Syllabus for the Course "Neural Networks, Machine Learning and Deep Learning"

1. Abstract of the Discipline

The course is aimed at developing an engineering understanding of machine learning methods and neural network architectures as components of modern digital systems.

This course examines machine learning not only as a mathematical framework, but also as an engineering technology for creating data-driven functionality for digital products. The training focuses on:

- setting machine learning problems;
- data engineering and dataset preparation;
- construction and comparison of models;
- neural network architectures;
- deep learning methods;
- integration of models into digital systems.

Particular attention is paid to the role of Foundation Models and modern deep learning architectures as the technological basis of intelligent systems.

Large Language Models (LLMs) are used in the course as:

- an engineering analysis tool;
- an intelligent assistant for model development;
- a means of reflection and hypothesis testing.

2. General Characteristics of the Discipline

The course belongs to the professional module of the Master's degree program «AI-Augmented Digital Systems Engineering» in the field of study 09.04.02 «Information Systems and Technologies».

The total volume of the course is 6 ECTS credits (216 academic hours). The course is implemented in the second and third semesters.

Class workload:

Second semester:

theoretical classes — 22 hours;
practical and laboratory classes — 23 hours.

Third semester:

theoretical classes — 24 hours;

practical and laboratory classes — 24 hours.

Form of interim assessment:

second semester — exam;
third semester — exam.

The course is implemented in the AI-Augmented Engineering Learning format, which involves the integration of large language models into the educational process.

3. The Place of the Discipline in the Structure of the Educational Program

The discipline is a central element of the educational track «Artificial Intelligence (AI Technologies)». The track forms a consistent line of training for specialists in the engineering of intelligent digital systems.

Within the track, the discipline occupies a central methodological position. It connects:

- the course «Fundamentals of Artificial Intelligence and Large Language Models: Prompt Engineering and Context Engineering»;
- with subsequent specialized disciplines:
- Testing, Verification and Validation of AI Systems;
- Natural Language Processing and Large Linguistic Models (NLP & LLM);
- Computer Vision;
- Edge AI Technologies.

While the first course of the track develops an understanding of the interaction between humans and AI models, the current course develops an engineering understanding of the architecture and training of the models themselves.

The course serves as a transition: from using models → to designing and training them → to integrating models into digital systems.

4. Objectives of Mastering the Discipline

The aim of the course is to develop students’ engineering understanding of machine learning and deep learning methods as tools for designing intelligent digital systems.

Particular attention is paid to the following educational outcomes:

- mastering the basic principles of machine learning;
- development of skills for setting ML problems;
- mastering methods of data preparation and analysis;
- study of the main classes of machine learning models;
- mastering neural network architectures;
- study of deep learning methods;
- development of skills for comparing models and assessing quality;

- mastering the principles of integrating models into digital systems;
- formation of a culture of experimental analysis of models;
- developing skills for critical use of LLMs in engineering activities.

5. Objectives of the Discipline

The main objectives of the discipline are:

- studying the types of machine learning problems;
- mastering data engineering methods;
- studying quality metrics of models;
- mastering methods of linear models and ensembles;
- study of unsupervised learning methods;
- mastering dimensionality reduction methods;
- study of the basic principles of neural networks;
- study of the backpropagation algorithm;
- mastering deep learning architectures;
- studying specialized architectures for different types of data;
- learning the principles of transfer learning;
- study of fundamental models;
- mastering the principles of ML pipeline;
- development of experimental management skills;
- mastering the principles of implementing models into digital systems.

6. Planned Learning Outcomes

Know:
- main types of machine learning tasks;
- methods of data preparation and analysis;
- model quality metrics;
- basic machine learning models;
- neural network architectures;
- deep learning methods;
- CNN, RNN and Transformer architectures;
- principles of transfer learning;
- the role of fundamental models;
- pipeline architecture;
- methods of experimental management.

Be able to:
- formalize the applied problem as an ML problem;
- analyze the data structure;
- select a model for the applied task;
- conduct model training;
- compare models by quality metrics;
- diagnose model errors;

- design an ML pipeline;
- integrate the model into the architecture of the digital system;
- critically analyze LLM recommendations.

Be proficient in:

- skills of experimental work with models;
- practices of behavior analysis of models;
- methods of comparing models;
- skills in architectural analysis of ML systems;
- methodology for maintaining an experiment log;
- skills in using LLMs as an engineering assistant.

7. Methodological Concept of the Discipline

Training is structured as an engineering trajectory for the development of models:

Problem Statement → Data → Model → Training → Evaluation → Architecture → Implementation

The course sequentially transfers the student: from understanding ML tasks → to building models → to deep learning architectures → to AI systems engineering.

Particular attention is paid to the following principles:

7.1. ML as an Engineering Discipline

Machine learning is viewed as an engineering process that includes:

- formalization of the problem;
- data preparation;
- model selection;
- training;
- quality assessment;
- integration of the model into the system.

