

**Федеральное государственное образовательное бюджетное  
учреждение высшего образования  
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ  
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»  
(Финансовый университет)**

**Департамент анализа данных и машинного обучения  
Факультета информационных технологий и анализа больших данных**

**Макрушин С.В.**

**ТЕХНОЛОГИИ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ**

**Рабочая программа дисциплины**

для студентов, обучающихся по направлению подготовки  
01.03.02 Прикладная математика и информатика,  
ОП «Прикладная математика и информатика  
(Анализ данных и принятие решений в экономике и финансах)

**Москва 2021**

**Федеральное государственное образовательное бюджетное  
учреждение высшего образования  
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ  
ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»  
(Финансовый университет)**

**Департамент анализа данных и машинного обучения  
Факультета информационных технологий и анализа больших данных**

УТВЕРЖДАЮ

Проректор по учебной и  
методической работе

\_\_\_\_\_ Е.А. Каменева  
29.12.2021 г.

**Макрушин С.В.**

**ТЕХНОЛОГИИ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ**

**Рабочая программа дисциплины**

для студентов, обучающихся по направлению подготовки  
01.03.02 Прикладная математика и информатика,  
ОП «Прикладная математика и информатика  
(Анализ данных и принятие решений в экономике и финансах)

*Рекомендовано Ученым советом  
Факультета информационных технологий и анализа больших данных  
(протокол № 15 от 22.12.2021 г.)*

*Одобрено Советом учебно-научного  
Департамента анализа данных и машинного обучения  
(протокол № 5 от 21.12.2021 г.)*

**Москва 2021**

**УДК 004.9**  
**ББК 32.973.202**

**Рецензент:** **Феклин В.Г.**, кандидат физико-математических наук, доцент Департамента анализа данных и машинного обучения.

**Макрушин С.В.**

**«Технологии обработки больших данных».** Рабочая программа дисциплины для студентов, обучающихся по направлению подготовки 01.03.02 Прикладная математика и информатика, ОП «Прикладная математика и информатика» (Анализ данных и принятие решений в экономике и финансах) – М.: Финансовый университет, Департамент анализа данных и машинного обучения, 2021 г. – 26. с.

Дисциплина «Технологии обработки больших данных» относится к Обще-профессиональному циклу дисциплин по направлению подготовки 01.03.02 Прикладная математика и информатика, ОП «Прикладная математика и информатика» (Анализ данных и принятие решений в экономике и финансах).

Рабочая программа дисциплины содержит цели и задачи дисциплины, требования к результатам освоения дисциплины, содержание дисциплины, тематику практических занятий и технологии их проведения, формы самостоятельной работы, систему оценивания, учебно-методическое и информационное обеспечение дисциплины

**УДК 004.9**  
**ББК 32.973.202**

*Учебное издание*

*Макрушин Сергей Вячеславович*

**Технологии обработки больших данных**

**Рабочая программа дисциплины**

Компьютерный набор, верстка: **С. В. Макрушин**

Формат 60x90/16. Гарнитура *Times New Roman*

Усл. п.л. 1,1. Изд. № 28.2 - 2021. Тираж - 36 экз.

Заказ № \_\_\_\_\_

**Отпечатано в Финансовом университете**

© **С. В. Макрушин, 2021**  
© **Финансовый университет, 2021**

## Оглавление

1. Наименование дисциплины.....	4
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине .....	4
3. Место дисциплины в структуре образовательной программы .....	5
4. Объем дисциплины(модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся.....	5
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий .....	6
5.1. Содержание дисциплины .....	6
5.2. Учебно-тематический план.....	9
5.3. Содержание семинаров, практических занятий .....	11
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине .....	12
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы.....	12
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю.....	14
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине.....	17
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины .....	24
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины .....	24
10. Методические указания для обучающихся по освоению дисциплины.	25
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем.....	26
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине. ....	26

## 1. Наименование дисциплины

«Технологии обработки больших данных».

## 2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ОПК-2	Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	1. Системно подходит к выбору математических методов и систем программирования для решения прикладных задач.	<b>Знать</b> основные ИТ работы с большими данными и современные технологии обработки больших данных <b>Уметь</b> загружать, преобразовывать и обрабатывать большие данные с помощью современных технологий
		2. Разрабатывает алгоритмы решения прикладных задач с использованием математических методов.	<b>Знать</b> методы решения задач прикладных математических моделей с помощью инструментов обработки больших данных. <b>Уметь</b> применять методы обработки больших данных для решения прикладных задач.
		3. Реализует алгоритмы с использованием современных систем программирования.	<b>Знать</b> современные ИТ и инструменты обработки больших данных, применимые для использования в реализации математических моделей. <b>Уметь</b> использовать современные ИТ и инструменты для обработки больших данных в решении прикладных математических задач.
ОПК-4	Способен понимать принципы работы современных информационных технологий и использовать их для	1. Использует информационно-коммуникационные технологии при поиске необходимой информации, сбора,	<b>Знать</b> современные инструменты обработки больших данных. <b>Уметь</b> обрабатывать данные с помощью специализированных

	решения задач профессиональной деятельности	визуализации и обработки данных.	инструментов, предназначенных для работы с данными соответствующего размера.
		2. Осуществляет рациональный выбор программного продукта в зависимости от поставленной задачи.	<b>Знать</b> рынок современных решений для обработки данных разного размера. <b>Уметь</b> определять критерии выбора ИТ систем для обработки данных и осуществлять выбор систем исходя из этих критериев.
		3. Владеет навыками обеспечения информационной безопасности автоматизированных систем.	<b>Знать</b> ограничения безопасности, связанные с использованием систем обработки больших данных. <b>Уметь</b> выбирать ИТ-решения для обработки больших данных исходя из требований безопасности.

