST IMPORTANT FORM OF THE EDUCATIONAL PROCESS

Independent student work is an integral part of mastering the *"Artificial Intelligence and Machine Learning"* course and plays a key role in developing solid and deep knowledge. Modern machine learning methods require not only a theoretical understanding of algorithms but also a confident mastery of tools, data analysis skills, interpretation of results, and critical thinking.

In the process of independent work, the student gets the opportunity to:

- to consolidate and deepen the material studied in lectures and practical classes;
- learn to independently find and study professional sources of information;
- apply the studied algorithms to real or near-real problems;
- develop practical skills in programming, working with Python libraries and data visualization;
- master different approaches to solving the same problem, choose optimal methods, and justify the choice made;
- prepare for the final test by demonstrating confidence in basic concepts and algorithms.

Furthermore, independent work develops discipline, initiative, the ability to plan one's own learning, and adapt to a constantly evolving subject area. These skills are especially important in the field of artificial intelligence, where the pace of technological change requires constant knowledge updating.

Independent work -is planned educational, educational and research, scientific research work of students, carried out outside of class (classroom) time on assignment and with the

methodological guidance of the teacher, but without his direct participation (with partial direct participation of the teacher, leaving the leading role to the students' work).

Independent student work at a university is an important form of academic and research activity. Independent student work plays a significant role in the rating system of learning. Studying at a university involves two parts, almost equal in scope and mutual influence: the learning process and self-study. Therefore, independent student work should be an effective and purposeful part of student work.

The concept for modernizing Russian education defines the main objectives of vocational education as "the preparation of a qualified worker of the appropriate level and profile, competitive in the labor market, competent, responsible, fluent in their profession and oriented in related fields of activity, capable of effective work in their specialty at the level of world standards, ready for continuous professional growth, social and professional mobility."

The solution to these problems is impossible without increasing the role of students' independent work on educational material, strengthening the responsibility of teachers for developing independent work skills, stimulating students' professional growth, and fostering creativity and initiative.

Society places a wide range of demands on modern professionals, among which a significant one is the ability of graduates to independently acquire knowledge from various sources, systematize the information obtained, and evaluate specific financial situations. This ability is developed throughout the entire educational period through students' participation in laboratory classes, completing assignments and tests, and writing coursework and final theses. Students' independent work plays a crucial role throughout the entire educational process.

Students' independent work options are varied. The following are included in this course, " Artificial Intelligence and Machine Learning ":

Use of the material of the educational and methodological complex of the discipline

The first step requires familiarization with the required minimum content of the core curriculum for the discipline being studied. Next, students must review the course syllabus, which covers the content of lecture course topics, the relationship between lecture topics and laboratory classes, and the topics and types of independent work. Specific reporting forms are provided for each type of independent work, as outlined in the curriculum for the course " Artificial Intelligence and Machine Learning ."

Working with literature

To successfully master the course, you must independently and thoroughly study the presented topics using the recommended sources of information provided in the curriculum.

Independent work introduces students to scientific creativity, searching for and solving current modern problems.

GOALS AND MAIN OBJECTIVES OF THE SRS

Objectives of independent work:

- Deepening of theoretical knowledge obtained in lectures;
- Consolidation of practical skills acquired at seminars;
- Development of skills in solving applied machine learning problems in practice;
- Formation of a critical approach to the selection of models, parameters and metrics.

Forms of independent work:

- Study of recommended sources and lecture materials;
- Analysis and solution of the proposed problems;
- Completion of individual practical assignments in Jupyter Notebook;
- Preparation of mini-reports and interpretation of the results obtained;
- Independent review of the material and preparation for testing.

The primary goal of organizing and implementing independent work should align with the goal of student education—the preparation of a bachelor's degree holder with a higher education degree. When organizing independent work, developing the ability to work independently to acquire knowledge, skills, and the ability to organize educational and research activities is an important and necessary condition.

