МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ:
Проректор по учебной работе,
качеству образования – первый
проректор
_____Хагуров Т.А.
подпись
« 29 » августа 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
Б1.В.12 «Технологии обработки больших данных»

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность «Современные методы машинного обучения и компьютерного зрения»

Форма обучения очная

Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Технологии обработки больших данных» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 01.03.02 Прикладная математика и информатика

Программу составил(а):

Приходько Татьяна Александровна, доцент, к. т. н.

ФИО, должность, ученая степень, ученое

подпись

Рабочая программа дисциплины утверждена на заседании центра искусственного интеллекта
протокол № 01 «28» августа 2025 г.
Руководитель центра ИИ Коваленко А.В.

подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики
протокол № 01 «28» августа 2025 г.
Председатель УМК факультета Коваленко А.В.

подпись

Рецензенты:
Мостовой Евгений Викторович, генеральный директор ООО «Портал-Юг», e-mail: mostovoy@portal-yug.ru

Луценко Евгений Вениаминович, доктор экономических наук, кандидат технических наук, профессор кафедры компьютерных технологий и систем Федерального государственного бюджетное образовательное учреждение высшего образования «Кубанский государственный аграрный университет имени И.Т. Трубилина», e-mail: prof.lutsenko@gmail.com

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Цель дисциплины - Изучение принципов обработки больших данных, технологий распределенных вычислений, облачных платформ и инструментов анализа данных.

1.2 Задачи дисциплины

- Изучение архитектурных решений для работы с Big Data.
- Освоение методов обработки структурированных и неструктурированных данных.
- Применение распределенных вычислений (Hadoop, Spark).
- Разработка алгоритмов анализа данных в распределенных средах.
- Использование облачных платформ для обработки больших данных.

Требования к знаниям и навыкам:

- Умение работать с распределенными системами (Hadoop, Spark).
- Опыт обработки данных в облачных средах (Yandex Cloud).
- Навыки анализа данных с помощью Python (Pandas, PySpark, Dask).
- Понимание архитектуры Big Data-решений.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Технологии обработки больших данных» относится к Блок 1 Дисциплины, часть, формируемая участниками образовательного процесса.

Дисциплина изучается в 7-м семестре. Для успешного освоения необходимы знания, полученные в дисциплинах: «Алгебра и введение в тензорный анализ», «Теория вероятностей и математическая статистка», «Подготовка данных машинного обучения», «Технологии управления данными NoSQL», «Многомерный статистический анализ», и «Машинное обучение», «Программирование».

Преподавание ведется в виде лекций и лабораторных занятий с использованием интерактивных методов. Лабораторные работы направлены на практическое освоение методов и инструментов классификации на реальных данных.

Дисциплина формирует компетенции, необходимые для выполнения выпускной квалификационной работы и профессиональной деятельности в области вычислительных технологий.

1.4 Профессиональные роли в структуре образовательной программы

Роль 1: **Data Engineer (Инженер по данным)**

Задачи:

1. Проектирование и построение ETL-процессов
2. Создание и оптимизация хранилищ данных
3. Обеспечение качества и доступности данных
4. Настройка инфраструктуры для обработки больших данных
5. Интеграция разрозненных источников данных
6. Работа с данными в области природопользования, медицины, связи и телекоммуникаций

Роль 2: **ML Engineer (Инженер МО)**

Задачи:

1. Реализация ML-моделей в продуктивных системах
2. Оптимизация производительности и масштабирование моделей

3. Разработка ML-пайплайнов и автоматизация процессов
4. Мониторинг качества моделей в продуктиве
5. Интеграция ML-решений с бизнес-приложениями

Роль 3: **MLOps (Специалист по эксплуатации ИИ)**

Задачи:

1 Автоматизация процессов обучения и развертывания моделей
2. Мониторинг производительности ML-систем
3. Управление версиями моделей и данных
4. Обеспечение CI/CD для ML-проектов
5. Оптимизация вычислительных ресурсов

