

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО
решением Ученого совета ННГУ
протокол № 10 от 27.08.2025

Рабочая программа дисциплины

Разработка и внедрение моделей машинного обучения

Уровень высшего образования
Бакалавриат

Направление подготовки / специальность
02.03.02 - Фундаментальная информатика и информационные технологии

Направленность образовательной программы
Системное программирование

Форма обучения
очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.ДВ.06.02 Разработка и внедрение моделей машинного обучения относится к части, формируемой участниками образовательных отношений образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ПК-Ф1: Способен планировать и организовывать аналитические работы с использованием технологий больших данных	<p>ПК-Ф1.1: Применяет методы машинного обучения и статистического анализа. Знает типы анализа больших данных, виды аналитики; теоретические и прикладные основы анализа больших данных; содержание этапов жизненного цикла больших данных</p> <p>ПК-Ф1.2: Обеспечивает соответствие результатов анализа бизнес-задачам заказчика. Умеет планировать и проводить аналитические работы с использованием технологий больших данных</p> <p>ПК-Ф1.3: Подготавливает отчеты и визуализации для презентации результатов. Умеет проводить анализ больших данных</p>	<p>ПК-Ф1.1:</p> <p>ПК-Ф1.1. 3-1. Знает основные алгоритмы машинного обучения</p> <p>ПК-Ф1.1. 3-2. Знает методы статистического анализа данных</p> <p>ПК-Ф1.1. 3-3. Знает критерии выбора алгоритмов для различных задач</p> <p>ПК-Ф1.1. У-1. Умеет реализовывать алгоритмы машинного обучения</p> <p>ПК-Ф1.1. У-2. Умеет интерпретировать результаты статистического анализа</p> <p>ПК-Ф1.2:</p> <p>ПК-Ф1.2. 3-1. Знает методы перевода бизнес-требований в аналитические задачи</p> <p>ПК-Ф1.2. 3-2. Знает ключевые бизнес-метрики в предметной области</p> <p>ПК-Ф1.2. 3-3. Знает принципы интерпретации результатов для бизнес-пользователей</p> <p>ПК-Ф1.2. У-1. Умеет адаптировать аналитические модели под бизнес-потребности</p> <p>ПК-Ф1.2. У-2. Умеет оценивать экономический эффект от аналитических решений</p>	Тест	Зачёт: Контрольные вопросы

		<p>ПК-Ф1.3:</p> <p>ПК-Ф1.3 3-1. Знает принципы эффективной визуализации данных</p> <p>ПК-Ф1.3 3-2. Знает инструменты создания аналитических отчетов</p> <p>ПК-Ф1.3 3-3. Знает методы сторителлинга на основе данных</p> <p>ПК-Ф1.3 У-1. Умеет выбирать оптимальные типы визуализации</p> <p>ПК-Ф1.3 У-2. Умеет создавать интерактивные дашборды</p>		
<p>ПК-Ф2: Способен проектировать, разрабатывать, внедрять, развертывать и управлять моделями машинного обучения</p>	<p>ПК-Ф2.1: Знает основные алгоритмы и методы машинного обучения</p> <p>ПК-Ф2.2: Знает основные концептуальные и теоретические модели искусственного интеллекта и машинного обучения</p> <p>ПК-Ф2.3: Умеет использовать методы машинного обучения на практике, оценивать качество методов, разрабатывать и анализировать концептуальные и теоретические модели машинного обучения</p>	<p>ПК-Ф2.1:</p> <p>ПК-Ф2.1. Знает основные алгоритмы и методы машинного обучения, основы языка Python или среды вычислений R</p> <p>ПК-Ф2.2:</p> <p>ПК-Ф2.2. Знает основные концептуальные и теоретические модели искусственного интеллекта и машинного обучения:</p> <p>Обучение с учителем</p> <p>Обучение без учителя</p> <p>Обучение с подкреплением</p> <p>Обучение с частичным участием учителя</p> <p>Глубокое обучение</p> <p>ПК-Ф2.3:</p> <p>ПК-Ф2.3. Умеет использовать методы машинного обучения на практике, оценивать качество методов, работать с библиотекой Scikit-Learn или средой для статистических вычислений R, разрабатывать и анализировать концептуальные и теоретические модели машинного обучения</p>	<p>Задачи</p>	<p>Зачёт:</p> <p>Контрольные вопросы</p>
<p>ППК-Р7: Способен</p>	<p>ППК-Р7.1: Применяет ИИ-</p>	<p>ППК-Р7.1:</p>	<p>Практическая</p>	