### 3. Место дисциплины в структуре образовательной программы

Дисциплина «Технологии обработки больших данных» относится к Общепрофессиональному циклу дисциплин по направлению подготовки 01.03.02 Прикладная математика и информатика, ОП «Прикладная математика и информатика (Анализ данных и принятие решений в экономике и финансах).

Изучение дисциплины «Технологии обработки больших данных» основывается на сумме знаний, полученных при изучении дисциплины «Алгоритмы и структуры данных в языке Python». Для изучения данной дисциплины студент должен обладать базовыми знаниями в области информационных технологий и программирования, навыками программирования на языке Python.

### 4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

*Очная форма обучения, 2021 г.п. и т.д.*

Вид учебной работы по дисциплине	Всего (в з.е. и часах)	Семестр 3 (в часах)	Семестр 4 (в часах)
<b>Общая трудоёмкость дисциплины</b>	7/252	108	144
<b>Контактная работа- Аудиторные занятия</b>	<b>100</b>	<b>50</b>	<b>50</b>
Лекции	32	16	16
Семинары, практические занятия	68	34	34
<b>Самостоятельная работа</b>	<b>152</b>	<b>58</b>	<b>94</b>
Вид текущего контроля	контрольные работы	контрольная работа	контрольная работа
Вид промежуточной аттестации	Зачет, экзамен	зачет	экзамен

## **5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий**

### **5.1. Содержание дисциплины**

#### **Тема 1. Библиотека NumPy и Pandas.**

В рамках темы рассматривается технологический стек Python для обработки и анализа данных, возможности Python как glue language, специфика библиотеки NumPy и ее роль в экосистеме Python. Организация массивов в NumPy: хранение данных, создание массивов, принципы реализации операций с едиными исходными данными. Универсальные функции и применение функций по осям в NumPy. Принцип распространения значений при выполнении операций в NumPy: общий алгоритм и примеры Маскирование и прихотливое индексирование в NumPy.

В рамках темы рассматриваются возможности библиотеки Pandas. Организация Pandas DataFrame и организация индексации для DataFrame и Series; применение универсальных функций и работа с пустыми значениями в Pandas. Объединение данных из нескольких Pandas DataFrame: общая логика и примеры. Рассматривается операция GroupBy в Pandas DataFrame и реализация в ней подхода «разбиение, применение и объединение».

## **Тема 2. Профилирование процессов обработки данных, библиотека Numba и векторизация в Numpy и Numba.**

В рамках темы рассматривается профилирование реализации алгоритмов на Python, принципы решения задачи оптимизации производительности алгоритма. Библиотека Numba: принципы работы, базовые примеры использования. Векторизация в numpy: ключевые параметры функции, примеры применения, использование обобщенной сигнатуры функции.

## **Тема 3. Использование различных форматов файлов в задачах обработки данных.**

Принципы работы с файлами, файлы и операционные системы. Специфика текстовых и бинарных файлов, форматы файлов CSV и Pickle, представление данных в этих форматах и взаимодействие с ними в Python.

Задача сериализации и десериализации данных и использование различных форматов файлов для ее решения. Описание формата файла JSON и пример описания данных в этом формате и взаимодействия с ним в Python.

Формат XML и модель DOM: общая характеристика, пример описания данных в XML и DOM, работа с ними с помощью библиотеки BeautifulSoup.

Проблематика форматов файлов для хранения и обработки больших данных. Форматы файлов NPY и HDF общая характеристика, пример взаимодействия с данными этих форматов в Python

## **Тема 4. Взаимодействие с базой данных и Excel в приложениях обработки данных.**

Взаимодействие из Python с базой данных на примере API SQLite. Базовые возможности работы с транзакциями. Возможности использования Excel для внешних приложений обработки данных. Взаимодействие с Excel из Python с помощью библиотеки XLWings: принципы работы и примеры использования.

## **Тема 5. Работа со строками в приложениях обработки данных.**

Основы работы с регулярными выражениями: базовый синтаксис, примеры. Модуль *re* в Python. Примеры использования регулярных выражений. Использование хеширования при работе со строками. Строки в библиотеке numpy.



## **Тема 6. Введение в обработку текста на естественном языке в задачах обработки данных.**

Сегментация и токенизация текста на естественном языке, стемминг и лемматизация, примеры на Python. Использование мемоизации на примере работы со строками. Расстояние Левенштейна: определение, алгоритм эффективного поиска оптимального редакционного предписания, пример поиска на Python. Векторное представление текста на естественном языке: общий алгоритм подходов TF; TF-IDF.

## **Тема 7. Параллельная обработка данных.**

В рамках темы рассматривается специфика современного аппаратного обеспечения для обработки больших данных и проблема масштабируемости параллельных вычислений. Многопроцессорные архитектуры с общей и разделяемой памятью – специфика и сравнение.

Подходы к декомпозиции крупных вычислительных задач на подзадачи для параллельного исполнения. Модели параллельного программирования и их сочетаемость с архитектурами параллельных вычислительных систем. Специфика различия между потоками и процессами.