1.5 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

BD-3	**Способен организовывать хранения данных, выбирая адекватные технологические решения**
BD-3.1	Разрабатывает, отлаживает и тестирует прикладные решения с элементами ИИ с применением различных технологий хранения структурированных данных, оценивает качество. Пишет аналитические запросы к данным и анализирует план запроса. Умеет создавать представления, хранимые процедуры, функции и триггеры. Знание типов СУБД: реляционные, NoSQL, колоночные, документные - Архитектура распределенных систем хранения - Принципы ACID, CAP-теорема - Методы индексации и партиционирования - Принципы транзакционности Уметь: Выбирать тип СУБД под задачи ИИ - Проектировать схемы данных для ML-моделей - Анализировать и оптимизировать планы запросов - Разрабатывать сложные аналитические запросы - Создавать database objects (views, procedures, functions) Владение SQL (оконные функции, CTE, сложные джойны) - Навык чтения explain plan - Оптимизация запросов через индексы - Создание хранимых процедур для ETL - Работа с триггерами для поддержания целостности
BD-4	**Способен применять различные модели и (или) технологии обработки данных**
BD-4.1	Осуществляет выбор технологий обработки больших данных, приемлемых для создания прикладной системы ИИ с заданными требованиями

	Способен организовывать распределенное хранилище и параллельную обработку на базе современных технологий (Hadoop, Spark) больших данных: Знать: Архитектуру Hadoop ecosystem (HDFS, YARN) - Модели вычислений: MapReduce, DAG - Принципы RDD и DataFrame в Spark – принципы Стриминговой обработки vs batch processing - Паттерны Lambda/Kappa архитектур Уметь: Выбирать стек технологий под задачи ИИ - Проектировать распределенные ETL-пайплайны - Оптимизировать производительность Spark-приложений - Организовывать шардирование и репликацию данных Владеть PySpark API - Настройка и администрирование Hadoop/Spark кластеров - Оптимизация через партиционирование, кэширование - Мониторинг производительности распределенных систем
BD-5	**Способен применять технологии организации инфраструктуры БД**
BD-5.1	Осуществляет выбор направления вспомогательных технологических решений для формирования единого стека работы с большими данными для решения поставленной задачи. Руководит проектами по организации инфраструктуры БД: Знать: принципы построения data lake, data warehouse - Методы оркестрации пайплайнов (Airflow, Prefect) - CI/CD для data projects - Мониторинг и observability в data-системах - Методологии управления data projects. Уметь: Формировать единый технологический стек - Управлять жизненным циклом data infrastructure - Выбирать инструменты мониторинга и оркестрации - Оценивать TCO инфраструктурных решений. Владеть навыками проектного управления в data-проектах - Составление ТЗ на инфраструктуру - Ведение технической документации - Оценка рисков инфраструктурных решений.
ML-1	**Способен применять знания об истории развития и трендах современного ИИ для формулирования корректных постановок задач и поиска перспективных способов решения проблем с помощью ИИ**
ML-1.2	Определяет тенденции развития, оценивает новизну и практическую значимость своих решений с точки зрения современного искусственного интеллекта. Проектирует и внедряет комплексные пайплайны предварительной обработки данных с использованием современных методов ИИ, автоматизации и feature engineering в различных предметных областях. Знать: современные архитектуры ML-моделей (трансформеры, GAN, RL) - Методы feature engineering и feature selection - MLOps принципы и best practices - Современные фреймворки

	автоматического ML - Тренды в области ИИ (LLM, мультимодальные модели) Уметь: Проектировать end-to-end ML пайплайны - Применять автоматизированный feature engineering - Оценивать бизнес-ценность ML-решений - Адаптировать state-of-the-art подходы под задачи - Владение MLflow, Kubeflow - Создание воспроизводимых ML-экспериментов - Автоматизация пайплайнов предобработки - A/B тестирование ML-моделей
PL-1	**Способен применять язык программирования Python для решения задач в области ИИ**
PL-1.3	Разрабатывает и поддерживает системы обработки больших данных различной степени сложности. Способен строить архитектуру вычислений с использованием cloud-native инструментов, в том числе бессерверных решений (Yandex Cloud Functions): Знает: архитектурные паттерны big data систем - Принципы serverless computing - Cloud-native подходы (контейнеризация, orchestration) - Асинхронное программирование в Python - Мониторинг и отладка распределенных систем. Умеет: Проектировать масштабируемые data-приложения - Использовать бессерверные архитектуры для ETL - Оптимизировать производительность Python-кода - Интегрировать различные cloud-сервисы. Владеет – навыками разработки на PySpark, Dask, Ray - Созданием и деплем cloud functions - Контейнеризацией приложений (Docker) - Настройкой автоматического масштабирования - Оптимизация costs в cloud-среде.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 4 зач. ед. (144 часов), их распределение по видам работ представлено в таблице