<p>применять искусственный интеллект (ИИ) для генерации и отладки программного кода (top)</p>	<p>инструменты для генерации программного кода ППК-Р7.2: Использует ИИ для анализа и отладки кода ППК-Р7.3: Оптимизирует код с помощью ИИ ППК-Р7.4: Оценивает этические и профессиональные аспекты применения ИИ в разработке</p>	<p>ППК-Р7.1. 3-1. Знает принципы работы современных генеративных ИИ-моделей для генерации кода ППК-Р7.1. 3-2. Знает ограничения и риски использования ИИ-генерации (безопасность, качество кода, лицензирование) ППК-Р7.1. У-1. Умеет формулировать корректные текстовые запросы (промты) для генерации кода ППК-Р7.1. У-2. Умеет интегрировать ИИ-инструменты в среду разработки</p> <p>ППК-Р7.2: ППК-Р7.2. 3-1. Знает методы ИИ-анализа кода ППК-Р7.2. 3-2. Знает форматы и инструменты для автоматизированного тестирования с ИИ ППК-Р7.2. У-1. Умеет настраивать ИИ-инструменты для поиска уязвимостей ППК-Р7.2. У-2. Умеет интерпретировать рекомендации ИИ по исправлению кода</p> <p>ППК-Р7.3: ППК-Р7.3. 3-1. Знает методы ИИ-оптимизации ППК-Р7.3. 3-2. Знает критерии качества кода, применяемые ИИ-системами ППК-Р7.3. У-1. Умеет использовать ИИ для рефакторинга ППК-Р7.3. У-2. Умеет проверять корректность оптимизаций, предложенных ИИ</p> <p>ППК-Р7.4: ППК-Р7.4. 3-1. Знает этические нормы использования ИИ</p>	<p>задача</p>	<p>Зачёт: Контрольные вопросы</p>
---	--	--	---------------	--

		(конфиденциальность, плагиат кода и т.п.) ППК-Р7.4. 3-2. Знает лицензионные ограничения сгенерированного кода ППК-Р7.4. У-1. Умеет проверять код на соответствие стандартам после ИИ-генерации ППК-Р7.4. У-2. Умеет документировать использование ИИ в разработке		
--	--	--	--	--

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	2
Часов по учебному плану	72
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	16
- занятия семинарского типа (практические занятия / лабораторные работы)	32
- КСР	1
самостоятельная работа	23
Промежуточная аттестация	0 Зачёт

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			Самостоятельная работа обучающегося, часы
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабораторные работы), часы	Всего	
Ф	Ф	Ф	Ф	Ф	
GitHub - технология.	11	3	5	8	3
Современные стили программирования.	12	3	5	8	4
Разработка через тестирование	13	3	6	9	4

Стандарты C++ программирования	13	3	6	9	4
Интегрирование ИИ-инструментов в среду разработки	11	2	5	7	4
Использование технологий больших данных в бизнес-процессах	11	2	5	7	4
Аттестация	0				
КСР	1			1	
Итого	72	16	32	49	23

Содержание разделов и тем дисциплины

Цели и задачи дисциплины.