Проблема Global Interpreter Lock в Python и способы обхода ее ограничений. Модуль Python multiprocessing – назначение и основные возможности, API multiprocessing.Pool.

## **Тема 8. Библиотека Dask.**

В рамках темы рассматривается библиотека для анализа больших объемов данных Python Dask, различные предлагаемые ей подходы к обработке данных. В частности, три ключевых структуры данных Dask: Dask.Array, Dask.DataFrame и Dask.Bag их специфика и принцип выбора структур данных при решении задач. Рассматривается граф зависимостей задач, как ключевая структура для организации параллельной обработки данных в Python Dask. Рассматривается принцип и примеры использования распараллеливание алгоритмов с помощью dask.delayed .

Рассматривается структура данных Dask.Array, специфика ее реализации и применения, процедура создания, поддерживаемые Dask.Array операции и ее отличия от NumPy ndarray. Рассматривается структура данных Dask.DataFrame, специфика ее реализации и применения, процедура создания, ограничения использования Dask.DataFrame. Рассматриваются операции мэппинга в Dask.DataFrame и операции Dask.DataFrame работающие со

скользящим окном. Рассматривается структура данных Dask.Bag, специфика ее реализации и применения, процедура создания, поддерживаемые Dask.Bag операции. Организация вычислений с помощью Map / Filter / Reduce : общий принцип и специфика параллельной реализации обработки данных с помощью Dask.Bag.

## **Тема 9. Обзор проблем обработки больших данных и вычисления общего назначения на GPU**

Большие данные – определение и причины возникновения задач обработки больших данных. Вызовы «Больших данных»: объем данных, слабая структурированность данных, связность данных, обработка данных с помощью независимых сервисов. Специфика аппаратного обеспечения для решения задач обработки больших данных. Проблема выбора типичных средств обработки данных, адекватных различным объемам данных. Принцип обработки данных на базе операций map / filter / reduce, принципы архитектуры hadoop. Источники больших данных и прикладные задачи обработки больших данных.

История развития и общая характеристика GPU. Архитектура Nvidia CUDA. Принципы организации вычислений в архитектуре Nvidia CUDA. Знакомство с библиотекой PyTorch. Понятие тензора в PyTorch. Базовые операции с тензорами в PyTorch.

### **5.2. Учебно-тематический план**

№ п/п	Наименование темы (раздела) дисциплины	Трудоемкость в часах					Формы текущего контроля успевае- мости
		Всего	Контактная работа- Аудиторная работа			Само- стоя- тельная работа	
			Общая, в т.ч.:	Лекции	Семинары, практические занятия		
1	Библиотека NumPy и Pandas	24	12	4	8	12	Участие в решении задач на практиче- ских заня- тиях. Об- суждения по резуль-
2	Профилирование процессов обра- ботки данных, библиотека Numba и вектори- зация в NumPy и Numba	18	8	2	6	10	

3	Использование различных форматов файлов в задачах обработки данных	24	12	4	8	12	Участие в решении задач на практических занятиях. Обсуждения по результатам самостоятельной работы
4	Взаимодействие с базой данных и Excel в приложениях обработки данных	24	12	4	8	12	
5	Работа со строками в приложениях обработки данных	18	6	2	4	12	
6	Введение в обработку текста на естественном языке в задачах обработки данных	26	6	2	4	20	
7	Параллельная обработка данных	26	6	2	4	20	
8	Библиотека Dask	60	26	8	18	34	
9	Обзор проблем обработки больших данных и вычисления общего назначения на GPU	32	12	4	8	20	Согласно учебному плану: контрольные работы
	В целом по дисциплине	252	100	32	68	152	
	Итого в %	100	40	32	68	60	

### 5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарских, практических занятиях, рекомендуемые источники из разделов 8,9 (указывается раздел и порядковый номер источника)	Формы проведения занятий
Библиотека NumPy и Pandas	<ul style="list-style-type: none"> <li>• Технологический стек Python для обработки и анализа данных</li> <li>• Возможности Python как glue language</li> <li>• Организация массивов в NumPy: хранение данных, создание массивов</li> <li>• Принципы реализации операций с едиными исходными данными. Универсальные функции и применение функций по осям в NumPy.</li> <li>• Организация Pandas DataFrame и организация индексации для DataFrame и Series.</li> <li>• Применение универсальных функций и работа с пустыми значениями в Pandas.</li> <li>• Объединение данных из нескольких Pandas DataFrame: общая логика и примеры.</li> </ul> <p>8[1], 9[9], 9[10]</p>	Интерактивная форма, работа на компьютере
Профилирование процессов обработки данных, библиотека Numba и векторизация в Numpy и Numba	<ul style="list-style-type: none"> <li>• профилирование реализации алгоритмов на Python</li> <li>• принципы решения задачи оптимизации производительности алгоритма</li> <li>• Библиотека Numba: принципы работы, базовые примеры использования.</li> </ul> <p>8[1], 8[2], 9[1], 9[2], 9[3]</p>	Интерактивная форма, работа на компьютере
Использование различных форматов файлов в задачах обработки данных	<ul style="list-style-type: none"> <li>• Формат файлов CSV, представление данных в этом формате и взаимодействие с ним в Python.</li> <li>• Формат файлов Pickle, представление данных в этом формате и взаимодействие с ним в Python.</li> <li>• Формат файлов JSON, представление данных в этом формате и взаимодействие с ним в Python.</li> <li>• Формат XML и модель DOM: общая характеристика, пример описания данных в XML и DOM</li> <li>• Работа с XML с помощью библиотеки BeautifulSoup.</li> </ul> <p>8[1], 8[2], 9[3], 9[4]</p>	Интерактивная форма, работа на компьютере
Взаимодействие с базой данных и Excel в приложениях обработки данных	<ul style="list-style-type: none"> <li>• Взаимодействие из Python с базой данных с помощью API SQLite.</li> <li>• Взаимодействие с Excel из Python с помощью библиотеки XLWings.</li> </ul> <p>8[1], 8[2].</p>	Интерактивная форма, работа на компьютере
Работа со строками в приложениях обработки данных	<ul style="list-style-type: none"> <li>• Основы работы с регулярными выражениями: базовый синтаксис, примеры.</li> <li>• Модуль re в Python.</li> </ul> <p>8[1], 8[2], 9[4]</p>	Интерактивная форма, работа на компьютере

