

МИНОБРНАУКИ РОССИИ  
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
НАУЧНО-ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР "ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ"

Рабочая программа дисциплины (модуля)  
**МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ**

Направление и направленность (профиль)  
09.03.04 Программная инженерия. Программная инженерия

Год набора на ОПОП  
2026

Форма обучения  
очная

Владивосток 2026

Рабочая программа дисциплины (модуля) «Методы машинного обучения» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.04 Программная инженерия (утв. приказом Минобрнауки России от 19.09.2017г. №920) и Порядком организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утв. приказом Минобрнауки России от 06.04.2021 г. N245).

Составитель(и):

*Борисов Р.П.*

*Кригер А.Б.*

Утверждена на заседании научно-образовательный центр "искусственный интеллект" от 27.05.2026 , протокол № 5

СОГЛАСОВАНО:

Заведующий кафедрой (разработчика)

Кригер А.Б.

<b>ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ</b>	
Сертификат	1582918206
Номер транзакции	0000000000F95288
Владелец	Кригер А.Б.

## 1 Цель, планируемые результаты обучения по дисциплине (модулю)

Целью освоения дисциплины «Методы машинного обучения» являются формирование у студентов компетенций в области использования технологий машинного обучения для анализа данных.

В ходе достижения цели решаются следующие задачи:

- изучение студентами стадий технологии машинного обучения;
- овладение студентами навыками работы с различными методами построения алгоритмов, способных обучаться;
- получение практических навыков реализации методов машинного обучения;
- создание основы для дальнейшего поэтапного формирования компетенций, составляющие которых перечислены и описаны в рабочей программе дисциплины.

Планируемыми результатами обучения по дисциплине (модулю), являются знания, умения, навыки. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы, представлен в таблице 1.

Таблица 1 – Компетенции, формируемые в результате изучения дисциплины (модуля)

Название ОПОП ВО, сокращенное	Код и формулировка компетенции	Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине		
			Код результата	Формулировка результата	
09.03.04 «Программная инженерия» (Б-ИН)	ОПК-6 : Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1к : Использует навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач	РД10	Знание	требований к разработке кода информационных систем и баз данных информационных систем на основе принципов машинного обучения
			РД11	Умение	разрабатывать информационные системы и базы данных информационных систем для методов машинного обучения
			РД12	Навык	разработки прототипа информационной системы и ее базы данных с применением методов машинного обучения
			РД7	Знание	способов сборки программных модулей и компонент алгоритмов машинного обучения в программный продукт
			РД8	Умение	собирать программные модули и компоненты машинного обучения в единый программный продукт, осуществлять отладку и тестирование программного продукта
			РД9	Навык	программирования программных модулей на современных языках программирования машинного обучения,

				сборки программных модулей и компонент в программный продукт, осуществлять отладку собранного программного продукта
	ОПК-6.2к : Применяет современные программные среды программирования, разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ	РД1	Знание	методов, моделей, алгоритмов, технологий и инструментальных средств машинного обучения
		РД2	Умение	использовать методы, модели, алгоритмы, технологии и инструментальные средства машинного обучения для работы с большими данными
		РД3	Навык	программирования методов, моделей, алгоритмов машинного обучения с использованием современных технологий и инструментальных средств
		РД4	Знание	требований к проведению аналитических исследований для сбора, обработки и анализа больших данных методами машинного обучения
		РД5	Умение	проводить аналитические исследования по сбору, обработке и анализу больших данных методами машинного обучения
		РД6	Навык	проведения аналитических исследований для решения практических задач методами машинного обучения

В процессе освоения дисциплины решаются задачи воспитания гармонично развитой, патриотичной и социально ответственной личности на основе традиционных российских духовно-нравственных и культурно-исторических ценностей, представленные в таблице 1.2.

Таблица 1.2 – Целевые ориентиры воспитания

Воспитательные задачи	Формирование ценностей	Целевые ориентиры
<b>Формирование научного мировоззрения и культуры мышления</b>		
Формирование осознания ценности научного мировоззрения и критического мышления	Гуманизм	Системное мышление

Формирование коммуникативных навыков и культуры общения		
Формирование навыков публичного выступления и презентации своих идей	Взаимопомощь и взаимоуважение	Умение работать в команде и взаимопомощь

## 2 Место дисциплины (модуля) в структуре ОПОП

Дисциплина «Методы машинного обучения» относится к *БИН - обязательной части Блока 1 «Дисциплины (модули)» учебного плана,*

## 3. Объем дисциплины (модуля)

Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу с обучающимися (по видам учебных занятий) и на самостоятельную работу, приведен в таблице 2.

Таблица 2 – Общая трудоемкость дисциплины

Название ОПОП ВО	Форма обучения	Часть УП	Семестр (ОФО) или курс (ЗФО, ОЗФО)	Трудо-емкость (З.Е.)	Объем контактной работы (час)					СРС	Форма аттес-тации	
					Всего	Аудиторная			Внеауди-торная			
						лек.	прак.	лаб.	ПА			КСР
09.03.04 Программная инженерия	ОФО	Б1.Б	6	3	55	18	36	0	1	0	53	3

## 4 Структура и содержание дисциплины (модуля)

### 4.1 Структура дисциплины (модуля) для ОФО

Тематический план, отражающий содержание дисциплины (перечень разделов и тем), структурированное по видам учебных занятий с указанием их объемов в соответствии с учебным планом, приведен в таблице 3.1

Таблица 3.1 – Разделы дисциплины (модуля), виды учебной деятельности и формы текущего контроля для ОФО

№	Название темы	Код ре-зультата обучения	Кол-во часов, отведенное на				Форма текущего контроля
			Лек	Практ	Лаб	СРС	
1	Основные понятия машинного обучения	РД2, РД3	2	2	0	3	реализованное практическое задание (python ноутбук)
2	Методы машинного обучения, основанные на статистических методах. Регрессии	РД2, РД3	4	6	0	10	реализованное практическое задание (python ноутбук)
3	Машинное обучение с учителем: классификация	РД2, РД3	6	4	0	8	реализованное практическое задание (python ноутбук)
4	Нейронные сети. Глубокое обучение	РД1, РД3	4	14	0	14	реализованное практическое задание (python ноутбук)

5	Решение прикладных задач методами машинного обучения	РД1, РД3	2	10	0	18	реализованное практическое задание (python ноутбук)
<b>Итого по таблице</b>			<b>18</b>	<b>36</b>	<b>0</b>	<b>53</b>	

#### 4.2 Содержание разделов и тем дисциплины (модуля) для ОФО

##### *Тема 1 Основные понятия машинного обучения.*

Содержание темы: Системы искусственного интеллекта. Связь машинного обучения с системами искусственного интеллекта. Основные модели машинного обучения. Создание обучающей выборки. Типы выборок. Переобучение. Решающие деревья.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчетов по кейс-задачам, подготовка к выступлению с докладом, подготовка к промежуточной аттестации.