The goal of students' independent work is to acquire fundamental knowledge, professional skills, and relevant professional abilities, as well as experience in creative and research activities. Independent work promotes the development of students' independence, responsibility, self-discipline, and a creative approach to solving academic and professional problems.

The objectives of the SRS are:

- systematization and consolidation of the acquired theoretical knowledge and practical skills of students;
- deepening and expanding theoretical knowledge;
- development of skills in using regulatory, legal, reference documentation and specialized literature;
- development of students' cognitive abilities and activity: creative initiative, independence, responsibility and organization;
- development of independent thinking, abilities for self-development, self-improvement and self-realization;
- development of research skills;
- use of material collected and obtained during independent study at seminars, in laboratory classes, when writing coursework and final qualifying papers, for effective preparation for final tests and exams.

TYPES OF INDEPENDENT WORK

In the educational process at a higher professional educational institution, two types of independent work are distinguished : in-class, under the guidance of a teacher, and out-of-class. The close interrelationship between these types of work allows for differentiation and the effectiveness of their results and depends on the organization, content, and logic of the educational process (interdisciplinary connections, advanced knowledge, etc.) :

Independent classroom work on the subject is carried out during classes under the direct supervision of the teacher and according to his/her instructions.

Extracurricular independent work is performed by a student on the teacher's instructions, but without his or her direct participation.

The main types of independent work of students without the participation of a teacher in this discipline are:

- formation and assimilation of the content of lecture notes based on the educational literature recommended by the lecturer, including information educational resources (electronic textbooks, electronic libraries, etc.);
- preparation for laboratory work, its design.

The main types of independent work of students with the participation of teachers are:

- ongoing consultations;
- acceptance and defense of laboratory work (during laboratory work).

ORGANIZATION OF SRS

The methodology for organizing students' independent work depends on the structure, nature, and characteristics of the course " Artificial Intelligence and Machine Learning ," the number of hours for its study, the type of assignments for students' independent work, the individual qualities of students, and the conditions of educational activities.

The process of organizing students' independent work includes the following stages:

- preparatory (defining goals, drawing up a program, preparing methodological support, preparing equipment);
- basic (implementation of the program, use of techniques for searching for information, assimilation, processing, application, transfer of knowledge, recording of results, self-organization of the work process);
- final (assessment of significance and analysis of results, their systematization, assessment of the effectiveness of the program and work methods, conclusions on areas of labor optimization).

Recommended topics and tasks for independent study:

1. Data preparation and visualization

- Study the types of features, scaling and coding methods.
- **Task** : select a dataset from ``sklearn.datasets`` (e.g. ``wine`` or ``breast_cancer``), visualize feature distribution (histograms, pair plots), apply ``StandardScaler`` and ``OneHotEncoder``.

2. Linear and logistic regression analysis

- Study the mathematical foundations and methods of quality assessment.
- **Task** : Train a linear regression model on Boston housing data, calculate MAE and R^2 ; build a logistic regression on ``breast_cancer`` and plot the ROC curve.

3. Retraining and regularization

- Study the concepts of L1 and L2 regularization, the meaning of the parameter ``C``.
- **Task** : train logistic regression with different regularization values on the ``digits`` dataset and compare the results on the training and test samples.

4. Classification methods: KNN and SVM

- Compare the performance of models with different parameters.
- **Task**: implement KNN and SVM on the same dataset (``iris``), plot accuracy graphs depending on the parameters (``k``, ``C``, ``kernel``).

5. Model ensembles

- Learn the basics of random forest and gradient boosting.
- **Task**: train ``RandomForestClassifier`` and ``GradientBoostingClassifier`` models, compare their accuracy and feature importance (``feature_importances_``).

6. Clustering methods

- To study the difference between K-means and hierarchical clustering.
- **Task** : Apply KMeans and Agglomerative Clustering to ``iris``, compare the results with the true labels. Build a dendrogram.