Введение в обработку текста на естественном языке в задачах обработки данных.	<ul style="list-style-type: none"> <li>Сегментация и токенизация текста на естественном языке, стемминг и лемматизация, примеры на Python.</li> <li>Расстояние Левенштейна: определение, алгоритм эффективного поиска оптимального редакционного предписания, пример поиска на Python.</li> </ul> 8[1], 8[2], 9[4], 9[5], 9[6]	Интерактивная форма, работа на компьютере
Параллельная обработка данных	<ul style="list-style-type: none"> <li>специфика современного аппаратного обеспечения для обработки больших данных и проблема масштабируемости параллельных вычислений.</li> <li>Подходы к декомпозиции крупных вычислительных задач на подзадачи для параллельного исполнения.</li> <li>Проблема Global Interpreter Lock в Python и способы обхода ее ограничений.</li> <li>Модуль Python multiprocessing – назначение и основные возможности, API multiprocessing.Pool.</li> </ul> 8[1], 8[2], 9[5], 9[7]	Интерактивная форма, работа на компьютере
Библиотека Dask	<ul style="list-style-type: none"> <li>Подход к обработке данных с помощью библиотеки Dask.</li> <li>Структура данных Dask.Array – принцип работы, API, примеры использования.</li> <li>Структура данных Dask.DataFrame – принцип работы, API, примеры использования.</li> <li>Структура данных Dask.Bag – принцип работы, API, примеры использования.</li> </ul> 8[1], 8[2], 9[8], 9[10], 9[11]	Интерактивная форма, работа на компьютере
Обзор проблем обработки больших данных и вычисления общего назначения на GPU	<ul style="list-style-type: none"> <li>Вызовы «Больших данных»: объем данных, слабая структурированность данных, связность данных, обработка данных с помощью независимых сервисов.</li> <li>Источники больших данных и прикладные задачи обработки больших данных.</li> <li>Архитектура Nvidia CUDA. Принципы организации вычислений в архитектуре Nvidia CUDA.</li> </ul> 8[1], 8[2], 9[8]	Интерактивная форма, работа на компьютере

## 6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

### 6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Библиотека NumPy	<ul style="list-style-type: none"> <li>Принцип распространения значений при</li> </ul>	Обзор литературы и

и Pandas	<p>выполнении операций в NumPy: общий алгоритм и примеры.</p> <ul style="list-style-type: none"> <li>• Маскирование и прихотливое индексирование в NumPy.</li> <li>• Операция GroupBy в Pandas DataFrame и реализация в ней подхода «разбиение, применение и объединение».</li> </ul> <p>8[1], 9[9], 9[10]</p>	веб-источников. Самостоятельное освоение инструментов аналитической обработки. Решение задач.
Профилирование процессов обработки данных, библиотека Numba и векторизация в Numpy и Numba	<ul style="list-style-type: none"> <li>• Векторизация в numpy: ключевые параметры функции, примеры применения</li> <li>• Использование обобщенной сигнатуры функции в numpy и numba.</li> </ul> <p>8[1], 8[2], 9[1], 9[2], 9[3]</p>	Обзор литературы и веб-источников. Самостоятельное освоение инструментов аналитической обработки. Решение задач.
Использование различных форматов файлов в задачах обработки данных	<ul style="list-style-type: none"> <li>• Формат файлов NPY, представление данных в этом формате и взаимодействие с ним в Python.</li> <li>• Формат файлов HDF, представление данных в этом формате и взаимодействие с ним в Python.</li> </ul> <p>8[1], 8[2], 9[3], 9[4]</p>	Обзор литературы и веб-источников. Самостоятельное освоение инструментов аналитической обработки. Решение задач.
Взаимодействие с базой данных и Excel в приложениях обработки данных	<ul style="list-style-type: none"> <li>• Базовые возможности работы с транзакциями с помощью API SQLite.</li> <li>• Продвинутое взаимодействие с Excel из Python с помощью библиотеки XLWings.</li> </ul> <p>8[1], 8[2], 9[5]</p>	Обзор литературы и веб-источников. Самостоятельное освоение инструментов аналитической обработки. Решение задач.
Работа со строками в приложениях обработки данных	<ul style="list-style-type: none"> <li>• Использование хэширования при работе со строками.</li> <li>• Строки в библиотеке numpy.</li> </ul> <p>8[1], 8[2], 9[4]</p>	Обзор литературы и веб-источников. Самостоятельное освоение инструментов аналитической обработки. Решение задач.
Введение в обработку текста на естественном языке в задачах обработки данных.	<ul style="list-style-type: none"> <li>• Использование мемоизации на примере работы со строками.</li> <li>• Векторное представление текста на естественном языке: общий алгоритм подходов TF; TF-IDF.</li> </ul> <p>8[1], 8[2], 9[4], 9[5], 9[6]</p>	Обзор литературы и веб-источников. Самостоятельное освоение инструментов аналитической обработки. Решение задач.
Параллельная обработка данных	<ul style="list-style-type: none"> <li>• Модели параллельного программирования и их сочетаемость с архитектурами параллельных вычислительных систем.</li> <li>• Специфика различия между потоками и процессами.</li> <li>• Многопроцессорные архитектуры с общей и разделяемой памятью – специфика и сравнение.</li> </ul>	Обзор литературы и веб-источников. Самостоятельное освоение инструментов аналитической обработки. Решение задач.
Библиотека Dask	<ul style="list-style-type: none"> <li>• Организация вычислений с помощью Map / Filter / Reduce: общий принцип и</li> </ul>	Обзор литературы и