Виды работ	Всего часов	Форма обучения очная
		7 семестр (часы)
Контактная работа, в том числе:	**70.3**	**70.3**
Аудиторные занятия (всего):	**68**	**68**
занятия лекционного типа	34	34
лабораторные занятия	34	34
практические занятия	-	-
семинарские занятия	-	-
Иная контактная работа:	**4,3**	**4,3**
Контроль самостоятельной работы (КСР)	4	4
Промежуточная аттестация (ИКР)	0,3	0,3
Самостоятельная работа, в том числе:	**36**	**36**
Курсовая работа/проект (КР/КП) (подготовка)	-	-
Контрольная работа	-	-
Расчётно-графическая работа (РГР) (подготовка)	12	12
Выполнение индивидуальных заданий по подготовке рефератов, сообщений, презентаций	8	8

Самостоятельная проработка и материала учебников и учебных пособий, подготовка к лабораторным занятиям		10	10
Подготовка к текущему контролю		6	6
Контроль:			
Подготовка к экзамену		35.7	35.7
Общая трудоемкость	**час.**	**144**	**144**
	в том числе контактная работа	**70.3**	**70.3**
	зач. ед	**4**	**4**

2.2 Структура дисциплины

№	Наименование разделов (тем)	Количество часов			
		Всего	Аудиторная работа		Внеаудиторная работа
			Л	ЛР	СРС
1	**2**	**3**	**4**	**6**	**7**
1.	Введение в Big Data. Проблематика и базовые концепции.	6	2	2	2
2.	Распределенная файловая система HDFS и объектное хранение. Hadoop и экосистема	6	2	2	2
3.	Модели вычислений. MapReduce и его эволюция.	6	2	2	2
4.	Введение в Apache Spark. Архитектура и RDD.	6	2	2	2
5.	Spark SQL и DataFrames.	6	2	2	2
6.	Оптимизация в Spark.	6	2	2	2
7.	Работа с Spark в облаке и кластерном режиме.	12	4	4	4
8.	Потоковая обработка данных со Structured Streaming.	8	2	2	4
9.	ETL-пайпланы на Spark. Best Practices..	6	2	2	2
10.	OLAP vs OLTP. Колоночные базы данных.	6	2	2	2
11.	Глубокое погружение в ClickHouse. Движки таблиц и партиционирование.	6	2	2	2
12.	Оптимизация запросов в ClickHouse.	6	2	2	2
13.	Интеграция Spark и ClickHouse.	6	2	2	2
14.	Инструменты оркестрации данных. Apache Airflow.	6	2	2	2
15	Архитектура Big Data-решений на практике: кейсы индустриальных партнеров	12	4	4	4
ИТОГО по разделам дисциплины		**104**	**34**	**34**	**36**
Контроль самостоятельной работы (КСР)		**4**			
Промежуточная аттестация (ИКР)		**0,3**			
Подготовка к текущему контролю		**35.7**			
Общая трудоемкость по дисциплине		**144**			

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля	Соответствие индикаторам компетенций
1	**2**	**3**		
1.	Введение в Big Data.	Понятие больших данных (Volume, Velocity, Variety, Veracity, Value). Области применения и	ЛР	ML-1.2

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля	Соответствие индикаторам компетенций
1	2	3		
	Проблематика и базовые концепции.	примеры использования (социальные сети, IoT, финансы, медицина). Различия между традиционными СУБД и Big Data-решениями. Эволюция подходов к обработке: от RDBMS к распределенным системам. Обзор экосистемы Hadoop (HDFS, YARN, MapReduce) и современных фреймворков (Spark, Flink). Жизненный цикл данных: сбор, ETL, хранение, анализ, визуализация. Облачные платформы для Big Data (Yandex Cloud, AWS, GCP) – обзор сервисов.		
2.	Распределенная файловая система HDFS и объектное хранение. Hadoop и экосистема	Принципы распределенных вычислений. Модели распределенных вычислений. Hadoop, Spark, Kafka Архитектура HDFS: NameNode, DataNode, принципы репликации и отказоустойчивости. Rack Awareness. Объектные хранилища (S3-совместимые): Yandex Object Storage, архитектура, преимущества перед HDFS. Команды для работы с HDFS и S3.	ЛР	BD-4.1, PL-1.3
3.	Модели вычислений. MapReduce и его эволюция.	Архитектура Hadoop (HDFS, YARN). Инструменты (Hive, Pig, HBase). Модель вычислений MapReduce. Аналоги: Apache Spark, преимущества и недостатки. Детальный разбор модели MapReduce: Map, Shuffle & Sort, Reduce. Ограничения и сложности MapReduce. Почему Spark пришел на смену MapReduce? Введение в Resilient Distributed Datasets (RDD).	ЛР	BD-4.1, BD-5.1
4.	Введение в Apache Spark. Архитектура и RDD.	RDD, DataFrame, Dataset. Оптимизация вычислений. Spark SQL и оптимизация запросов. Архитектура Spark: Driver, Executor, Cluster Manager (Standalone, YARN, Kubernetes). Концепция Resilient Distributed Datasets (RDD): свойства, lineage, трансформации, действия. Ленивые вычисления и планировщик (DAG Scheduler).	ЛР	BD-3.1, BD-4.1
5.	Spark SQL и DataFrames.	Apache Kafka: архитектура и применение. Apache Flink / Spark Streaming. Обработка событий в реальном времени. Ограничения RDD. Преимущества DataFrame API. Концепция Catalyst Optimizer и Tungsten Engine. Структура DataFrame, схема (Schema). Источники и приемники данных (Data Sources API).	ЛР	BD-4.1, PL-1.3
6.	Оптимизация в Spark.	Партиционирование данных: зачем нужно и как влияет на производительность. Стратегии соединения (Joins). Broadcast Hash Join.	ЛР	BD-5.1