7. Dimensionality reduction and visualization

- Learn PCA and t-SNE algorithms.
- **Task**: Reduce the ``digits`` dimension using PCA to 2 components and visualize the result. Repeat with t-SNE.

8. Hyperparameter selection

- Master working with `'GridSearchCV'`.

- **Task:** Find optimal parameters for SVM using GridSearchCV on the wine dataset.

Record the evaluation metric.

Typical tasks for independent work:

General approach to assessment:

- Each task is assessed on a **10-point scale** ;
- The overall grade for the independent work block can be calculated as an average or a sum (at the discretion of the teacher);
- It is possible to use **partial points (0.5, 1, etc.)** for partially completed assignment points;
- **No** points will be awarded for copying without understanding, non-working code, or missing interpretation .

No .	Exercise	Evaluation criteria (max. 10 points)
1	Explore the differences between supervised and unsupervised learning. Provide two examples of each type of problem from a professional field.	2 points – correct theoretical definition; 4 points – relevant examples; 4 points – explanation and analysis of differences.
2	<code>iris</code> or <code>wine</code> dataset . Perform preliminary analysis: dimensionality, feature types, and distributions.	3 points – loading and describing data; 3 points – visualization of distributions; 4 points – interpretation of results.
3	Scale features using <code>StandardScaler</code> and <code>MinMaxScaler</code> . Compare the effects of scaling.	3 points – correct implementation; 3 points – visualization of changes; 4 points – analytical comparison.
4	Encode categorical features using <code>OneHotEncoder</code> and <code>OrdinalEncoder</code> , explain the difference.	3 points - correct implementation; 3 points - examples; 4 points - explanation of the differences and when to apply what.
5	Build a linear regression model for Boston housing. Calculate MAE, MSE, and R^2 .	2 points – data preparation; 4 points – model and metrics implementation; 4 points – results interpretation.
6	Construct a logistic regression on <code>breast_cancer</code> , calculate metrics, and plot an ROC curve.	3 points – model training; 4 points – correct metric calculations; 3 points – ROC curve and conclusions.
7	Implement gradient descent manually for linear regression, plot the error function.	4 points - implementation without sklearn; 3 points - visualization; 3 points - graph interpretation.
8	Compare the performance of KNN and SVM on the same dataset, visualize and compare metrics.	3 points – correct implementation; 3 points – visualization; 4 points – analytical comparison.
9	Build a decision tree on <code>iris</code> , visualize it, and analyze signs of overfitting.	3 points – construction and visualization; 3 points – tree structure analysis; 4 points – signs of overfitting.
10	Train <code>RandomForestClassifier</code> , compare	3 points – correct implementation of