	специфика параллельной реализации обработки данных с помощью Dask.Bag. • Организация вычислений с помощью API Dask Delayed. 8[1], 8[2], 9[8], 9[10], 9[11]	веб-источников. Самостоятельное освоение инструментов аналитической обработки. Решение задач.
Обзор проблем обработки больших данных и вычисления общего назначения на GPU	• Специфика аппаратного обеспечения для решения задач обработки больших данных. • Проблема выбора типичных средств обработки данных, адекватных различным объемам данных. • Знакомство с библиотекой PyTorch. Понятие тензора в PyTorch. Базовые операции с тензорами в PyTorch. 8[1], 8[2], 9[8]	Обзор литературы и веб-источников. Самостоятельное освоение инструментов аналитической обработки. Решение задач.

## 6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

### *Примерные вопросы к контрольной работе № 1*

1. Большие данные – определение и причины возникновения задач обработки больших данных
2. Специфика современного аппаратного обеспечения для обработки больших данных и проблема масштабируемости параллельных вычислений
3. Выбор типичных средств обработки данных, адекватных различным объемам данных; принцип обработки данных на базе операций map / filter / reduce
4. Многопроцессорные архитектуры с общей и разделяемой памятью – специфика и сравнение
5. Подходы к декомпозиции крупных вычислительных задач на подзадачи для параллельного исполнения
6. Модели параллельного программирования и их сочетаемость с архитектурами параллельных вычислительных систем

### *Примеры заданий для контрольной работы № 1*

#### **Задание 1**

1. В массиве чисел, хранящихся в файле finance.hdf5, найти строку (вывести ее индекс и содержащиеся значения), в которой более всего

значений превышающих среднее значение по всему массиву. Для расчётов использовать `dask.array`

2. В массиве чисел, хранящихся в файле `finance.hdf5`, подсчитать количество строк, в которых более 600 значений больше среднего значения по всему массиву. Для расчётов использовать `dask.array`.
3. В массиве чисел, хранящихся в файле `finance.hdf5`, подсчитать количество значений, не отклоняющихся от среднего значения более чем на 3 стандартных отклонения. Для расчетов использовать `dask.array`

## **Задание 2**

1. В `accounts*.csv` найти `id`, для которого в столбце `amount` встречается наибольшее количество значений кратных трем. Выполнить задание с использованием `Dask`, распараллелив процесс обработки данных
2. В `accounts*.csv` найти `id`, для которого сумма положительных значений в столбце `amount` наибольшая. Выполнить задание с использованием `Dask`, распараллелив процесс обработки данных
3. В `accounts*.csv` найти `id`, для которого в столбце `amount` встречается наибольшее количество значений между 1000 и 1500. Выполнить задание с использованием `Dask`, распараллелив процесс обработки данных

## ***Примерные вопросы к контрольной работе № 2***

1. Различия между потоками и процессами, различие между различными планировщиками в `Dask`
2. Граф зависимостей задач – суть структуры данных, ее построение и использование в `Dask`
3. Три ключевых структуры данных `Dask`: их специфика и принцип выбора структуры данных при решении задач
4. `Dask.Array` – структура данных, специфика реализации и применения, процедура создания
5. `Dask.Array` – поддерживаемые операции и отличия от `NumPy ndarray`
6. Распараллеливание алгоритмов с помощью `dask.delayed` – принцип и примеры использования
7. Дополнительные параметры декоратора `dask.delayed` – назначение и примеры использования



8. Использование `dask.delayed` для объектов и операции над объектами `dask.delayed`, включая ограничения их использования
9. `Dask.DataFrame` - структура данных, специфика реализации и применения, процедура создания `Dask.DataFrame`
10. Ограничения использования `Dask.DataFrame` и операции мэппинга в `Dask.DataFrame`
11. Поддержка `Dask.DataFrame` операций работающих со скользящим окном
12. Совместное использование промежуточных результатов в `Dask`: принцип работы и примеры использования
13. `Dask.Bag` - структура данных, специфика реализации и применения, процедура создания `Dask.Bag`
14. Организация вычислений с помощью `Map / Filter / Reduce` : общий принцип и специфика параллельной реализации обработки данных в `Dask.Bag`
15. API `Dask.Bag` – функции мэппинга, фильтрации и преобразования

### ***Примеры заданий для контрольной работы № 2***

#### **Задание 1**

1. В (20 баллов) Датасет: `all_k.zip`

Подсчитать, сколько раз в текстовых файлах, лежащих в `all_k.zip`, встречаются предложения трех видов: вопросительные (в окончании имеют вопросительный знак), побудительные (в окончании имеют восклицательный знак и не имеют вопросительного) и повествовательные (в окончании имеют точку или троеточие, при этом нужно исключить учет точек, встречающихся в сокращениях, таких как "т.к.").