7.2. Experimental ML Culture

The central element of learning is experimentation. Each hypothesis must be tested through:

- model training;
- measuring metrics;
- error analysis.

7.3. Architectural Thinking

The course develops an understanding that a model is part of the architecture of a digital system. Discussed:

- the role of the ML component in the system;
- computing resource limitations;
- performance requirements;
- model implementation architecture.

8. The Role and Contribution of LLM in the Educational Process

LLMs are integrated into the course as a tool for engineering activities.

8.1. LLM as a Tool for Formulating Hypotheses

LLMs can be used for:

- hypothesis generation;
- formulation of model variants;
- analysis of alternative approaches.

8.2. LLM as an Analytical Assistant

LLMs are used when:

- analyzing model results;
- interpreting metrics;
- discussing architectural solutions.

8.3. LLM as an Engineering Opponent

LLMs are used for:

- identifying weaknesses of the model;
- proposing alternative solutions;
- testing the logic of experiments.

8.4. Principle of Mandatory Verification

Any LLM claim must be verified through:

- experiment;
- measurement;
- analysis of results.

The course reinforces the professional principle: trust the experiment, not the formulation.

9. Educational Technologies

The discipline uses:

- lectures in the format of an engineering master class;
- digital laboratory work;
- experimental studies of models;
- analysis of AI system architectures;
- project activities;
- using LLMs as an intelligent assistant.

10. Differentiated Assessment Model

The assessment is based on the principle of humane ambition.

Basic level:

- correct implementation of the model;
- correct choice of metrics;
- basic understanding of the algorithm.

Advanced level:

- comparative analysis of models;
- explanation of model errors;
- argumentation of architectural choice.

Research level:

- additional experiments;
- model behavior analysis;
- critical assessment of LLM recommendations.

11. Final Certification

An exam is held each semester. The exam consists of two parts:

Theoretical part: Understanding is checked:

- machine learning methods;
- neural network architectures;
- principles of model training;
- ML systems engineering.

Practical defense: The student presents the results of work on an end-to-end project:

- statement of the problem;
- data description;
- constructed models;
- comparative analysis;
- pipeline architecture.

Curriculum schedule for the course

Key:

- **Lecture (Lect.)** — theoretical material delivery.
- **Practical / Lab.** — hands-on activities, simulations and discussions.
- **Self-directed work (SDS)** — independent study on project tasks and analysis.
- **LLM use** — applying Large Language Models for analysis, generation and decision support (integrated into independent study tasks).

Semester 2

Week	**Content**	**Lect. (hours)**	**Practice (hours)**	**SDS (hours)**
1	**Theory:** — Role of ML in modern digital systems; — Distinction between algorithmic programming, statistical modelling, and data-driven approaches; — ML component in system architecture. **Practice:** — Analysis of real digital systems with ML components (recommendation, prediction, classification); — Decomposition of a sample system into software modules and model-driven modules in a digital lab. **SDS:** — Select a real digital product/service; — Analyse: problem solved by ML, input data, output result, why traditional programming is insufficient. **Independent work:** — Choose project subject area; — Formulate preliminary applied problem for ML solution; — Describe model application context and expected value; — Use LLM as assistant for problem statements and architectural context. **Educational impact:** Students develop an understanding that ML is an engineering mechanism for digital system functionality, not just a mathematical technique. They gain basic understanding of where the ML component appears in system architecture and what problems it solves.	2	2	6
2	**Theory:** — Classification of ML problems (supervised, unsupervised, reinforcement learning); — Problem formats: classification, regression, clustering, ranking, anomaly detection; — Engineering formulation specifics.	2	2	6

Week	Content	Lect. (hours)	Practice (hours)	SDS (hours)
	Practice: — Translate real-world cases into ML problem terms; — Analyse datasets: identify features, target variable, expected result. **SDS:** — For given scenarios: define ML problem class, justify model type, propose quality metrics; — Verbally formalise problem without premature algorithm choice. **Independent work:** — Determine project problem type (classification, regression, etc.); — Specify target variable, observation objects, usefulness criterion; — Use LLM to generate alternative problem formulations and verify logical correctness. **Educational benefit:** Students acquire ability to translate applied engineering problems into formalised ML problem statements. They learn to distinguish when to use ML tools and when data-driven learning is excessive.			
3	**Theory:** — Data as primary ML resource: structure, objects, features, labelling; — Issues: incomplete, noisy, unbalanced, biased data; — Impact on training accuracy and result interpretation. **Practice:** — Exploratory data analysis: feature distributions, gaps, outliers, artifacts; — Digital lab: data cleaning, formatting, normalising, feature coding. **SDS:** — Prepare analytical report on selected dataset: origin, structure, feature types, risks, potential training issues; — Identify data aspects leading to incorrect model conclusions. **Independent work:** — Form primary project dataset or identify external dataset; — Describe data structure, acquisition mechanism, dataset limitations; — Use LLM for analysis templates, validation checklists, bias explanation, limited synthetic data generation. **Educational impact:** Students understand that model quality is determined primarily by data quality and structure. This fosters an engineering approach to data as an object of design, preparation, and control.	2	2	6
4	**Theory:** — Loss function, training error, test error, model generalisation ability; — Distinction: good results on known sample vs. correct work on new data. **Practice:** — Examine scenarios where high training accuracy doesn’t lead to high practical quality; — Build simple models, compare results on training/test sets. **SDS:** — Interpret model metrics: identify good quality indicators and generalisation problems; — Explain role of train/test split and its limitations. **Independent work:** — Define project model evaluation framework: data partitioning, validation logic, metrics list; — Establish methodology for comparing model variants.	2	2	6