##### *Тема 2 Методы машинного обучения, основанные на статистических методах. Регрессии.*

Содержание темы: Понятие регрессии. Линейная регрессия. Логистическая регрессия. Обобщающая способность машинного обучения. Кросс-валидация.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчетов по кейс-задачам, подготовка к выступлению с докладом, подготовка к промежуточной аттестации.

##### *Тема 3 Машинное обучение с учителем: классификация.*

Содержание темы: Понятие классификации. Ошибки классификатора. Логические методы классификации. Метод градиентного бустинга. Многоклассовая классификация.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчетов по кейс-задачам, подготовка к выступлению с докладом, подготовка к промежуточной аттестации.

##### *Тема 4 Нейронные сети. Глубокое обучение.*

Содержание темы: Однослойные нейронные сети. Модель нейрона. Архитектуры нейронных сетей. Основы глубокого обучения. Многослойные нейронные сети. Свёртка. Свёрточные нейронные сети. Рекуррентные нейронные сети.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчетов по кейс-задачам, подготовка к выступлению с докладом, подготовка к промежуточной аттестации.

##### *Тема 5 Решение прикладных задач методами машинного обучения.*

Содержание темы: Кластеризация, ансамблевые модели, прогнозирование в прикладных задачах.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, практическое занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчетов по кейс-задачам, подготовка к выступлению с докладом, подготовка к промежуточной аттестации.

## 5 Методические указания для обучающихся по изучению и реализации дисциплины (модуля)

### 5.1 Методические рекомендации обучающимся по изучению дисциплины и по обеспечению самостоятельной работы

Во время самостоятельной проработки лекционного материала особое внимание следует уделять возникшим вопросам, непонятым терминам, спорным точкам зрения. В случае необходимости обращаться к преподавателю за консультацией. Студент должен четко уяснить, что именно с лекции начинается его подготовка к практическому занятию. Вместе с тем, лекция лишь организует мыслительную деятельность, но не обеспечивает глубину усвоения программного материала.

При подготовке к практическому занятию особое внимание необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы студент должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале. В процессе подготовки к практическому занятию рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретает практика в изложении и разъяснении полученных знаний.

При выполнении заданий необходимо использовать теоретический материал, делать ссылки на соответствующие теоремы, свойства, формулы и др. Решение выполняется подробно и содержит необходимые пояснительные ссылки.

Самостоятельная работа также включает работу на практических занятиях, во время применения «Метода кооперативного обучения» студенты работают в малых группах (3 – 4 чел.) над заданиями, в процессе выполнения которых они могут совещаться друг с другом, а также обращаться за помощью к преподавателю.

При необходимости следует обращаться за консультацией к преподавателю. Перед консультацией, необходимо хорошо продумать вопросы, которые требуют разъяснения.

#### **Общие требования и критерии оценки практических работ:**

**Требования к оборудованию:** дистрибутив языка программирования Python пакет Anaconda.

**Рекомендации по выполнению:** при выполнении работы следуйте правилам, о которых Вам рассказал преподаватель входе практической работы.

#### **Критерии оценки:**

Балл	Критерии оценки
0	Работа не выполнена.
5	Работа выполнена полностью. Студент не владеет теоретическим и практическим материалом, допуская грубые ошибки, испытывает затруднения в формулировке собственных суждений, неспособен ответить на дополнительные вопросы.
8	Работа выполнена полностью. Студент владеет теоретическим и практическим материалом на минимально допустимом уровне, отсутствуют ошибки при описании теории, испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допуская незначительные ошибки на дополнительные вопросы.
10	Работа выполнена полностью. Студент владеет теоретическим и практическим материалом, отсутствуют ошибки, формулирует собственные, самостоятельные, обоснованные, аргументированные суждения, представляет полные и развернутые ответы на дополнительные вопросы.

### 1. Практическая работа №1 «Машинное обучение. Решающие деревья»

**Цель:** Использование решающих деревьев для практических задач.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** решающее дерево, интерпретируемость решающего дерева, антифрод-система.

## **2. Практическая работа №2 «Машинное обучение. Линейные модели»**

**Цель:** Изучение линейной регрессии.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** задача регрессии, реализация на языке программирования Python линейной регрессии.

## **3. Практическая работа №3 «Машинное обучение. Логистическая регрессия»**

**Цель:** Изучение логистической регрессии.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** реализация на языке программирования Python логистической регрессии.

## **4. Практическая работа №4 «Задача регрессии»**

**Цель:** Изучение задачи регрессии.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** области применения задач регрессии, переобучение, Обобщающая способность машинного обучения, кросс-валидация, ансамбль моделей машинного обучения, персонализированные предложения на основе транзакций.

## **5. Практическая работа №5 «Машинное обучение. Логические методы классификации»**

**Цель:** Изучение логистических методов классификации.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** решающее дерево, случайный лес (Random Forest), методы градиентного бустинга, реализация на языке программирования Python логистических методов классификации.

## **6. Практическая работа №6 «Построение метода решения методом градиентного бустинга»**

**Цель:** Использование машинного обучения для решения задачи из области анализа данных.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** изучение задачи из области анализа данных, метрика AUC ROC, ROC-кривая, метод градиентного бустинга, новые признаки, типы транзакций, MCC-коды, визуализация данных.

## **7. Практическая работа №7 «Однослойные нейронные сети»**

**Цель:** Изучение однослойных нейронных сетей.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** простейший пример однослойной нейронной сети, архитектуры нейронных сетей, реализация на языке программирования Python однослойной нейронной сети.

## **8. Практическая работа №8 «Основы глубокого обучения. Модель нейрона»**

**Цель:** Изучение основ глубокого обучения.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** Основы глубокого обучения, типы нейронных сетей, модель нейрона, реализация на языке программирования Python модели нейрона с пороговой функцией активации, реализация на языке программирования Python модели нейрона с логистической функцией активации.

**9. Практическая работа №9 «Многослойные нейронные сети»**

**Цель:** Изучение многослойных нейронных сетей.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** Знакомство с фреймворком PyTorch, реализация одного нейрона и нейронных сетей на PyTorch, реализация многослойной нейронной сети на PyTorch.

**10. Практическая работа №10 «Многоклассовая классификация»**

**Цель:** Изучение задачи классификации объектов по нескольким классам.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** Эффективное обучение нейронных сетей, обучение на больших выборках, модификации градиентного спуска, помогающие ускорить обучение нейронных сетей.

**11. Практическая работа №11 «Сверточные нейронные сети»**

**Цель:** Изучение сверточных нейронных сетей.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** Операция свертки, сверточный и пулинг слой, операция свертки и пулинга, сверточные нейронные сети на PyTorch, dataset MNIST, техника Transfer Learning, архитектуры сверточных нейронных сетей.

**12. Практическая работа №12 «Сверточные нейронные сети. Приложения»**

**Цель:** Изучение сверточных нейронных сетей.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** обучение сверточных нейронных сетей, приложения сверточных нейронных сетей: задача классификации, задача определения наличия объектов на снимке и задача сегментации изображений.