Выполнить задание с использованием `Dask` (корректным!), распараллелив процесс обработки данных (использование `Dask` должно приводить к истинной параллельной обработке данных).

#### **Задание 2**

1. Датасет: `all_k.zip`

Подсчитать, сколько раз встречается каждое из личных местоимений в именительном падеже (полный список: я, ты, он, она, оно, мы, вы, они) в текстовых файлах, лежащих в папке: `all_k.zip`.

Выполнить задание с корректным использованием Dask, распараллелив процесс обработки данных (использование Dask должно приводить к истинной параллельной обработке данных).

### **Критерии балльной оценки различных форм текущего контроля успеваемости**

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Департамента анализа данных и машинного обучения.

## **7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине**

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. **«Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине».**

**Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний**

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения ( умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ОПК-2 Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и ре-	1. Системно подходит к выбору математических методов и систем программирования для решения прикладных задач.	<b>Знать</b> основные ИТ работы с большими данными и современные технологии обработки больших данных  <b>Уметь</b> загружать, преобразовывать и	Перечислить основные технологии обработки больших данных.  Выполните загрузку датасета формата hdf5 для обработки в

ализации алгоритмов решения прикладных задач		обрабатывать большие данные с помощью современных технологий	библиотеке Dask.
	2. Разрабатывает алгоритмы решения прикладных задач с использованием математических методов.	<b>Знать</b> методы решения задач прикладных математических моделей с помощью инструментов обработки больших данных.  <b>Уметь</b> применять методы обработки больших данных для решения прикладных задач.	Перечислите технологии позволяющие выполнять математические вычисления с помощью параллельной обработки больших массивов данных.  Выполните задачу матричного умножения для двух больших массивов данных.
	3. Реализует алгоритмы с использованием современных систем программирования.	<b>Знать</b> современные ИТ и инструменты обработки больших данных, применимые для использования в реализации математических моделей.  <b>Уметь</b> использовать современные ИТ и инструменты для обработки больших данных в решении прикладных математических задач.	Перечислите технологии позволяющие выполнять математические вычисления и математическое моделирование с помощью параллельной обработки больших массивов данных.  Выполните нормализации большого числового массива.
ОПК-4 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	1. Использует информационно-коммуникационные технологии при поиске необходимой информации, сбора, визуализации и обработки данных.	<b>Знать</b> современные инструменты обработки больших данных.  <b>Уметь</b> обрабатывать данные с помощью специализированных инструментов, предназначенных для работы с данными соответствующего размера.	Перечислить основные технологии обработки больших данных на базе языка Python.  Выполните загрузку датасета формата csv для обработки в библиотеке Dask.

	2. Осуществляет рациональный выбор программного продукта в зависимости от поставленной задачи.	<p><b>Знать</b> рынок современных решений для обработки данных разного размера.</p> <p><b>Уметь</b> определять критерии выбора ИТ систем для обработки данных и осуществлять выбор систем исходя из этих критериев.</p>	<p>Перечислить основные технологии обработки больших данных на базе языка Python.</p> <p>Уметь выбирать технологии обработки больших с открытым исходным кодом.</p>
	3. Владеет навыками обеспечения информационной безопасности автоматизированных систем.	<p><b>Знать</b> ограничения безопасности, связанные с использованием систем обработки больших данных.</p> <p><b>Уметь</b> выбирать ИТ-решения для обработки больших данных исходя из требований безопасности.</p>	<p>Перечислить форматы хранения информации, распространенные при решении задач обработки больших данных.</p> <p>Уметь выбирать технологии обработки больших с открытым исходным кодом.</p>

### *Теоретические вопросы для подготовки к зачету*

1. Профилирование реализации алгоритмов на Python, принципы решения задачи оптимизации производительности алгоритма
2. Проблема Global Interpreter Lock в Python и способы обхода ее ограничений
3. Технологический стек Python для обработки и анализа данных, Python как glue language, специфика библиотеки NumPy и ее роль в экосистеме Python
4. Организация массивов в NumPy: хранение данных, создание массивов, принципы реализации операций с едиными исходными данными
5. Универсальные функции и применение функций по осям в NumPy
6. Принцип распространения значений при выполнении операций в NumPy: общий алгоритм и примеры
7. Маскирование и прихотливое индексирование в NumPy