Week	Content	Lect. (hours)	Practice (hours)	SDS (hours)
	Educational impact: Students begin to understand that model quality is a measurable ability to produce reliable results on new data. This lays foundation for meaningful model evaluation and discussions about retraining.			
5	**Theory:** — Overfitting and underfitting: causes and manifestations; — Trade-off: model complexity vs. generalisation ability; — Bias-variance trade-off. **Practice:** — Compare models of varying complexity on same dataset, observe overfitting signs; — Experiments with model parameters, data volume, result quality. **SDS:** — Analyse learning curves/comparative results, identify overfitting/underfitting areas; — Write rationale for engineering actions to improve situation. **Independent work:** — Check project base model for overfitting signs, identify corrections (feature changes, regularisation, model simplification, data increase); — Document successful and unsuccessful solutions. **Educational impact:** Students understand why a model can show impressive results on known data but be practically unusable. A culture of diagnosing models based on behaviour (not just numerical indicators) is fostered.	1	2	6
6	**Theory:** — Quality metrics for classification/regression: accuracy, precision, recall, F-score, ROC-AUC, MAE, MSE; — Metric choice depends on application problem, error cost, data structure. **Practice:** — Examine situations where high accuracy is misleading (e.g., unbalanced data); — Calculate metrics for several models, compare quality interpretation under different criteria. **SDS:** — Given model results: choose best model for application scenario, justify via metrics and error cost; — Explain why same model can be evaluated differently under different criteria. **Independent work:** — Finalise project metrics, justify reflection of model usefulness in subject area; — Create basic model evaluation table for course use. **Educational impact:** Students acquire ability to correctly select and interpret model quality metrics. They see that the «best model» depends on applied context, not just formal metric maximum.	2	2	6
7	**Theory:** — Linear and logistic regression as basic, interpretable, methodologically important ML models; — Prerequisites for application, linear dependence intuition, baseline model role. **Practice:** — Construct linear models on real/training data, interpret coefficients, analyse limitations; — Emphasise good engineering	1	2	6

Week	Content	Lect. (hours)	Practice (hours)	SDS (hours)
	Practice: start with simple, explainable model. **SDS:** — Implement and compare linear/logistic regression for corresponding problem types; — Written analysis: where interpretability provides advantage, where linear model is insufficient; — Explain baseline model value in ML system design. **Independent work:** — Create first baseline model for project (starting point for quality); — Record results, prepare basis for comparison with complex models. **Educational impact:** Students understand simple models have high methodological and engineering value. They develop skill of starting design with explainable solution, not most complex architecture.			
8	**Theory:** — Decision trees: nonlinear dependencies; — Ensemble methods: Random Forest, gradient boosting; — Composing weak/unstable models into robust prediction system. **Practice:** — Discuss decision tree interpretability advantages, single tree robustness limitations; — Build and compare tree, random forest, boosting: document quality, robustness, tuning requirements. **SDS:** — Brief comparative analysis of ensemble methods on selected dataset: quality gain vs. interpretability/computational complexity loss; — Formulate engineering choices (not just run models). **Independent work:** — Add at least one nonlinear model (decision tree/ensemble) to project, compare with baseline; — Update trade-off map: quality, interpretability, complexity. **Educational impact:** Students understand ensemble methods' importance in applied ML. They develop skill of comparing models by quality and engineering limitations (interpretability, operational cost).	1	2	6
9	**Theory:** This unit explores unsupervised learning methods aimed at identifying structure in data, primarily clustering. The intuition behind cluster analysis, the k-means algorithm, and the basic principles of hierarchical clustering are discussed, as well as the limitations of such methods. **Practice:** During practical classes, students apply clustering methods to unlabeled data and discuss the practical implications of the resulting clusters. A digital lab unit visualizes clusters and analyzes the dependence of the results on the choice of features and the number of clusters. **SDS:** The student is asked to conduct a cluster analysis of a small dataset and provide a written interpretation of the meaning of the identified groups. The student is required to separately indicate which conclusions are relatively reliable and which are tentative.	1	1	6