**13. Практическая работа №13 «Рекуррентные нейронные сети»**

**Цель:** Изучение рекуррентных нейронных сетей.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** Forward pass, backward pass, проблемы использования рекуррентных нейронных сетей, архитектура рекуррентного нейрона-LSTM, Gated Recurrent Unit, двунаправленные рекуррентные нейронные сети.

**14. Практическая работа №14 «Natural Language Processing»**

**Цель:** Автоматическая обработка естественного языка с использованием методов машинного обучения.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** Natural Language Processing, первичная обработка текста, извлечение фактов, история внедрения методов машинного обучения в Google Translate.

**15. Практическая работа №15 «Машинное обучение: тематическое моделирование»**

**Цель:** Тематическое моделирование с использованием методов машинного обучения.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** Применение методов машинного обучения: категоризация документов, классификация документов, аннотирование документов, поиск иерархической структуры.

**16. Практическая работа №16 «Машинное обучение: тематическое моделирование»**

**Цель:** Тематическое моделирование с использованием методов машинного обучения.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** Применение методов машинного обучения: выявление трендов в данных, поиск информации, анализ информационных потоков, анализ биоинформационных данных, рубрикация документов.

**17. Практическая работа №17 «Reinforcement Learning»**

**Цель:** Изучение метода Reinforcement Learning.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** знакомство с Reinforcement Learning и базовыми определениями, пример простейшей среды, OpenAi Gym, реализация на языке программирования Python Reinforcement Learning.

**18. Практическая работа №18 «Приложения методов машинного обучения»**

**Цель:** Изучение практических приложения использования методов машинного обучения.

**Планируемые результаты обучения** в соответствии с компетенциями, составляющие которых перечислены и описаны в РПД к данной дисциплине

**Содержание практической работы:** приложения использования методов машинного обучения: методы машинного обучения в автомобильной промышленности, автомобили с системами автопилота, робототехника, персональные умные ассистенты.

## **5.2 Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов**

При необходимости обучающимся из числа лиц с ограниченными возможностями здоровья и инвалидов (по заявлению обучающегося) предоставляется учебная информация в доступных формах с учетом их индивидуальных психофизических особенностей:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; индивидуальные задания, консультации и др.

## **6 Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю)**

В соответствии с требованиями ФГОС ВО для аттестации обучающихся на соответствие их персональных достижений планируемыми результатами обучения по

дисциплине (модулю) созданы фонды оценочных средств. Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 1.

## **7 Учебно-методическое и информационное обеспечение дисциплины (модуля)**

### **7.1 Основная литература**

1. Котельников, Е. В. Введение в машинное обучение и анализ данных : учебное пособие / Е. В. Котельников, А. В. Котельникова. — Киров : ВятГУ, 2023. — 68 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/390698> (дата обращения: 25.05.2026). — Режим доступа: для авториз. пользователей.

2. Машинное обучение с использованием Python : учебно-методическое пособие / составители А. В. Осин, К. А. Хализев. — Москва : МТУСИ, 2025. — 20 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/501209> (дата обращения: 25.05.2026). — Режим доступа: для авториз. пользователей.

3. Платонов, А. В. Машинное обучение : учебное пособие для вузов / А. В. Платонов. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 89 с. — (Высшее образование). — ISBN 978-5-534-20732-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/558662> (дата обращения: 01.09.2025).

### **7.2 Дополнительная литература**

1. Айвазян, С. А. Методы эконометрики : учебник / С. А. Айвазян. — Москва : Магистр : ИНФРА-М, 2022. — 512 с. - ISBN 978-5-9776-0153-5. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1840468> (дата обращения: 31.05.2026)

2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2025. — 343 с. — (Высшее образование). - ISBN 978-5-16-020255-6. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2166199> (дата обращения: 31.05.2026)

3. Плохотников, К. Э. Математика и анализ данных : учебное пособие / К.Э. Плохотников. — Москва : ИНФРА-М, 2025. — 1114 с. - ISBN 978-5-16-114323-0. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2226505> (дата обращения: 31.05.2026)

### **7.3 Ресурсы информационно-телекоммуникационной сети "Интернет", включая профессиональные базы данных и информационно-справочные системы (при необходимости):**

1. Образовательная платформа "ЮРАЙТ"  
2. Электронно-библиотечная система "ZNANIUM.COM"  
3. Электронно-библиотечная система "ЛАНЬ"  
4. Open Academic Journals Index (ОАИ). Профессиональная база данных - Режим доступа: <http://oaji.net/>

5. Президентская библиотека им. Б.Н.Ельцина (база данных различных профессиональных областей) - Режим доступа: <https://www.prlib.ru/>

6. Информационно-справочная система "Консультант Плюс" - Режим доступа: <http://www.consultant.ru/>

**8 Материально-техническое обеспечение дисциплины (модуля) и перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения**

Основное оборудование:

- Компьютеры
- Проектор
- Мультимедийный комплект №2 в составе:проектор Casio XJ-M146,экран 180\*180,крепление потолочное
- Персональный компьютер №3 "В-tronix professional 3872\2015"
- Телевизионнаяпанель телевизор Samsung UE40H5003

Программное обеспечение:

- Adobe Reader 10 Russian
- Microsoft Office 2010 Standard Russian
- Microsoft Windows Professional 7 Russian
- Python
- КонсультантПлюс

МИНОБРНАУКИ РОССИИ  
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
НАУЧНО-ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР "ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ"

Фонд оценочных средств  
для проведения текущего контроля  
и промежуточной аттестации по дисциплине (модулю)

**МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ**

Направление и направленность (профиль)  
09.03.04 Программная инженерия. Программная инженерия

Год набора на ОПОП  
2026

Форма обучения  
очная

Владивосток 2026

## 1 Перечень формируемых компетенций

Название ОПОП ВО, сокращенное	Код и формулировка компетенции и	Код и формулировка индикатора достижения компетенции
09.03.04 «Программная инженерия» (Б-ИН)	ОПК-6 : Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1к : Использует навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач
		ОПК-6.2к : Применяет современные программные среды программирования, разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ

Компетенция считается сформированной на данном этапе в случае, если полученные результаты обучения по дисциплине оценены положительно (диапазон критериев оценивания результатов обучения «зачтено», «удовлетворительно», «хорошо», «отлично»). В случае отсутствия положительной оценки компетенция на данном этапе считается несформированной.