8. Модуль `multiprocessing` – назначение и основные возможности, API `multiprocessing.Pool`
9. Различия между потоками и процессами, различие между различными планировщиками в Dask
10. Граф зависимостей задач – суть структуры данных, ее построение и использование в Dask
11. Выбор типичных средств обработки данных, адекватных различным объемам данных; принцип обработки данных на базе операций `map` / `filter` / `reduce`
12. Многопроцессорные архитектуры с общей и разделяемой памятью – специфика и сравнение
13. Подходы к декомпозиции крупных вычислительных задач на подзадачи для параллельного исполнения
14. Модели параллельного программирования и их сочетаемость с архитектурами параллельных вычислительных систем
15. Профилирование реализации алгоритмов на Python, принципы решения задачи оптимизации производительности алгоритма
16. Проблема `Global Interpreter Lock` в Python и способы обхода ее ограничений
17. Технологический стек Python для обработки и анализа данных, Python как `glue language`, специфика библиотеки `NumPy` и ее роль в экосистеме Python
18. Организация массивов в `NumPy`: хранение данных, создание массивов, принципы реализации операций с едиными исходными данными
19. Универсальные функции и применение функций по осям в `NumPy`
20. Принцип распространения значений при выполнении операций в `NumPy`: общий алгоритм и примеры
21. Маскирование и прихотливое индексирование в `NumPy`
22. Модуль `multiprocessing` – назначение и основные возможности, API `multiprocessing.Pool`

### ***Теоретические вопросы для подготовки к экзамену***

1. Большие данные – определение и причины возникновения задач обработки больших данных
2. Специфика современного аппаратного обеспечения для обработки больших данных и проблема масштабируемости параллельных вычислений
3. Выбор типичных средств обработки данных, адекватных различным объемам данных; принцип обработки данных на базе операций map / filter / reduce
4. Многопроцессорные архитектуры с общей и разделяемой памятью – специфика и сравнение
5. Подходы к декомпозиции крупных вычислительных задач на подзадачи для параллельного исполнения
6. Модели параллельного программирования и их сочетаемость с архитектурами параллельных вычислительных систем
7. Профилирование реализации алгоритмов на Python, принципы решения задачи оптимизации производительности алгоритма
8. Проблема Global Interpreter Lock в Python и способы обхода ее ограничений
9. Технологический стек Python для обработки и анализа данных, Python как glue language, специфика библиотеки NumPy и ее роль в экосистеме Python
10. Организация массивов в NumPy: хранение данных, создание массивов, принципы реализации операций с едиными исходными данными
11. Универсальные функции и применение функций по осям в NumPy
12. Принцип распространения значений при выполнении операций в NumPy: общий алгоритм и примеры
13. Маскирование и прихотливое индексирование в NumPy
14. Векторизация в numpy: ключевые параметры функции, примеры применения, использование обобщенной сигнатуры функции
15. Numba: принципы работы, базовые примеры использования
16. Организация Pandas DataFrame и организация индексации для DataFrame и Series
17. Применение универсальных функций и работа с пустыми значениями в Pandas
18. Объединение данных из нескольких Pandas DataFrame: общая логика и примеры

- 19.Операция GroupBy в Pandas DataFrame и реализация в ней подхода «разбиение, применение и объединение»
- 20.Специфика текстовых и бинарных файлов, форматы файлов CSV и Pickle, представление данных в этих форматах и взаимодействие с ними в Python
- 21.Задача сериализации и десериализации, описание формата файла JSON и пример описания данных в этом формате и взаимодействия с ним в Python
- 22.Формат XML и модель DOM: общая характеристика, пример описания данных в XML и DOM, работа с ними с помощью библиотеки BeautifulSoup
- 23.Форматы файлов NPY и HDF общая характеристика, пример взаимодействия с данными этих форматов в Python
- 24.Взаимодействие из Python с базой данных на примере API SQLite, базовые возможности работы с транзакциями
- 25.Взаимодействие с Excel из Python с помощью XLWings: принципы работы и примеры использования
- 26.Основы работы с регулярными выражениями: базовый синтаксис, примеры использования модуля re в Python
- 27.Сегментация и токенизация текста на естественном языке, стемминг и лемматизация, примеры на Python
- 28.Расстояние Левенштейна: определение, алгоритм эффективного поиска оптимального редакционного предписания, пример поиска на Python
- 29.Векторное представление текста на естественном языке: общий алгоритм подходов TF; TF-IDF
- 30.Модуль multiprocessing – назначение и основные возможности, API multiprocessing.Pool
- 31.Различия между потоками и процессами, различие между различными планировщиками в Dask
- 32.Граф зависимостей задач – суть структуры данных, ее построение и использование в Dask
- 33.Три ключевых структуры данных Dask: их специфика и принцип выбора структуры данных при решении задач
- 34.Dask.Array – структура данных, специфика реализации и применения, процедура создания
- 35.Dask.Array – поддерживаемые операции и отличия от NumPy ndarray

- 36.Распараллеливание алгоритмов с помощью `dask.delayed` – принцип и примеры использования
- 37.Дополнительные параметры декоратора `dask.delayed` – назначение и примеры использования
38. Использование `dask.delayed` для объектов и операции над объектами `dask.delayed`, включая ограничения их использования
- 39.`Dask.DataFrame` - структура данных, специфика реализации и применения, процедура создания `Dask.DataFrame`
- 40.Ограничения использования `Dask.DataFrame` и операции мэппинга в `Dask.DataFrame`
- 41.Поддержка `Dask.DataFrame` операций работающих со скользящим окном
- 42.Совместное использование промежуточных результатов в `Dask`: принцип работы и примеры использования
- 43.`Dask.Bag` - структура данных, специфика реализации и применения, процедура создания `Dask.Bag`
- 44.Организация вычислений с помощью `Map / Filter / Reduce` : общий принцип и специфика параллельной реализации обработки данных в `Dask.Bag`
- 45.API `Dask.Bag` – функции мэппинга, фильтрации и преобразования
- 46.API `Dask.Bag` – функции группировки и свертки