	with standard tree.	both models; 3 points – accuracy comparison; 4 points – feature importance analysis.
11	Train <code>GradientBoostingClassifier</code> on <code>wine</code> , select parameters.	3 points – model training; 3 points – parameter enumeration; 4 points – justification of choice and comparison.
12	Conduct cross-validation for logistic regression, compare with train/test.	3 points – CV implementation; 3 points – results comparison; 4 points – conclusions about the model stability.
13	Find SVM hyperparameters using <code>GridSearchCV</code> , present the table.	3 points – search implementation; 3 points – table design; 4 points – justification of choice.
14	K-means clustering on <code>iris</code> , comparison with real labels, silhouette score.	3 points - implementation; 3 points - comparison of clusters and labels; 4 points - quality assessment.
15	Hierarchical clustering, dendrogram, determination of the number of clusters.	3 points – model construction; 3 points – dendrogram visualization; 4 points – correct interpretation.
16	Apply PCA to <code>digits</code> , visualize in 2D.	3 points - PCA implementation; 3 points - visualization; 4 points - interpretation of dimensionality changes.
17	Compare PCA and t-SNE visualizations. Explain the identified structures.	3 points – implementation and graphs; 3 points – comparison of results; 4 points – explanation of differences.
18	To study the influence of regularization on logistic regression, to plot the dependence of accuracy on C.	3 points - experiment with C; 3 points - graph; 4 points - interpretation and conclusions.
19	Create a table of classification algorithms: pros, cons, and applications.	3 points – completeness of the table; 3 points – accuracy of information; 4 points – clarity of wording and practicality.
20	Prepare a Jupyter Notebook with a full ML cycle on an arbitrary dataset.	2 points – data selection and loading; 4 points – completion of all stages (pre-processing, model, metrics); 4 points – logical structure and design.
21	Select any dataset from <code>sklearn.datasets</code> and perform exploratory data analysis (EDA).	4 - Correct and varied EDA; 3 - Conclusions; 3 - Neat design and visualization.
22	Implement code that checks for missing values and outliers.	3 - verification code; 3 - emission visualization; 4 - interpretation.
23	Visualize feature correlations using a heatmap (<code>seaborn.heatmap</code>).	3 - correct code; 3 - clear visualization; 4 - conclusions about multicollinearity.
24	Compare scaling methods: <code>StandardScaler</code> , <code>MinMaxScaler</code> , <code>RobustScaler</code> .	4 - application of all methods; 3 - visualization; 3 - conclusions on differences.
25	Apply <code>PolynomialFeatures</code> and train polynomial regression.	3 - generation of new features; 4 - training and visualization; 3 -

		conclusions.
26	Implement your own cross-validation split function (without sklearn).	5 - implementation; 2 - correctness; 3 - explanation of logic.
27	To investigate the impact of imbalanced classes on the accuracy of logistic regression.	4 - create/load unbalanced set; 3 - compare; 3 - conclude.
28	Use SMOTE to balance classes.	3 - correct application; 4 - comparison of results before and after; 3 - conclusions.
29	Apply the classifier to randomly generated data (make_classification).	3 - set generation; 4 - training and metrics; 3 - interpretation of problem complexity.
30	To investigate the influence of the number of features on the model performance.	3 - creating dataset versions; 4 - training; 3 - graph/dependency analysis.
31	Implement greedy feature selection (SelectKBest) and compare with the model without selection.	3 - implementation; 4 - accuracy comparison; 3 - conclusions.
32	Apply sklearn's Pipeline for the full cycle: scaling + model.	4 - correct pipeline structure; 3 - conclusions; 3 - code refactoring.
33	To investigate the impact of outliers on the accuracy of KNN.	3 - creating a set of outliers; 4 - metrics; 3 - analyzing the model behavior.
34	Perform classification using NaiveBayes, compare with logistic regression.	3 - model training; 4 - metrics and table; 3 - conclusions.
35	Implement a voting method (VotingClassifier) based on 3 models.	3 - ensemble assembly; 4 - accuracy comparison; 3 - explanation of the effect.
36	Implement a random forest model using only the most important features.	3 - importance determination; 4 - feature selection; 3 - quality comparison.
37	Perform regression using Ridge and Lasso.	3 - implementation; 4 - weight graphs; 3 - comparison and interpretation.
38	Plot a graph of the dependence of classification accuracy on the proportion of the training sample.	4 - create partitions; 3 - plot a graph; 3 - conclusions.
39	Compare the accuracy of models with normalized and unnormalized features.	3 - comparison; 4 - results table; 3 - explanation of differences.
40	Write your own implementation of logistic regression (without sklearn).	4 — implementation; 3 — testing; 3 — comparison with sklearn.
41	Visualize decision boundaries for SVM and KNN on two-dimensional data.	4 - graph; 3 - abstract; 3 - explanation of features.
42	Implement comparison of models by training and prediction time.	3 - correct measurements; 4 - graph or table; 3 - interpretation.
43	To study the influence of noise in data on classification.	3 - adding noise; 4 - comparing models; 3 - conclusions.
44	Compare models in terms of resistance to overfitting on small and large data sets.	4 - implementation; 3 - visualization of results; 3 - analysis.
45	Implement training and validation curves (learning_curve) for SVM.	3 - code; 4 - graphics; 3 - interpretation.