## 2 Показатели оценивания планируемых результатов обучения

**Компетенция ОПК-6 «Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов»**

Таблица 2.1 – Критерии оценки индикаторов достижения компетенции

Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине			Критерии оценивания результатов обучения
	Код	Тип	Результат	
ОПК-6.1к : Использует навыки и программирования, отладки и тестирования прототипов программно-технических комплексов задач	РД 7	Знание	способов сборки программных модулей и компонент алгоритмов машинного обучения в программный продукт	сформировавшееся систематическое знание способов сборки и программных модулей и компонент алгоритмов машинного обучения в программный продукт
	РД 8	Умение	собирать программные модули и компоненты машинного обучения в единый программный продукт, осуществлять отладку и тестирование программного продукта	сформировавшееся систематическое умение собирать программные модули и компоненты машинного обучения в единый программный продукт, осуществлять отладку и тестирование программного продукта
	РД 9	Навык	программирования программных модулей на современных языках программирования машинного обучения, сборки программных модулей и компонент в программный продукт,	сформировавшееся систематическое владение навыками программирования программных модулей на современных языках программирования машинного обучения, сборки программных модулей и компонент

			осуществлять отладку собранного программного продукта	нт в программный продукт, осуществлять отладку собранного программного продукта
	РД 10	Знание	требований к разработке кода информационных систем и баз данных информационных систем на основе принципов машинного обучения	сформировавшееся систематическое знание требований к разработке кода информационных систем и баз данных информационных систем на основе принципов машинного обучения
	РД 11	Умение	разрабатывать информационные системы и базы данных информационных систем для методов машинного обучения	сформировавшееся систематическое умение разрабатывать информационные системы и базы данных информационных систем для методов машинного обучения
	РД 12	Навык	разработки прототипа информационной системы и ее базы данных с применением методов машинного обучения	сформировавшееся систематическое владение навыками разработки прототипа информационной системы и ее базы данных с применением методов машинного обучения
ОПК-6.2к : Применяет современные программные среды программирования, разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ	РД 1	Знание	методов, моделей, алгоритмов, технологий и инструментальных средств машинного обучения	сформировавшееся систематическое знание методов, моделей, алгоритмов, технологий и инструментальных средств машинного обучения
	РД 2	Умение	использовать методы, модели, алгоритмы, технологии и инструментальные средства машинного обучения для работы с большими данными	использовать методы, модели, алгоритмы, технологии и инструментальные средства машинного обучения для работы с большими данными
	РД 3	Навык	программирования методов, моделей, алгоритмов машинного обучения с использованием современных технологий и инструментальных средств	сформировавшееся систематическое владение навыками программирования методов, моделей, алгоритмов машинного обучения с использованием современных технологий и инструментальных средств
	РД 4	Знание	требований к проведению аналитических исследований для сбора, обработки и анализа больших данных методами машинного обучения	сформировавшееся систематическое знание требований к проведению аналитических исследований для сбора, обработки и анализа больших данных методами машинного обучения
	РД 5	Умение	проводить аналитические исследования по сбору, обработке и анализу больших данных методами машинного обучения	сформировавшееся систематическое умение проводить аналитические исследования по сбору, обработке и анализу больших данных методами машинного обучения
	РД 6	Навык	проведения аналитических исследований для решения практических задач методами машинного обучения	сформировавшееся систематическое владение навыками проведения аналитических исследований для решения практических задач методами машинного обучения

Таблица заполняется в соответствии с разделом 1 Рабочей программы дисциплины (модуля).

### 3 Перечень оценочных средств

Таблица 3 – Перечень оценочных средств по дисциплине (модулю)

Контролируемые планируемые результаты обучения	Контролируемые темы дисциплины	Наименование оценочного средства и представление его в ФОС		
		Текущий контроль	Промежуточная аттестация	
Очная форма обучения				
РД1	Знание : методов, моделей, алгоритмов, технологий и инструментальных средств машинного обучения	1.4. Нейронные сети. Глубокое обучение	Практическая работа	Кейс-задача
		1.5. Решение прикладных задач методами машинного обучения	Практическая работа	Кейс-задача
РД2	Умение : использовать методы, модели, алгоритмы, технологии и инструментальные средства машинного обучения для работы с большими данными	1.1. Основные понятия машинного обучения	Практическая работа	Кейс-задача
		1.2. Методы машинного обучения, основанные на статистических методах. Регрессии	Практическая работа	Кейс-задача
		1.3. Машинное обучение с учителем: классификация	Практическая работа	Кейс-задача
РД3	Навык : программирования методов, моделей, алгоритмов машинного обучения с использованием современных технологий и инструментальных средств	1.1. Основные понятия машинного обучения	Практическая работа	Кейс-задача
		1.2. Методы машинного обучения, основанные на статистических методах. Регрессии	Практическая работа	Кейс-задача
		1.3. Машинное обучение с учителем: классификация	Практическая работа	Кейс-задача
		1.4. Нейронные сети. Глубокое обучение	Практическая работа	Кейс-задача
		1.5. Решение прикладных задач методами машинного обучения	Практическая работа	Кейс-задача

### 4 Описание процедуры оценивания

Качество сформированности компетенций на данном этапе оценивается по результатам текущих и промежуточных аттестаций при помощи количественной оценки, выраженной в баллах. Максимальная сумма баллов по дисциплине (модулю) равна 100 баллам.

Вид учебной деятельности	Оценочное средство		
	Практические задания	кейс-задача	Итого
Лекции	10		10
Практические занятия	60		60
Промежуточная аттестация		30	30
Итого	70	30	100

Сумма баллов, набранных студентом по всем видам учебной деятельности в рамках дисциплины, переводится в оценку в соответствии с таблицей.

Сумма баллов по дисциплине	Оценка по промежуточной аттестации	Характеристика качества сформированности компетенции
от 91 до 100	«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций, обладает всестороннее, систематическое и глубокое знание учебного материала, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой, умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными знаниями, умениями, применяет их в ситуациях повышенной сложности.
от 76 до 90	«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
от 61 до 75	«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных знаний, умений, навыков по некоторым дисциплинарным компетенциям, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
от 41 до 60	«не зачтено» / «неудовлетворительно»	У студента не сформированы дисциплинарные компетенции, проявляется недостаточность знаний, умений, навыков.
от 0 до 40	«не зачтено» / «неудовлетворительно»	Дисциплинарные компетенции не сформированы. Проявляется полное или практически полное отсутствие знаний, умений, навыков.

## 5 Примерные оценочные средства

### 5.1 Примеры заданий для выполнения практических работ

#### Примерный перечень практических заданий:

1. Методы МО с учителем: линейные регрессии;
2. Методы МО с учителем, классификация: логистические регрессии (бинарная, множественная)
3. Методы МО с учителем, классификация: деревья решений
4. Методы МО без учителя, кластеризация: EM кластеризация, иерархическая кластеризация
5. Нейронные сети: перцептрон. Реализация регрессий с помощью нейронных сетей
6. Ансамблевые метод и модели

#### Краткие методические указания

На выполнение одного практического задания отводится не менее двух академических часов

После выполнения каждого практического задания студент должен представить работоспособный скрипт (программу) с комментариями, результаты ее исполнения (в данном случае так называемый ноутбук python)

При необходимости, студент должен внести исправления в скрипт, уточнить выводы

#### Шкала оценки

№	Баллы	Описание
5	9-10	Студент демонстрирует умения на итоговом уровне: умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
4	5-6	Студент демонстрирует умения на среднем уровне: освоил основные умения, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.