Week	Content	Lect. (hours)	Practice (hours)	SDS (hours)
	Independent Student Work as Part of an End-to-End Project: If the project task allows for unsupervised learning, the student explores the data structure using clustering; if not, this unit is used as a means of exploratory analysis of the feature space. The result is recorded as a brief engineering report on the data structure. **LLM in the educational process** helps formulate hypotheses about the possible interpretation of clusters, generate questions about clustering results, and critically discuss whether the identified groups are a statistical artifact. The effect is enhanced analytical reflection and the development of the skill of cautious interpretation. **Educational impact of the week:** Students gain an understanding of how to identify hidden structure in unlabeled data and the limitations of such analysis. They develop the ability to distinguish between statistical structure and the applied meaning of data.			
10	**Theory: Methods for reducing the dimensionality of feature spaces** are studied , primarily the principal component analysis (PCA). The topics discussed include visualization, simplifying data structure, and reducing feature redundancy, as well as limiting the interpretation of the resulting components. **Practice:** During seminars, students analyze cases where a large number of features complicates model analysis and training. In the lab, students apply PCA to real data, visualize objects in reduced dimensionality, and compare model behavior before and after feature transformation. **SDS:** Students analyze dimensionality reduction on a single dataset and describe what is preserved and what is lost during this transformation. They are also required to evaluate whether dimensionality reduction improves the quality of the model or, conversely, makes it less explicable. **Independent Student Work within an End- to-End Project:** If necessary, the student applies dimensionality reduction to project data and evaluates the impact of this step on interpretability, learning speed, and the quality of the result. If the application is impractical, the student must provide a justified reason for abandoning this tool. **LLM in the educational process:** LLM is used to explain the concept of dimensionality reduction intuitively, assist in interpreting graphs, and formulating analytical conclusions. The result is a faster understanding of the geometric meaning of the method and the development of the ability to explain results not only mathematically but also from an engineering perspective .	1	1	6

Week	Content	Lect. (hours)	Practice (hours)	SDS (hours)
	Educational impact: Students understand how to work with multidimensional data and how to simplify the feature space without completely losing its semantic structure. They develop the skill of connecting mathematical data transformations with real-world engineering goals.			
11	**Theory:The artificial neuron** is studied as the basic computational element of neural network models, as well as the perceptron as a historically important and conceptually simple neural architecture. The ideas of linear separation, activation, and limitations of single-layer models are discussed. **Practice:** During practical classes, students implement a simple artificial neuron model and observe its behavior on binary classification problems. In the lab, an experiment is conducted with linearly separable and inseparable data to demonstrate the limits of the perceptron's capabilities. **SDS:** The student explains in writing how the perceptron differs from logistic regression, where these models are similar, and where they differ in meaning. Additionally, they are required to describe why the historical development of neural networks required a transition to multilayer architectures. **Independent Student Work as Part of an End-to-End Project** : The first neural network hypothesis is formed for the project: it is assessed whether there are grounds for using a neural model instead of classical methods. The student documents the expected benefits and possible risks of such a transition. **LLM** can be used as an explanatory tool for visualizing neuron logic and for commenting on the code implementing the simplest model. The effect is increased accessibility of abstract concepts and a faster transition from theory to implementation. **Educational impact:** Students understand how the concept of a neural network grows from the simplest computational element. This provides a foundation for the transition to multilayer networks and deep learning.	1	1	6
11	**Theory: The artificial neuron** is studied as the basic computational element of neural network models, as well as the perceptron as a historically important and conceptually simple neural architecture. The ideas of linear separation, activation, and limitations of single-layer models are discussed. **Practice** : During practical classes, students implement a simple artificial neuron model and observe its behavior on binary classification problems. In the lab, an experiment is	1	1	6

Week	Content	Lect. (hours)	Practice (hours)	SDS (hours)
	conducted with linearly separable and inseparable data to demonstrate the limits of the perceptron's capabilities. **SDS:** The student explains in writing how the perceptron differs from logistic regression, where these models are similar, and where they differ in meaning. Additionally, they are required to describe why the historical development of neural networks required a transition to multilayer architectures. **Independent Student Work as Part of an End-to-End Project** : The first neural network hypothesis is formed for the project: it is assessed whether there are grounds for using a neural model instead of classical methods. The student documents the expected benefits and possible risks of such a transition. **LLM** can be used as an explanatory tool for visualizing neuron logic and for commenting on the code implementing the simplest model. The effect is increased accessibility of abstract concepts and a faster transition from theory to implementation. **Educational impact of the week:** Students understand how the concept of a neural network grows from the simplest computational element. This provides a foundation for the transition to multilayer networks and deep learning.			
12	**Theory:** The architecture of multilayer neural networks, the role of hidden layers and activation functions in modeling complex relationships, is examined . It is also discussed why increasing depth and nonlinearity allows neural networks to approximate significantly more complex functions than linear models. **Practice:** During seminars, students compare the behavior of single-layer and multilayer architectures on the same problem. In lab work, simple MLP models are built and trained, and the influence of the number of layers, the number of neurons, and the activation function is analyzed. **SDS:** The student conducts an experimental comparison of several MLP configurations and describes how changing the architecture affects the quality and stability of training. It is important not only to obtain the result but also to justify why one option was superior to another. **Independent Student Work as Part of an End-to-End Project** : If the project task allows for the use of a multilayer network, the student implements the first working MLP model and compares it with previously constructed approaches. The result is an updated map of model variants with a preliminary choice of future work. **LLM in the educational process** helps formulate hypotheses for architectural changes, provides commentary on	1	1	6