46	Construct a confusion matrix and analyze model errors.	3 - implementation; 4 - visualization; 3 - interpretation.
47	Implement an analysis of the influence of categorical features on the quality of the model before and after coding.	3 - examples; 4 - model results; 3 - explanation of differences.
48	To investigate the impact of imbalanced classes on decision trees and random forest.	3 - implementation; 4 - comparison; 3 - interpretation.
49	Train a classification model on arbitrary text data (TF-IDF + logistic regression).	3 - preprocessing; 4 - training and metrics; 3 - inference.
50	To analyze the relationship between the number of features and model overfitting.	3 - change of dimension; 4 - training and graph; 3 - interpretation of results.

Students' activities to form and develop skills for independent study

In the process of independent work, the student acquires the skills of self-organization, self-control, self-management, self-reflection and becomes an active, independent subject of educational activity.

Doing independent work under the supervision of a teacher

The student must:

- master the minimum content assigned to students for independent work and proposed by the teacher in accordance with the educational standards of higher education;
- plan independent work in accordance with the independent work schedule proposed by the teacher;
- The student must carry out independent work in the organizational forms provided for by the curriculum and the teacher's work program;
- carry out independent work and report on its results in accordance with the schedule for presenting results, types and deadlines for reporting on students' independent work.

a student can:

- in addition to that proposed by the teacher (subject to justification and agreement with him) and the minimum required content determined by the program for a given discipline:
 - independently determine the level (depth) of elaboration of the content of the material;
 - suggest topics and questions for independent study;
 - within the general schedule for completing independent work, propose a reasonable individual schedule for completing and reporting on the results of independent work;
 - propose your own options for organizational forms of independent work;
 - use for independent work methodological aids, teaching aids, and developments in addition to the list proposed by the teacher;
 - use not only control, but also self-control of the results of independent work in accordance with the methods of self-control proposed by the teacher or chosen independently.

Students' independent work should have a significant impact on the development of their future bachelor's degree students. It is planned by the student themselves. Each student determines their own work schedule and the amount of effort expended on mastering the academic content in each subject. They complete their extracurricular work according to their individual plan, depending on their preparation, time, and other circumstances.

Questions for checking the independent work system

1. What is machine learning? How is it different from rule-based programming?
2. What types of problems can be solved using supervised learning?
3. What is the difference between supervised and unsupervised learning?
4. Give examples of classification and regression problems.

5. What are training and test sets? Why is it important to distinguish between them?
6. What is overfitting and how to detect it?
7. What methods can be used to combat overfitting?
8. What is the difference between L1 and L2 regularization?
9. What are features and target variable?
10. Why are methods of normalization and standardization of features needed?
11. What is one-hot encoding and when is it used?
12. What is linear regression and what problems does it solve?
13. How are coefficients interpreted in a linear regression model?
14. What is logistic regression and how is it different from linear regression?
15. What problems is logistic regression suitable for?
16. How does the sigmoid function work and how is it used in logistic regression?
17. What is classification accuracy?
18. What is the difference between precision and recall?
19. When is it important to use the F1 measure?
20. What does the ROC curve and AUC show?
21. How does KNN work?
22. How to choose the optimal number of neighbors in KNN?
23. What is Support Vector Machine (SVM)?
24. What is the difference between a linear and a nonlinear kernel in SVM?
25. How does stochastic gradient descent work?
26. Why use mini-batches when training a model?
27. What is a decision tree and how is it built?
28. How does a decision tree make decisions based on new data?
29. Why are trees prone to overfitting and how can this be prevented?
30. What are the advantages of ensemble models over single models?
31. What is a random forest and how does it work?
32. What is gradient boosting?
33. What is the difference between K-means and hierarchical clustering?
34. What is PCA and why is it used?
35. What is the difference between PCA and t-SNE?
36. What methods for assessing the quality of clustering do you know?
37. What are model hyperparameters?
38. How does GridSearchCV work and what is it used for?
39. What Python libraries are used for machine learning?
40. How to interpret the results of the model and decide on their applicability?