3	3-4	Студент демонстрирует умения и навыки на базовом уровне: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных умений, навыков по дисциплинарной компетенции, испытываются значительные затруднения при оперировании умениями и при их переносе на новые ситуации.
2	1-2	Студент демонстрирует умения и навыки на уровне ниже базового: проявляется недостаточность умений и навыков.
1	0	Студентом проявляется полное или практически полное отсутствие умений и навыков.

## 5.2 Задания для решения кейс-задачи

### Пример кейс-задания для итогового контроля сформированности знаний, умений навыков

Задание:

Представлены Экологические данные - степень загрязненности российских рек и бассейнов российских рек.

Используйте изученные методы машинного обучения, ответьте на вопросы.

1. Можно ли утверждать, что объем загрязненных сточных вод определяется:

Экономическими показателями (ВРП, Расходами на охрану окружающей среды), показателями выбросов, объемом улавливания загрязняющих веществ. Какие из указанных показателей имеют более существенное влияние и как это определить?

Выбрать метод МО, позволяющий ответить на данный вопрос, оценить модель, обосновать ответ используя статистические показатели.

2. Можно ли утверждать, по степени объема загрязняющих выбросов регионы в европейской части России и в азиатской части существенно различаются? Выбрать метод МО, позволяющий ответить на данный вопрос, оценить, обосновать ответ. Можно сделать вывод не прибегая к МО?

Результат представить в форме развернутого пояснения, обозначениями и выводами (строка комментариев в коде)

*Краткие методические указания*

**Порядок выполнения задания:**

1. загрузить исходный датафрейм;
2. провести "гуманитарный анализ" задачи (при необходимости экономической, социальной и пр. смысл поставленной задачи)
3. выбрать метод МО, который предположительно может проверить гипотезу и помочь ответить на поставленные вопросы
4. создать программный код, проверить гипотезу, сделать выводы
5. если ответ не получен, выбрать другой метод МО

*Шкала оценки*

№	Баллы	Описание
5	41-50	Студент демонстрирует умения на итоговом уровне: способен свободно выполнить задание, свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
4	31-40	Студент демонстрирует умения на среднем уровне: освоил основные умения и навыки, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.
3	16-30	Студент демонстрирует умения и навыки на базовом уровне: допускаются значительные ошибки, проявляется отсутствие отдельных умений, навыков по дисциплинарной компетенции, испытываются значительные затруднения при оперировании умениями и при их переносе на новые ситуации.
2	10-15	Студент демонстрирует умения и навыки на уровне ниже базового: проявляется недостаточность умений и навыков.
1	0-9	Студентом проявляется полное или практически полное отсутствие умений и навыков, но присутствовал на занятиях

## Ключи для ФОС «Методы машинного обучения»

### 1. Практическая работа №1: Машинное обучение, методы и модели, наборы данных, загрузка данных

Наборы данных по вариантам: <https://disk.yandex.ru/i/nY93FrjzthRO2Q>

```
# импортируем библиотеки
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
!pip install seaborn

PATH_TO_FILE: str = '/content/employee.csv' # Загрузка набора данных
## p.s. в левой панели нажать на папку и перетащить туда файл,
## а wince.csv заменить на свой вариант (ПКМ -> копировать путь)

df = pd.read_csv(PATH_TO_FILE) # функция превращение данных из CSV в
DataFrame
PATH_TO_FILE: str = '/content/employee.csv' # Загрузка набора данных
## p.s. в левой панели нажать на папку и перетащить туда файл,
## а wince.csv заменить на свой вариант (ПКМ -> копировать путь)

df = pd.read_csv(PATH_TO_FILE) # функция превращение данных из CSV в
DataFrame
# Вывод первых 12 строк
df.head(12)
satisfaction_level      last_evaluation number_project average_monthly_hours time_spend_company
      Work_accident promotion_last_5years department      salary left
0      0.38      0.53      2      157      3      0      0      sales      low      1
1      0.80      0.86      5      262      6      0      0      sales      medium      1
2      0.11      0.88      7      272      4      0      0      sales      medium      1
3      0.72      0.87      5      223      5      0      0      sales      low

# Информация о наборе данных
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 10 columns):
#   Column                Non-Null Count  Dtype
---  -
0   satisfaction_level     14999 non-null  float64
1   last_evaluation       14999 non-null  float64
2   number_project        14999 non-null  int64
3   average_monthly_hours 14999 non-null  int64
4   time_spend_company    14999 non-null  int64
5   Work_accident         14999 non-null  int64
6   promotion_last_5years 14999 non-null  int64
7   department            14999 non-null  object
8   salary                14999 non-null  object
```

```
9 left 14999 non-null int64
dtypes: float64(2), int64(6), object(2)
memory usage: 1.1+ MB
```

## 2. Практическая работа №2: Методы машинного обучения – регрессии.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score

# Загрузка набора данных
try:
    df = pd.read_csv('/content/income.csv')
except FileNotFoundError:
    df = pd.read_csv('income.csv')

df['is_work'] = np.where(df['hours-per-week'] >= 35, 1, 0)

y = df['is_work'] # выбираем целевую переменную (категориальную)
X = df.drop('is_work', axis=1) # переменные для проверки влияния

# Создаем копию данных
category_columns: list[str] = X.select_dtypes(include=['object']).columns #
собираем колонки помеченные как object

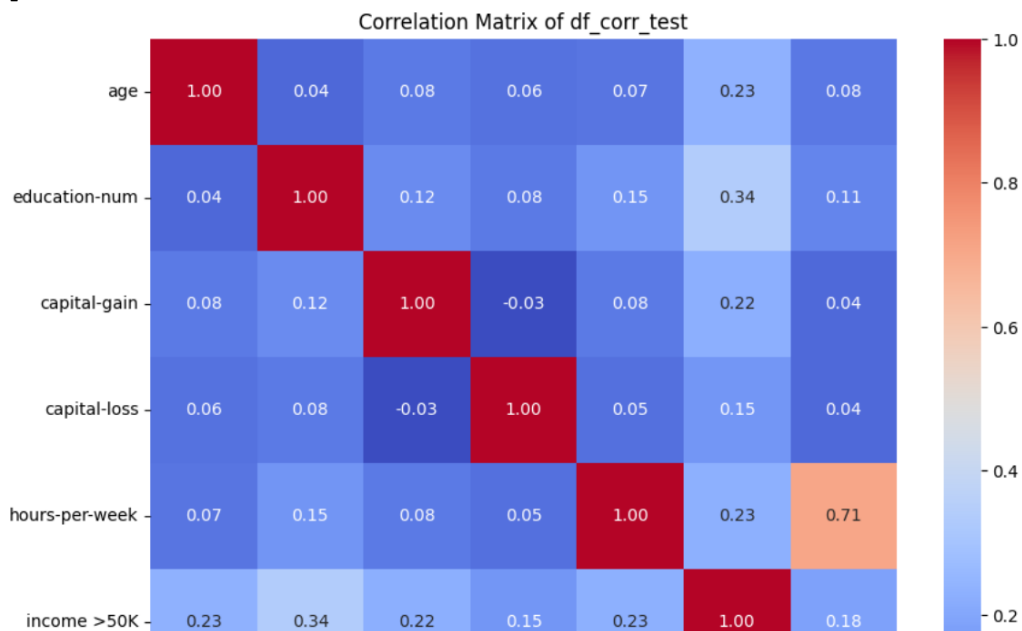
# Применяем One-Hot Encoding к категориальным столбцам, исключая первую
категорию (drop_first=True)
X = pd.get_dummies(X, columns=category_columns, drop_first=True)
df_corr_test = X.copy()