Week	Content	Lect. (hours)	Practice (hours)	SDS (hours)
	differences between configurations, and can be used for initial refactoring of experimental code. The result is an accelerated research cycle, but with mandatory manual verification of hypotheses and results. **Educational benefit:** Students understand how network depth enhances the expressive power of a model. A transition from classical machine learning to purely neural network thinking is formed.			
13	**Theory:** Gradient descent, error computation, and backpropagation as the basis for training neural networks are explored . Intuitive concepts of the error surface, optimization steps, and the influence of hyperparameters on learning dynamics are explored. **Practice:** During practical classes, students observe the network training process and analyze the change in the loss function across epochs. In the lab, experiments with learning are performed. rate , batch size , and number of epochs, typical scenarios of unstable and stable learning are recorded. **SDS:** The student receives a set of learning curves and must explain which of them correspond to stable optimization, and which indicate problems with the training setup. Additionally, they are required to independently conduct one experiment with hyperparameters and interpret the result. **Independent student work within the framework of an end-to-end project.** For the project, a training procedure for a neural network model is configured, basic hyperparameters and results are recorded. The student begins to keep a structured experiment log, which will later become part of the project defense. **Use of LLM in the educational process** . LLM is used to explain the behavior of learning curves, generate hypotheses for tuning hyperparameters , and assist in the design of the experiment log. The educational effect is the development of the ability to think in the logic of the experiment, and not only in the logic of a single model run. **Educational effect of the week.** Students understand the basic mechanism of neural network training and acquire an understanding that successful learning is engineering. a configurable process, not an automatic procedure.	1	1	6
14	**Theory: An overview of key deep learning architectures** is provided : convolutional networks, recurrent networks, and attention-based models. It is discussed that different architectural families correspond to different types of data and tasks—images, sequences, text, and complex multimodal scenarios.	1	1	6

Week	Content	Lect. (hours)	Practice (hours)	SDS (hours)
	Practice : In seminars, students compare architectures with task types and discuss why it is impossible to mechanically apply the same model to all data. The lab provides a comparative analysis of several ready-made architectures at the level of inputs, internal blocks, and outputs. **SDS:** The student prepares a brief structured overview of three architectural families and indicates the tasks for which each is particularly effective. It is important to highlight not only the differences but also the general logic of the transition from universal ideas to specialized architectures. **Independent Student Work as Part of an End-to-End Project:** The student correlates their project task with typical families of DL architectures and identifies which of them are potentially relevant. This becomes the basis for choosing a direction in the third semester. **LLM in the educational process** serves as a tool for comparative descriptions of architectures, generating comparison tables, and quickly preparing a set of self-assessment questions. The result is better structuring of complex material and a faster transition from overview knowledge to meaningful architectural choices. **Educational impact of the week:** Students gain a holistic understanding of the deep learning landscape. They develop an understanding that model architecture should be chosen based on the nature of the data and the task, not on fashion or name recognition.			
15	**Theory: The life cycle of a machine learning model** is examined : problem definition, data, training, validation, implementation, maintenance, and updating. The concept of an ML pipeline is introduced as a coherent engineering sequence of actions that ensures the creation and operation of a model within a digital system. **Practice:** The practical classes discuss typical pitfalls of the "laboratory" approach to ML, when the model exists separately from the system architecture. In the lab, students design a comprehensive ML pipeline diagram for their projects and identify key quality control points. **SDS:** The student describes the life cycle of a selected machine learning model or project in terms of engineering stages and risks. It is required to demonstrate that the model is not only a training stage but also an object of maintenance and integration. **Independent Student Work within an End-to-End Project** The first architectural diagram of the project's ML component is formed : input data, processing, training, model storage, inference , and quality assessment. This diagram becomes the basis for the defense at the end of the second semester.	2	1	6