Сформировать у обучающихся системные знания и практические навыки в области современных технологий разработки программного обеспечения, управления кодом, архитектуры приложений, а также внедрения ИИ-инструментов и технологий больших данных в бизнес-процессы. Научить применять лучшие практики программирования, проектирования и командной работы для повышения качества, поддерживаемости и эффективности программных продуктов.

Содержание разделов и тем.

Введение. GitHub - технология. Открытые источники комментированного кода в задачах компьютерного зрения и глубокого обучения. Современные стили программирования. Роль стиля в качестве кода.

Несовместимость версий. Книга Roberts C. Martin «Clean Code», 2008. Практики написания и поддержания чистого кода. Ключевые понятия качества кода, их влияние. Примеры влияния нарушения singleresponsibility principle (SRP) на чистоту кода. Недостатки использования комментариев. Разработка через тестирование (Test-Driven Development). Что такое TDD, его назначение. Цикл разработки по TDD. Атрибуты хорошего unit-теста. Виртуальные машины и контейнеры. Концепция докер-контейнеров: виртуализация операционных систем, группы контейнеров, выделенные корни, виртуальные сетевые адаптеры. Докер-соглашения: dockerfile; docker build docker run; volumes; docker registry: hub.docker.com, registry.gitlab.com, etc. Докер-практики: Distribution package; CI/CDenvironment; “Clean Room”.

Принципы ООП. SOLID: SRP, OCP, DIP (его реализация, связь с OCP), LSP, ISP, DIP в DDD. Закон Деметера. Стандарты C++ программирования. Паттерн Domain-Driven Design. Диаграмма DDD. Назначение сборки Application. Способ стыковки Domain и Infrastructure. DIP в DDD. Дизайн слоя доступа к данным. Типичные проблемы архитектуры при отсутствии слоя Data Access. Способы организации Data Access. Преимущества Data Access Layer. Дизайн презентационного слоя (паттерн Model-View-Presenter). Типичные проблемы с UI. Диаграмма Passive View. Назначение и преимущества применения Passive View. Методология «Экстремальное программирование» (XP). Ценности и практики XP. Стратегия управления и кодирования. Парное программирование. Методология Scrum и схема спринта. Методология Kanban. Рефакторинг.

Интегрирование ИИ-инструментов в среду разработки. Классификация ИИ-инструментов (автодополнение, анализ кода, генерация тестов и др.). Примеры популярных платформ и сервисов (GitHub Copilot, Tabnine, Kite и др.). Сравнительный анализ функциональности и эффективности. Принципы встраивания ИИ-инструментов в IDE. API и плагины: технические аспекты интеграции. Совместимость с различными языками программирования и фреймворками. Этапы внедрения ИИ-инструментов в проект. Примеры кейсов и сценариев использования.

Использование технологий больших данных в бизнес-процессах. Основные технологии и инструменты (Hadoop, Spark, NoSQL и др.). Улучшение принятия управленческих решений. Оптимизация операций и логистики. Архитектура решений и выбор платформ. Проблемы безопасности, конфиденциальности и качества данных.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Самостоятельная работа обучающихся осуществляется в виде работы с рекомендованной обязательной и дополнительной литературой:

1. Гибкая методология разработки программного обеспечения. Курс на Интуит: <http://www.intuit.ru/studies/courses/583/439/info>.
2. Технологии командной разработки программного обеспечения информационных систем. Курс на Интуит: <http://www.intuit.ru/studies/courses/4806/1054/info>.
3. Анализ и оценка методов разработки программного обеспечения (Agile). Курс на Интуит: <http://www.intuit.ru/studies/courses/3505/747/info>.

Дополнительная литература:

1. Скопин И. Основы менеджмента программных проектов. <http://www.intuit.ru/studies/courses/38/38/info>.
2. Meyer В. Объектно-ориентированное программирование и программная инженерия. Курс на Интуит: <http://www.intuit.ru/studies/courses/2309/609/info>.
3. Кознов Д. Введение в программную инженерию. Курс на Интуит: <http://www.intuit.ru/studies/courses/497/353/info>

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ПК-Ф1:

1. Что из перечисленного не относится к основным технологиям больших данных?