# Добавляем целевую переменную в копию данных
df_corr_test['y'] = y
df_corr_test

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# Выводим матрицу корреляции (только для числовых столбцов)
correlation_matrix = df_corr_test.select_dtypes(['int', 'float']).corr()

# Отображаем матрицу корреляции с помощью тепловой карты
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title("Correlation Matrix of df_corr_test")
```

```
plt.show()
```



```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
```

```
def resolve_regression(X, y, regularization=None):
    """
    Выполняет регрессию с учетом регуляризации.

    Parameters:
    - X: Входные данные (признаки).
    - y: Целевые значения.
    - regularization: Тип регуляризации ('l1', 'l2', None). Если None,
используется обычная линейная регрессия.
    """
    # Разделение данных на обучающую и тестовую выборки (80% / 20%)
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

    # Инициализация модели в зависимости от типа регуляризации
    if regularization == 'l1':
        model = Lasso() # L1-регуляризация (Lasso)
    elif regularization == 'l2':
        model = Ridge() # L2-регуляризация (Ridge)
    else:
        model = LinearRegression() # Линейная регрессия без регуляризации

    # Обучение модели
    model.fit(X_train, y_train)

    # Предсказание на тестовой выборке
    y_pred = model.predict(X_test)
```

```

# Вычисление метрик качества
mae = mean_absolute_error(y_test, y_pred) # Средняя абсолютная ошибка
mse = mean_squared_error(y_test, y_pred) # Средняя квадратичная ошибка
r2 = r2_score(y_test, y_pred)           # Коэффициент детерминации
(R^2)

# Вывод результатов
print(f"Regularization: {regularization}")
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"R^2 Score: {r2:.2f}")

return model, X_train, X_test, y_train, y_test
Regularization: None
Mean Absolute Error (MAE): 0.20
Mean Squared Error (MSE): 0.07
R^2 Score: 0.54

```

### 3. Практическая работа №3: Методы машинной классификации

Загрузка библиотек и предобработка данных

```

!pip install pandas scikit-learn seaborn scipy
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score

# Загрузка набора данных
df = pd.read_csv('/content/income.csv')
df['is_work'] = np.where(df['hours-per-week'] >= 35, 1, 0)
df.is_work = df.is_work.astype('category')
df['education-num'] = df['education-num'].astype('category')

y = df['is_work'] # выбираем целевую переменную (категориальную)
X = df.drop('is_work', axis=1) # переменные для проверки влияния

X = X.drop('hours-per-week', axis=1)
df.drop('hours-per-week', axis=1).info()

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---

```

```
0 age 32561 non-null int64
1 workclass 30725 non-null object
2 education 32561 non-null object
3 education-num 32561 non-null category
4 marital-status 32561 non-null object
5 occupation 30718 non-null object
6 relationship 32561 non-null object
7 race 32561 non-null object
8 sex 32561 non-null object
9 capital-gain 32561 non-null int64
10 capital-loss 32561 non-null int64
11 native-country 31978 non-null object
12 income >50K 32561 non-null int64
13 is_work 32561 non-null category
```

```
dtypes: category(2), int64(4), object(8)
```

```
memory usage: 3.0+ MB
```

Обучаем логистическую регрессию (без доп. манипуляций) на дисбалансе

```
# from sklearn.model_selection import train_test_split
# from sklearn.linear_model import LogisticRegression

# # Разделяем данные на обучающую и тестовую выборки
# X_train, X_test, y_train, y_test = train_test_split(X_disbalanced,
y_disbalanced, test_size=0.3, random_state=42)

# # Обучаем модель
# model = LogisticRegression(penalty=None, solver='saga', max_iter=10000)
# model.fit(X_train, y_train)

# # Предсказания
# y_pred = model.predict(X_test)

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Применяем One-Hot Encoding к категориальным переменным
X_disbalanced_encoded = pd.get_dummies(X_disbalanced, drop_first=True)

# Разделяем данные на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X_disbalanced_encoded,
y_disbalanced, test_size=0.3, random_state=42)

# Обучаем модель с использованием L2 регуляризации
model = LogisticRegression(penalty='l2', solver='saga', max_iter=10000)
model.fit(X_train, y_train)

# Вывод результатов
print("Predictions:", y_pred)
from sklearn.metrics import classification_report, accuracy_score

print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Accuracy: 0.9066739012479653

Classification Report:

	precision	recall	f1-score	support
0	0.91	1.00	0.95	1671
1	0.00	0.00	0.00	172
accuracy			0.91	1843
macro avg	0.45	0.50	0.48	1843
weighted avg	0.82	0.91	0.86	1843

Мы видим что модель в классе 0 смогла определить правильно на 91% в дисбалансном режиме, а класс 1 вообще не смогла определить. Мы видим, что все наблюдения которые были даны класса 0 было правильно определены, а для класса 1 не смогла обнаружить какие либо наблюдения у класса 1

Мы делаем вывод, что в среднем модель показывает плохую производительность по следующим параметрам:

Precision (Точность): 0.45

Recall (Вспомнить): 0.50

F1-score (F1-счет): 0.48

Алгоритм дерева принятия решений (Decision Tree) — это один из методов машинного обучения, который используется для решения задач классификации и регрессии. Дерево решений представляет собой структуру, в которой каждый внутренний узел соответствует некоторому признаку, а каждая ветвь — определенному значению этого признака. Листовые узлы представляют собой классы или значения, которые необходимо предсказать

```
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

# Предполагаем, что X и y уже определены
# Применяем One-Hot Encoding к категориальным переменным
category_columns = X.select_dtypes(include=['object']).columns # выбираем категориальные колонки
X_encoded = pd.get_dummies(X, columns=category_columns, drop_first=True)

# Разделяем данные на обучающую и тестовую выборки
X_train_multi, X_test_multi, y_train_multi, y_test_multi =
train_test_split(X_encoded, y, test_size=0.3, random_state=42)

# Обучаем модель
tree_model = DecisionTreeClassifier(random_state=42, max_depth=10)
tree_model.fit(X_train_multi, y_train_multi)

# Предсказания
y_pred_multi = tree_model.predict(X_test_multi)
```

```
# Получение названий классов
class_names = np.unique(y) # Получаем уникальные значения целевой
переменной
```

```
# Преобразуем названия классов в строковый формат
```

```
class_names = [str(name) for name in class_names]
```

```
Classification Report:
```

	precision	recall	f1-score	support
0	0.88	0.94	0.91	7455
1	0.75	0.58	0.65	2314
<b>accuracy</b>			0.85	9769
<b>macro avg</b>	0.81	0.76	0.78	9769
<b>weighted avg</b>	0.85	0.85	0.85	9769

Точность для классов составляет:

0: 88% точности

1: 75% точности

Проанализированно точно данные для каждого классов составляет:

0: 94% точности

1: 58% точности

Тут мы видим, что было плохо проанализированно для 1 класса, а для 0 класса практически точно

По

F1-Score

Высокий F1-Score для класса 0 (0.91) показывает хорошую производительность, но для класса 1 (0.65) он ниже, что указывает на необходимость улучшения модели для этого класса.