Week	Content	Lect. (hours)	Practice (hours)	SDS (hours)
	LLM is used in the educational process to generate templates for architectural diagrams, risk lists, and self-assessment questions for the model lifecycle. The result is a shift from a fragmented understanding of individual algorithms to a systemic understanding of ML as an engineering process. **Educational impact of the week:** Students understand that a machine learning model lives within the lifecycle of a digital system. This develops a holistic understanding of the subject matter as the engineering of an ML/DL component, not simply a set of algorithms.			
16	**Examination week/**The exam is a combined exam that includes two interconnected parts: theoretical and practical. The theoretical part is designed to test understanding of the key concepts covered in the second semester, while the practical part focuses on defending the student's work on the end-to-end project. **Theoretical Exam:** The student answers questions on defining machine learning problems, data engineering, errors and model generalization, quality metrics, linear models, ensembles, clustering, dimensionality reduction, basic principles of neural networks, and the life cycle of an ML model. This assessment evaluates not only knowledge of definitions but also the ability to connect theory with engineering application scenarios. **Practical Defense:** The student presents the preliminary stage of the end-to-end project: an applied problem statement, data description, a baseline model, one or more alternative models, justification for the choice of metrics, and an analysis of quality and limitations. A mandatory element of the defense is a discussion of model errors, data issues, and the logic behind further project development in the third semester. **Results submitted for the exam** : The following are presented for defense: project problem statement, dataset description, experiment log, model comparison table, selected metrics, error analysis, and a high-level ML pipeline diagram. The assessment takes into account the correctness of the problem formalization, the meaningfulness of the model selection, and the student's ability to explain their engineering decisions. **LLM can be used in the educational process and during** the LLM preparation stage as a tool for preparing for oral responses, generating practice questions, structuring presentations, and identifying weaknesses in argumentation. The educational effect lies in practicing reflection and	0	6	30

Week	Content	Lect. (hours)	Practice (hours)	SDS (hours)
	self-assessment, but the final defense is based solely on the student's own understanding and actual work. **Educational outcome of the semester:** By the end of the second semester, the student demonstrates an understanding of the basic principles of machine learning and initial experience in constructing and evaluating models. They are able to formalize a problem, work with data, compare models, and explain the logic behind their selection in the context of a digital system.			

Semester 3

Week	Content	Theory	Practice	SDS
1	**Theory:** The operating principles of convolutional neural networks (CNNs) as architectures designed for processing images and other spatially organised data are explored. Key topics: — convolution, — kernel, — pooling, — feature maps, — hierarchical feature extraction. **Practice**: — In seminars, students compare fully connected and convolutional architectures and discuss why images require a special model structure. — In the lab, a basic CNN model is implemented and trained for a simple image classification task. **SDS (Current Homework)**: — Prepare a brief explanation of why a convolutional architecture is more appropriate for images than a fully connected network of the same scale. — Analyse the model's results and errors using specific examples. Independent Student Work (End-to-End Project): — If the project involves visual data, establish a CNN as a working architecture. — If not, compare the project with possible specialised architectures. **Educational Impact:** Students understand how model architecture should relate to the structure of input data. A basic understanding of specialised neural network families and their role in Computer Vision is developed.	2	2	6
2	**Theory:** The evolution and principles of modern computer vision architectures are examined: — deep CNNs, — residual connections, — network scaling, — generalised patterns for constructing visual models. Examples: ResNet, EfficientNet and other families as architectural responses to depth and computational cost limitations. **Practice**: — Seminars: analyse architectural solutions that have enabled deeper and more reliably trained visual networks. — Lab: compare several architectures in terms of quality, computational complexity, and suitability for practical application. **SDS (Current Homework)**: — Analytical note on how architectural innovations in computer vision relate to engineering	2	2	6

Week	Content	Theory	Practice	SDS
	limitations (depth, speed, memory, quality). — Focus on understanding the logic behind models, not just their names. Independent Student Work (End-to-End Project): — Select a modern architecture as a reference or for comparison with a simpler model (if project involves visual data). — Otherwise, conduct analytical groundwork for a future Computer Vision course. **Educational Impact**: Students see that deep learning development responds to real-world limitations of model training and operation. They understand how architectural evolution occurs under practical requirements.			
3	**Theory:** Exploration of problems where data is sequential in nature: — time series, — text, — signals, — events. Key distinctions: — independent observations vs. sequential data, — importance of order and context. **Practice**: — Seminars: examine forecasting problems, event sequence analysis, text processing. — Lab: explore a simple model on sequential data; analyse what information is lost when order is ignored. **SDS (Current Homework)**: — Comparative analysis: which data is independent vs. sequential, and why this affects model architecture choice. — Explain both formal distinction and practical implications. Independent Student Work (End-to-End Project): — Evaluate if the problem has a sequential structure. — Outline requirements for a model that accounts for order and context (part of architectural profile). **Educational Impact:** Students distinguish between models for “static” and “sequential” data. This forms the basis for transitioning to recurrent networks, attention mechanisms, and NLP.	2	2	6
4	**Theory:** Recurrent neural networks (RNNs) as architectures for processing sequences using internal state. Topics: — limitations of basic RNNs, — rationale for LSTM (Long Short-Term Memory) for robust long-term dependency processing. **Practice** Exercises: — Compare classic RNNs with advanced memory architectures. — Analyse problems with vanishing/exploding gradients. — Lab: train a simple recurrent model on sequential data and analyse its behaviour. **SDS (Current Homework)**: — Explain why recurrent models require memory. — Analyse when LSTM is justified vs. inferior to modern alternatives. Independent Student Work (End-to-End Project): — Consider RNN/LSTM as architecture options (if project has sequential structure). — Develop understanding of problems requiring memory architectures (even if project is unrelated). **Educational Impact:** Students understand how models work with memory and context. This provides foundation for discussing attention mechanisms and Transformers.	2	2	6