- а) *Hadoop*
- б) *Spark*
- в) *SQL Server*
- г) *NoSQL*

2. Для чего используется фреймворк *Hadoop*?

- а) Для разработки мобильных приложений
- б) Для распределённого хранения и обработки больших объёмов данных
- в) Для создания презентаций
- г) Для ведения бухгалтерии

Критерии оценивания (оценочное средство - Тест)

Оценка	Критерии оценивания
зачтено	Выполнено верно более 80% вопросов

Оценка	Критерии оценивания
не зачтено	Выполнено менее 80% вопросов

5.1.2 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ПК-Ф2:

1. Дана обучающая выборка

x1	0 2 1 1 1 2 4 4 4 6
x2	4 4 3 5 4 1 0 2 1 1
y	0 0 0 0 0 1 1 1 1 1

Методом линейного дискриминантного анализа для каждого класса построить дискриминантную функцию и записать уравнение разделяющей поверхности.

2. Дана обучающая выборка (см. таблицу выше). Методом квадратичного дискриминантного анализа построить дискриминантные функции.

Критерии оценивания (оценочное средство - Задачи)

Оценка	Критерии оценивания
зачтено	Задача решена полностью или решена основная часть задачи, или задача решена с недочетами
не зачтено	Задача не решена или сделан первый этап решения задачи

5.1.3 Типовые задания (оценочное средство - Практическая задача) для оценки сформированности компетенции ППК-Р7:

Практическое задание для каждого студента — индивидуальное задание по обеспечению переносимости DL-системы средней сложности с использованием Docker и других изученных технологий, включая автотестирование. В качестве исходной системы глубокого обучения

может быть выбрано собственное DL-приложение или приложение, выбранное с сайта приложений с открытым исходным кодом (<https://paperswithcode.com/>). Выбор согласовывается с преподавателем. Проект представляется автором всем участникам практического занятия в режиме on-line. В течение представления, инструкция по запуску проекта, и проверке корректности результата, тестируется студентом, назначенным преподавателем.

Критерии оценивания (оценочное средство - Практическая задача)

Оценка	Критерии оценивания
зачтено	Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Результаты работы представлены преподавателю в срок.
не зачтено	Выполнены не все практические задания или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, получен неверный ответ, результаты работы не представлены преподавателю).

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатор достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено			зачтено			
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но	Продемонстрированы все основные умения. Решены все основные задачи с отдельным и несущест	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов

			не в полном объеме	в полном объеме, но некоторые с недочетами	некоторые с недочетами	енными недочетам и, выполнены все задания в полном объеме	
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-Ф1

1. Основные технологии и инструменты для работы с большими данными.
2. Роль *Hadoop* в распределённом хранении и обработке данных.

3. Преимущества *Apache Spark* для анализа данных в реальном времени.
4. Типы *NoSQL*-баз данных, используемые в бизнес-процессах.
5. Влияние технологий больших данных на принятие управленческих решений.
6. Оптимизация логистики с помощью анализа больших данных.
7. Ключевые компоненты архитектуры решений на базе больших данных.
8. Критерии выбора платформы для внедрения больших данных в бизнесе.
9. Основные проблемы безопасности при работе с большими данными.
10. Значение качества данных для эффективности бизнес-процессов.