GENERAL RECOMMENDATIONS FOR ORGANIZING INDEPENDENT WORK

The primary form of student independent work is studying lecture notes, supplementing them, referring to recommended literature, and actively participating in laboratory classes. However, to ensure successful learning and its intensification, the following subjective factors must be considered:

1. Knowledge of the school curriculum and a solid system of gaps necessary for mastering core university courses. This is especially important for mathematical subjects. It is important to distinguish between gaps in knowledge that hinder the assimilation of new material and limited abilities. By investing effort in overcoming these gaps, the student will ensure adequate academic performance and gain confidence in their abilities.

2. Availability of skills and abilities for mental work:

- a) the ability to take notes during lectures and when working with a book;
- b) mastery of logical operations: comparison, analysis, synthesis, generalization, definition of concepts, rules of systematization and classification.

3. The specificity of cognitive mental processes: attention, memory, speech, observation, intellect, and thinking. Poor development of any of these can become a serious obstacle to learning.

4. Good performance, which is ensured by normal physical condition. After all, serious study is a complex, multifaceted, and varied undertaking. Learning outcomes are assessed not by the quantity of information imparted, but by the quality of its assimilation, the ability to use it, and the development of one's capacity for further independent learning.

5. Match your chosen activity or profession to your individual abilities. It's essential to develop the ability to self-regulate your emotional state and eliminate circumstances that disrupt your professional spirit and interfere with your intended work.

6. Mastering an optimal work style that ensures success in your work. Alternating work and breaks, rest periods, individually determined sleep duration, preference for evening or morning classes, stress tolerance during exams, and specific preparation techniques.

7. The level of demands on oneself, determined by the established self-esteem.

An adequate assessment of knowledge, strengths, and weaknesses is an important component of human self-organization; without it, successful work on managing one's behavior and activities is impossible.

One of the main features of higher education is that constant external control is replaced by self-control; the active role in learning belongs not so much to the teacher as to the student.

Knowing the basic methods of scientific organization of mental work, one can achieve the best results with the least expenditure of time, resources and labor effort.

The effectiveness of assimilation of incoming information depends on a person's performance at any given moment of his activity.

Work capacity is a person's ability to work at a high level of intensity for a certain period of time. A distinction is made between internal and external factors of work capacity.

Internal factors of performance include intellectual abilities, willpower, and health status.

To external:

- organization of the workplace, work and rest schedule;
- level of work organization - ability to obtain information and use it;
- the amount of mental workload.

The outstanding Russian physiologist N. E. Vvedensky identified the following conditions for the productivity of mental activity:

- you need to enter into any work gradually;
- the regularity and rhythm of work. Different people have more or less different work tempos;
- habitual sequence and systematicity of activity;
- correct alternation of work and rest.

Rest doesn't necessarily require complete inactivity; it can be achieved by simply changing tasks. Work capacity fluctuates throughout the day. The most productive *time is morning* (8:00 a.m. to 2:00 p.m.), with peak productivity occurring between 10:00 a.m. and 1:00 p.m., followed by *afternoon* (4:00 p.m. to 7:00 p.m.) and *evening* (8:00 p.m. to midnight). It's best to study very difficult material at the beginning of each work period (preferably morning) after a good rest. Breaks of 10-15 minutes are needed after 1-1.5 hours of work, and after 3-4 hours of work, rest should be longer – about an hour.

An integral part of the scientific organization of mental work is the mastery of the technique of mental work.

The time available for students to complete their curriculum consists of two components: one is scheduled classroom work at the university, and the other is independent work outside of class. Assignments and materials for independent work are provided during scheduled classes.