Модель хорошо работает для 0 класса, а для 1 класса есть определенные проблемы Это произошло из-за того что данные не сбалансированы и нужно сбалансировать их.

#### 4. Обучение без учителя

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score

# Загрузка набора данных
df = pd.read_csv('/content/income.csv')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

**K-means** — это алгоритм кластеризации, который делит данные на  $k$  групп (кластеров). Основная идея: найти центры кластеров так, чтобы точки внутри одного кластера были ближе друг к другу, чем к точкам из других кластеров.

1. Инициализация: выбираем  $k$  случайных центров (центроидов).
2. Присвоение точек: каждая точка относится к ближайшему центру. Обновление центров: пересчитываем центры кластеров как среднее всех точек в кластере.
3. Повторяем шаги 2–3, пока центры не перестанут меняться или не достигнем максимального числа итераций.

---

```
[ ]
! pip install ipywidgets

import ipywidgets as widgets
from IPython.display import display, clear_output

def interactive_kmeans_visualization(X, max_clusters=10, max_iterations=50):
    # Преобразуем X в numpy array, если это DataFrame
    if isinstance(X, pd.DataFrame):
        X = X.values

    # Проверка на наличие NaN или inf
    if np.any(np.isnan(X)) or np.any(np.isinf(X)):
        print("Данные содержат NaN или inf. Пожалуйста, очистите данные.")
        return

    def plot_kmeans_iteration(X, centroids, labels=None, step=0):
        clear_output(wait=True)
        plt.figure(figsize=(8, 6))
        if labels is not None:
            plt.scatter(X[:, 0], X[:, 1], c=labels, cmap='viridis', s=30,
alpha=0.6)
        else:
            plt.scatter(X[:, 0], X[:, 1], s=30, alpha=0.6)
            plt.scatter(centroids[:, 0], centroids[:, 1], c='red', marker='x',
s=100, label='Centroids')
        plt.title(f'K-means Iteration {step}')
        plt.legend()
        plt.show()

    def k_means_interactive(X, k, max_iters):
        np.random.seed(42)
        centroids = X[np.random.choice(X.shape[0], k, replace=False)]

        for step in range(max_iters):
            # Вычисление расстояний
            distances = np.linalg.norm(X[:, np.newaxis] - centroids, axis=2)
            # Присвоение меток
            labels = np.argmin(distances, axis=1)
            # Визуализация
```

```

plot_kmeans_iteration(X, centroids, labels, step)
# Обновление центроидов
new_centroids = np.array([X[labels == i].mean(axis=0) for i in
range(k)])
# Проверка на сходимость
if np.all(centroids == new_centroids):
    break
centroids = new_centroids

def run_kmeans(k, max_iters):
    k_means_interactive(X, k, max_iters)

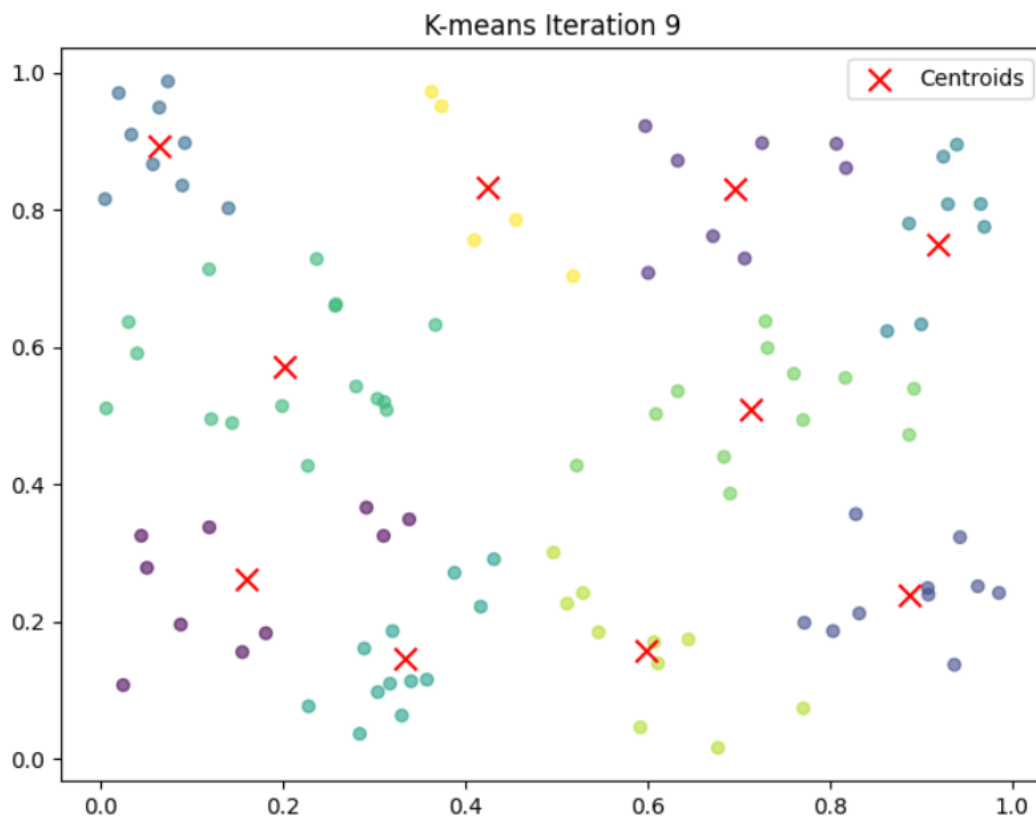
k_slider = widgets.IntSlider(value=3, min=2, max=max_clusters, step=1,
description='Clusters (k):')
iter_slider = widgets.IntSlider(value=10, min=1, max=max_iterations,
step=1, description='Max Iterations:')

interactive_plot = widgets.interactive(run_kmeans, k=k_slider,
max_iters=iter_slider)
display(interactive_plot)
# Для примера, используем случайные данные
np.random.seed(42)
df = pd.DataFrame(np.random.rand(100, 2), columns=['Feature1', 'Feature2'])

# Вызываем функцию для интерактивной визуализации K-means
interactive_kmeans_visualization(df)

```

 Clusters (k):  10  
 Max Iterati...  10



## Кейс-задача

Представлены Экологические данные - степень загрязненности российских рек и бассейнов российских рек.