5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-Ф2

1. Нейронные сети. Персептрон Розенблатта. Алгоритм обучения персептрона как метод стохастического градиентного спуска. Нейронные сети для решения задач классификации и восстановления регрессии. Обучение сети. Регуляризация как метод борьбы с переобучением. Понятие о глубоких нейронных сетях.
2. Машина опорных векторов. Ядра и спрямляющие пространства.
3. Деревья решений. Метод CART (classification and regression trees) для решения задач классификации и восстановления регрессии. Отсечения ветвей и выбор финального дерева. Методы обработки пропущенных значений.
4. Ансамбли решающих правил (классификаторов). Простое и взвешенное голосования. Бустинг. Алгоритм AdaBoost. Оценка ошибки предсказания. Бустинг и аддитивные модели. Градиентный бустинг. Алгоритм градиентного бустинга деревьев решений (MART). Баггинг. Алгоритм случайных деревьев (случайный лес.).
5. Обучение без учителя. Кластеризация. Кластеризация методами теории графов. Метод центров тяжести. Метод медиан. Метод нечетких множеств. EM-алгоритм.
6. Иерархическая кластеризация. Агломеративные и разделяющие методы.
7. Основы теории Вапника–Червоненкиса. Лемма Бернштейна. Теорема о равномерной сходимости эмпирического риска к ожидаемому риску в случае конечного класса решающих правил. Обоснование принципа минимизации эмпирического риска.
8. Размерность Вапника–Червоненкиса. Лемма Зауэра. Теорема о равномерной сходимости эмпирического риска к ожидаемому риску в случае конечной размерности Вапника–Червоненкиса. Принцип структурной минимизации риска.

5.3.3 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ППК-Р7

1. GitHub – технология.
2. Открытые источники комментированного кода в задачах компьютерного зрения и глубокого обучения.
3. Современные стили программирования.
4. Роль стиля в качестве кода. Книга Roberts C. Martin «Clean Code», 2008.

5. Несовместимость версий.

6. Практики написания и поддержания чистого кода. Ключевые понятия качества кода, их влияние.

7. Примеры влияния нарушения single-responsibility principle (SRP) на чистоту кода.

Недостатки использования комментариев.

8. Разработка через тестирование (Test-Driven Development). Что такое TDD, его назначение. Цикл разработки по TDD. Атрибуты хорошего unit-теста / Test-Driven Development.

9. Виртуальные машины и контейнеры. Концепция докер-контейнеров: виртуализация операционных систем, группы контейнеров, выделенные корни, виртуальные сетевые адаптеры.

10. Докер-соглашения: dockerfile; docker build docker run; volumes; docker registry: hub.docker.com, registry.gitlab.com, etc.

11. Докер-практики: Distribution package; CI/CD environment; "Clean Room".

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Владение основным и дополнительным материалом достаточное или с незначительными ошибками и погрешностями
не зачтено	владение материалом, необходимым по данному предмету, недостаточно. Работу за время семестра можно оценить как неудовлетворительную

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Барков И. А. Объектно-ориентированное программирование : учебник для вузов / Барков И. А. - 2-е изд., стер. - Санкт-Петербург : Лань, 2023. - 700 с. - Книга из коллекции Лань - Информатика. - ISBN 978-5-507-47113-3., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=864800&idb=0>.

Дополнительная литература:

1. Зыков С. В. Объектно-ориентированное программирование : учебник и практикум / С. В. Зыков. - 2-е изд. - Москва : Юрайт, 2023. - 151 с. - (Высшее образование). - ISBN 978-5-534-16941-6. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=870794&idb=0>.

2. Объектно-ориентированное программирование / Иванова Г.С., Ничушкина Т.Н. - Москва : МГТУ

им. Н.Э. Баумана, 2014., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=661368&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

1. The R Project for Statistical Computing <https://www.r-project.org/>
2. Welcome to Python.org <https://www.python.org/>
3. scikit-learn: machine learning in Python scikit-learn.org/
4. Домашняя страница учебного курса <https://github.com/UNN-VMK-Software/agilecourse-theory>.

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 02.03.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Золотых Николай Юрьевич, доктор физико-математических наук, доцент.

Заведующий кафедрой: Золотых Николай Юрьевич, доктор физико-математических наук.

Программа одобрена на заседании методической комиссии от 25.06.2025, протокол № Протокол №11.