Week	Content	Theory	Practice	SDS
5	**Theory:**Attention mechanism as a way to dynamically highlight relevant elements of an input sequence based on task context. Key ideas: — transition from “fixed memory” to context-dependent information selection. **Practice**: — Seminars: explore attention intuition via translation, summarisation, sequence analysis. — Lab: simplified attention mechanism example with visualisation of attention weights. **SDS (Current Homework)**: — Written explanation: why attention became a turning point in neural architecture development. — Connect attention to general architectural logic (beyond NLP). Independent Student Work (End-to-End Project): — Evaluate if project requires complex context handling and architectural implications. — Record as part of extended architectural profile. **Educational Impact**: Students grasp attention as a universal architectural mechanism, not just an NLP detail.	1	2	6
6	**Theory:**Transformer architecture as a system built around attention, rejecting recurrent sequence processing. Components: — encoder, decoder, — positional coding, — role as foundation of modern large models. **Practice**: — Seminars: compare Transformer with recurrent models; discuss advantages in training/scaling. — Lab: structural analysis of Transformer architecture (input, layer, output levels). **SDS (Current Homework)**: — Coherent engineering description of Transformer: constituent blocks, success factors, limitations, and costs. Independent Student Work (End-to-End Project): — Assess Transformer potential (if project involves text/sequences/complex context). — Establish conceptual connection to future NLP/Foundation Models courses (if not applicable). **Educational Impact:** Students gain systemic understanding of Transformers as foundation of modern AI ecosystem. Bridge between classical deep learning and large-scale model engineering is created.	1	2	6
7	**Theory**: Pre-training and adaptation paradigm: — transfer learning, — fine-tuning, — feature extraction, — economics of reusing pre-trained models. **Practice**: — Seminars: scenarios where pre-trained models benefit quality, data, and computing resources. — Lab: use pre-trained model as feature source or adaptation basis. **SDS (Current Homework)**: — Analytical analysis: when transfer learning is optimal vs. not advantageous. — Compare architecture, data, and resources. Independent Student Work (End-to-End Project): — Evaluate use of pre-trained model: fixed feature extractor, training basis, or part of complex system. — Present as design justification. **Educational Impact:** Students understand modern AI engineering increasingly relies on adapting trained models, not	1	2	6

Week	Content	Theory	Practice	SDS
	building from scratch. Mature understanding of cost, scalability, and model reuse is formed.			
8	**Theory:** Fundamental models are considered as large, pre-trained models that serve as a universal basis for a wide range of downstream tasks. The properties of such models are discussed: scale, pre-training on large datasets, versatility, adaptability, and their role in modern AI system architecture. **Practice** : In seminars, students discuss how fundamental models differ from highly specialized architectures and why they have become a new technological paradigm. The lab analyzes an application scenario in which a single fundamental model can be adapted to multiple tasks. **SDS (Current Homework):** The student prepares a short essay or analytical note on why fundamental models have changed engineering practice in AI. The distinction between a "large model" and a "fundamental model" as an architectural and ecosystem category is specifically required. **Independent student work within an end-to-end project.** The student evaluates whether their project can be implemented using a fundamental model and what advantages or limitations this creates in terms of data, cost, speed of implementation, and explainability. This becomes an important project decision . **LLM in the educational process** . The LLM is used not only as a subject for discussion but also as a metacognitive tool : the student analyzes familiar large language models as special cases of fundamental models. The effect is the integration of the previously studied LLM course with ML/DL engineering logic and a deeper understanding of the modern AI paradigm. **Educational impact:** Students understand fundamental models as a new foundation for building AI systems. A holistic picture emerges of the transition from individual architectures and specialized models to scalable base platforms for applied artificial intelligence.	1	2	6
9	**Theory:** Explores the ML pipeline as a reproducible engineering sequence of model processing: data, preparation, training, evaluation, storage, deployment, and updating. Reproducibility, data and model versioning, and the discipline of recording experiments as part of engineering maturity are discussed. **Practice** : Students analyze common errors . The hoc approach, where the model is created as a one-time experiment without a reproducible infrastructure. In the lab, a comprehensive pipeline is designed for individual projects, highlighting key stages and dependencies. **SDS (Current Homework):** The student is asked to describe the ML pipeline of a single application system and identify its critical points for the quality and stability of the result. Particular attention is paid to how a violation of reproducibility undermines the engineering reliability of the solution.	1	1	6