Используйте изученные методы машинного обучения, ответьте на вопросы.

1. Можно ли утверждать, что объем загрязненных сточных вод определяется:

Экономическими показателями (ВРП, Расходами на охрану окружающей среды), показателями выбросов, объемом улавливания загрязняющих веществ. Какие из указанных показателей имеют более существенное влияние и как это определить?

Выбрать метод МО, позволяющий ответить на данный вопрос, оценить модель, обосновать ответ используя статистические показатели.

2. Можно ли утверждать, по степени объема загрязняющих выбросов регионы в европейской части России и в азиатской части существенно различаются? Выбрать метод МО, позволяющий ответить на данный вопрос, оценить, обосновать ответ.

Можно сделать вывод не прибегая к МО?

Результат представить в форме развернутого пояснения, обозначениями и выводами (строка комментов в коде)

Код на python для анализа продемонстрировать преподавателю.

### «Ключи» - результат выполнения задания

Вопрос 1: Влияние экономических и экологических показателей на объем загрязненных сточных вод

#### Метод множественной линейной регрессии (ОЛС)

На основании результатов модели множественной линейной регрессии можно сделать следующие выводы о влиянии экономических и экологических показателей на объем загрязненных сточных вод.

#### Статистические показатели

**R-squared = 0.580:** Это означает, что примерно 58% вариации в объеме сброса загрязненных сточных вод объясняется выбранными независимыми переменными. Это неплохой показатель, но также предполагает, что другие факторы могут влиять на объем сброса.

**F-statistic и Prob (F-statistic):** Значение F-statistic равно 117.5, а p-значение составляет  $6.92e-786.92e-786.92e-78$ . Это говорит о том, что модель в целом является значимой и можно отвергнуть нулевую гипотезу о том, что все коэффициенты равны нулю.

#### Влияние независимых переменных:

##### ВРП (млн руб):

Коэффициент  $2.436e-05$ , p-значение 0.014

Это означает, что с увеличением ВРП на 1 млн руб объем сточных вод увеличивается на  $2.436e-05$  млн м<sup>3</sup>. Поскольку p-значение  $< 0.05$ , влияние ВРП является статистически значимым.

##### Расходы на охрану окружающей среды (млн руб):

Коэффициент 0.0080, p-значение 0.000

Статистически значимое влияние на объем сточных вод. Увеличение расходов на 1 млн руб связано с увеличением сброса на 0.0080 млн м<sup>3</sup>.

##### Выбросы от автомобильного транспорта (тыс. тонн):

Коэффициент 0.1715, р-значение 0.000

Значимое влияние на объем сброса. Увеличение на 1 тыс. тонн связано с увеличением сточных вод на 0.1715 млн м<sup>3</sup>.

#### **Выбросы от стационарных источников (тыс. тонн):**

Коэффициент -0.0118, р-значение 0.513

Не имеет статистически значимого влияния на объем сточных вод, р-значение > 0.05.

#### **Улавливание загрязняющих веществ (тыс. тонн):**

Коэффициент 0.0192, р-значение 0.002

Значительное положительное влияние на объем сточных вод.

#### **Вывод**

На основании анализа можно утверждать, что объем загрязненных сточных вод в основном определяется экономическими показателями (ВРП, Расходы на охрану окружающей среды) и выбросами от автомобильного транспорта. Показатели выбросов от стационарных источников не оказали значимого влияния.

Вопрос 2: Сравнение загрязняющих выбросов между регионами

#### **Метод для анализа различий**

Для сравнения объемов выбросов в европейской и азиатской частях России также можно использовать t-тест для независимых выборок. Это позволит выяснить, существует ли статистически значимая разница в средних выбросах двух групп.

#### **Оценка с использованием t-теста**

На основании данных:

**t-статистика:** 0.4186

**р-значение:** 0.6758

#### **Интерпретация**

**р-значение > 0.05** указывает на то, что не существует статистически значимых различий в объеме загрязняющих выбросов между европейской и азиатской частями России.

**t-статистика** близка к нулю, что дополнительно поддерживает это утверждение.

#### **Выводы без использования МО**

Различия в выбросах между двумя регионами являются незначительными.

#### **Модели:**

Множественная линейная регрессия показала значительное влияние ВРП и расходов на охрану окружающей среды на объем загрязненных сточных вод.

Выбросы от стационарных источников не оказали значительного влияния.

#### **Сравнение регионов:**

t-тест показал отсутствие статистически значимых различий в загрязняющих выбросах между европейской и азиатской частями России.

#### **Код:**

```
import openpyxl
import pandas as pd
import statsmodels.api as sm

# Открываем Excel файл
book = openpyxl.load_workbook("Экология.xlsx", data_only=True)
```

```

# Выбираем активный лист
sheet = book.active

# Читаем данные в DataFrame
data = pd.DataFrame(sheet.values)

# Заменяем первую строку на заголовки
data.columns = data.iloc[0]
data = data[1:]

# Преобразуем колонки в нужные типы (например, float для числовых значений)
numeric_columns = ['ВВП, млн. руб', 'Расходы на охрану окружающей среды,
млн. руб',
                    'Выбросы от автомобильного транспорта, тыс. тонн',
                    'Выбросы от стационарных источников, тыс. тонн',
                    'Улавливание загрязняющих веществ, тыс. тонн',
                    'Сброс загрязненных сточных вод недостаточно очищенных,
млн м3']

for col in numeric_columns:
    data[col] = pd.to_numeric(data[col], errors='coerce')

# Определяем зависимую и независимые переменные
X = data[numeric_columns[:-1]] # Все, кроме последней колонки (сброс
сточных вод)
y = data['Сброс загрязненных сточных вод недостаточно очищенных, млн м3']

# Добавляем константу
X = sm.add_constant(X)

# Построение модели множественной регрессии
model = sm.OLS(y, X).fit()

# Получаем результаты
print(model.summary()) # Выводим сводку модели

# Создаем два отдельных набора данных
europa = data[data['Округ'].isin(['Центральный федеральный округ',
                                'Северо-Западный федеральный округ',
                                'Приволжский федеральный округ',
                                'Южный федеральный округ',
                                'Уральский федеральный округ'])]
azia = data[data['Округ'].isin(['Сибирский федеральный округ',
                                'Дальневосточный федеральный округ'])]

# Извлекаем выбросы от стационарных источников, удаляя NaN
europa_emissions = europa['Выбросы от стационарных источников, тыс.
тонн'].dropna()
azia_emissions = azia['Выбросы от стационарных источников, тыс.
тонн'].dropna()

from scipy import stats

# Выполним t-тест
t_stat, p_value = stats.ttest_ind(europa_emissions, azia_emissions)

print(f't-статистика: {t_stat}, p-значение: {p_value}')

```