### **Пример экзаменационного билета**

1. (20 баллов) Создать двухмерный массив 30 на 4, содержащий случайные целые числа от 0 до 100.  
Интерпретируя массив как 30 векторов из 4х компонент, вернуть массив 5 на 4, состоящий из векторов с наибольшей длиной (евклидовой нормой).  
Решить задачу средствами `numpy` и/или `pandas`. Не использовать циклы и конструкции стандартного Python там, где можно использовать возможности данных библиотек.
2. (20 баллов) Датасет: `Chinook_Sqlite.sqlite`  
С помощью кода на Python с использованием `sqlite3` и SQL решить задачу. Реализовать функции на Python:
  1. Которая возвращает все имеющиеся плейлисты.
  2. Которая по имени плейлиста возвращает количество треков в нем и их суммарную продолжительность.
3. (20 баллов) Датасет: `all_k.zip`  
Подсчитать, сколько раз во всех текстовых файлах, лежащих в



all\_k.zip, встречаются реплики прямой речи, оформленные в виде диалога (В этом случае каждая реплика начинается с новой строки, перед репликами ставится тире (перед тире возможны различные пробельные символы)). Выполнить задание с использованием Dask, распараллелив процесс обработки данных.

Выполнить задание с использованием Dask (корректным!), распараллелив процесс обработки данных (использование Dask должно приводить к истинной параллельной обработке данных).

## **8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

### **Основная литература**

1. Колдаев, В. Д. Структуры и алгоритмы обработки данных : учебное пособие / В. Д. Колдаев. - Москва : РИОР : ИНФРА-М, 2020. - 296 с. - ЭБС ZNANIUM.com. - URL: <https://znanium.com/catalog/product/1054007> (дата обращения: 07.02.2022). – Текст: электронный.

### **Дополнительная литература**

2. Нагаева, И. А. Основы алгоритмизации и программирования: практикум : учебное пособие / И. А. Нагаева, И. А. Кузнецов. – Москва : Берлин : Директ-Медиа, 2021. – 169 с. – ЭБС Университетская библиотека ONLINE. – URL: <https://biblioclub.ru/index.php?page=book&id=598404> (дата обращения: 07.02.2022). – Текст : электронный.

## **9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>

2. Электронно-библиотечная система BOOK.RU <http://www.book.ru>

3. Электронно-библиотечная система «Университетская библиотека ОНЛАЙН» <http://biblioclub.ru/>

4. Электронно-библиотечная система Znanium <http://www.znanium.com>

PyLru 1.0.9 [Электронный ресурс]: сайт. – Режим доступа: <https://pypi.python.org/pypi/pylru>

5. Python Data Analysis Library [Электронный ресурс]: сайт. – Режим доступа: <http://pandas.pydata.org/>
6. Python Documentation [Электронный ресурс]: сайт. – Режим доступа: <http://python.org/doc/>
7. Python Standard Library [Электронный ресурс]: сайт. – Режим доступа: <https://docs.python.org/2/library/>
8. Scikit-learn Machine Learning in Python [Электронный ресурс]: сайт. – Режим доступа: <http://scikit-learn.org>
9. Официальный сайт продукта <https://www.python.org/>
10. Каталог курсов Интернет Университета Информационных Технологий <http://www.intuit.ru/>
11. The Python Tutorial // <https://docs.python.org/3/tutorial/index.html>
12. NumPy User Guide // <http://docs.scipy.org/doc/numpy/user/index.html>
13. Pandas User Guide <http://pandas.pydata.org/pandas-docs/stable/>
14. Dask User Guide <https://docs.dask.org/en/latest/>

## **10. Методические указания для обучающихся по освоению дисциплины**

При изучении теоретического материала необходимо опираться на рабочую программу дисциплины, материалы лекций и литературу из основного списка. Кроме этого, необходимо активно работать с Интернет-источниками и пособиями других авторов, помогающими усвоить материал отдельных разделов программы.

Необходимо конспектировать лекции, помечая сложные и непонятные моменты с тем, чтобы задать вопросы лектору в конце лекции или же на консультации.

При подготовке к семинарским занятиям необходимо изучить вопросы, вынесенные на самостоятельное изучение, так как семинарские занятия предполагают их обсуждение и дискуссию по теме; кроме того, задания для самостоятельной работы необходимы для того, чтобы успешно выполнить самостоятельные задания на семинарах.

Индивидуальные задания для работы на компьютере, файлы с выполненными заданиями необходимо хранить в личной сетевой папке в компьютерной сети вуза.

**11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем**

**11. 1. Комплект лицензионного программного обеспечения:**

1. Windows, Microsoft Office.
2. Антивирус Kaspersky
3. Дистрибутив Python Anaconda

**11.2. Современные профессиональные базы данных и информационные справочные системы**

1. Информационно-правовая система «Гарант»
2. Информационно-правовая система «Консультант Плюс»
3. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
4. Система комплексного раскрытия информации «СКРИН» - <http://www.skrin.ru/>

**11.3. Сертифицированные программные и аппаратные средства защиты информации**

- не используются

**12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.**

Для проведения лекций и практических занятий необходима аудитория, оснащенная проектором и компьютерами с постоянным подключением к сети Интернет.