Week	Content	Theory	Practice	SDS
	Independent Student Work within an End-to-End Project: The project receives a full-fledged process structure: the stages of data preparation, training, evaluation, and preservation of artifacts are recorded. The student develops the first diagram of the model's life cycle as an engineering object. **Using LLM in the Educational Process:** LLM can be used to generate pipeline description templates , a list of checkpoints, and a set of questions for reproducibility audits. The effect is to help systematize the process and develop the ability to see the entire contours of engineering activity beyond the "model code." **Educational impact:** Students begin to think not in terms of isolated experiments, but in terms of reproducible processes. The course develops connections with DevOps , Big Data, and the subsequent logic of MLOps approaches.			
10	**Theory:** Experiment management is considered as a mandatory part of ML system development: recording hypotheses, parameters, datasets, results, and model versions. It is discussed why engineering work with models is impossible without the discipline of traceability of decisions and reproducible comparisons. **Practice:** During seminars, students analyze typical "experiment loss" situations, when it is impossible to explain why one model version turned out to be better than another. In the laboratory, project experiment logging is organized, recording parameters, quality, and comments on the results. **SDS (Current Homework):** The student formats several conducted experiments in a single standardized format and prepares a comparative conclusion based on them. An important task is not simply to accumulate results, but to transform them into an engineering- ready decision-making history. **Independent Student Work within an End-to-End Project** : The project introduces mandatory experiment logging and recording of model versions, datasets, and training configurations. This becomes one of the artifacts of the final defense. **LLM is used in the educational process** to help structure journals, generate standardized entry templates, and identify gaps in experiment descriptions. The result is a stronger culture of project documentation and reflection on one's own engineering solutions. **Educational impact:** Students understand that high-quality modeling requires not only programming and computation, but also rigorous experiment management discipline. This lays the foundation for a mature engineering process in ML.	1	1	6
11	**Theory :** The transition from a trained model to a working component of a digital system is studied : model serialization, API access, batch and online inference , and runtime limitations.	1	1	6

Week	Content	Theory	Practice	SDS
	The differences between a "model in a laptop" and a "model in a service" are discussed, as well as the risks of implementation. **Practical:** Practical classes examine scenarios for deploying a model as a separate service, an embedded module, or a component of a more complex architecture. In the lab, students design a model deployment scheme for their project and describe the requirements for integration with an external system. **SDS (Current Homework):** The student prepares a description of the architectural variant of model deployment, including the input and output format, latency requirements, and resource constraints. An important part of the assignment is justifying the chosen implementation format for the application scenario. **Independent Student Work within an End-to-End Project :** A target operational scenario is defined for the project : batch processing, an online service, an internal analytical circuit, etc. The student records the model implementation scheme and a list of technical constraints. **LLM is used in the educational process** to generate service architectural templates, prepare API descriptions, and integration test scenarios. The result is a strengthened engineering mindset regarding the system as a product, not as a set of code files. **Educational impact:** Students understand that model implementation is an independent engineering task requiring the design of interfaces, constraints, and operational conditions. The ML course is linked to the program's architectural and DevOps disciplines.			
12	**Theory:** Computational limitations of model operation are considered : latency, throughput, memory, model size, and inference cost . Quantization, distillation, architecture simplification, and other approaches to improving efficiency without completely losing quality are discussed. **Practice** : During seminars, students analyze engineering tradeoffs between model quality and performance requirements. In the lab, several models or configurations are compared based on accuracy, speed, and resource consumption. **SDS (Current Homework):** The student prepares a brief report on the quality-speed-resources tradeoff for a selected model or family of models. The assignment should demonstrate that model selection in a real system depends not only on accuracy but also on operational constraints. **Student Independent Work as Part of an End-to-End Project:** Acceptable requirements for latency, model size, and operational format are established for the project. Given several model options, the student begins to evaluate them not only by quality but also by cost of use. **Using LLM in the educational process** helps structure comparative performance analysis, generates templates for	1	1	6

Week	Content	Theory	Practice	SDS
	tradeoff tables, and can suggest scenarios where inference optimization becomes critical. The effect is the development of the ability to view a model as a technical component with measurable resource characteristics. **Educational impact:** Students understand that a model's effectiveness is measured by more than just the quality of its predictions. They develop an engineering mindset in which operational constraints become part of architectural choices.			