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля	Соответствие индикаторам компетенций
1	2	3		
		Кэширование и персистентность данных в памяти. Анализ плана выполнения запроса (explain).		
7.	Работа с Spark в облаке и кластерном режиме.	Yandex Data Proc, Yandex Object Storage. Развертывание кластера. Развертывание Spark на YARN в Yandex Cloud (используя Yandex Data Proc). Жизненный цикл Spark-приложения: client vs cluster mode. Передача конфигураций и зависимостей (.jar файлы, Python-пакеты).Мониторинг приложений через Spark UI.	ЛР	BD-4.1, BD-5.1
8.	Потоковая обработка данных со Structured Streaming.	Классификация, кластеризация, рекомендательные системы. Принципы потоковой обработки. Модель Structured Streaming: таблица результатов, инкрементальное выполнение. Источники и стоки (sinks): Kafka, файловые системы, консоль. Окна (Windows) и водяные знаки (Watermarks).	ЛР	BD-4.1, ML-1.2
9.	ETL-пайпланы на Spark. Best Practices.	Проектирование надежных ETL-процессов. Обработка ошибок, повторные попытки. Интеграция с системами оркестрации (Apache Airflow). Паттерны использования: Lambda/ Kappa Architecture.	ЛР	BD-4.1, ML-1.2
10.	OLAP vs OLTP. Колоночные базы данных.	Различия между OLTP и OLAP системами. Принципы работы колоночных СУБД (хранение, сжатие, векторизация). Обзор рынка: ClickHouse, Apache Druid, Amazon Redshift, Google BigQuery.	ЛР	ML-1.21
11.	Глубокое погружение в ClickHouse. Движки таблиц и партиционирование.	Архитектура ClickHouse: столбцы, сжатие, индексы (запросы primary key). Движки таблиц: семейство MergeTree – основа производительности. Партиционирование и сортировка данных. Зеркалирование и шардирование (ReplicatedMergeTree).	ЛР	BD-4.1, BD-5.1
12.	Оптимизация запросов в ClickHouse.	Как работают индексы в ClickHouse (запросы primary key). Векторизованное выполнение запросов. Материализованные представления и проекции. Мониторинг и профилирование запросов.	ЛР	BD-5.1
13.	Интеграция Spark и ClickHouse.	Паттерны взаимодействия: Spark -> ClickHouse и ClickHouse -> Spark. Использование официального JDBC-коннектора. Использование специализированных коннекторов (clickhouse-spark-connector). Best Practices по вставке данных (батчи, асинхронность).	ЛР	BD-4.1, ML-1.2
14.	Инструменты оркестрации	Проблема управления ETL-пайпланами.	ЛР	ML-1.2

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля	Соответствие индикаторам компетенций
1	**2**	**3**		
	данных. Apache Airflow.	Основные концепции Airflow: DAG, Operator, Task, Scheduler. Создание и планирование пайпланов.		
15.	Архитектура Big Data-решений на практике: кейсы индустриальных партнеров	Сборка классических архитектур: Lambda, Kappa. Роль каждого компонента в стэке (Kafka, Spark, ClickHouse, Airflow). Критерии выбора технологий под конкретную бизнес-задачу. Мониторинг, логирование и обеспечение надежности всей системы.	РГР	

2.3.2 Лабораторные занятия

№	Наименование раздела (темы)	Тематика лабораторных работ	Форма текущего контроля
1.	Введение в Big Data. Проблематика и базовые концепции.	**Настройка окружения и первое знакомство с Yandex Cloud.** • Создание аккаунта в Yandex Cloud. • Знакомство с интерфейсом Yandex Cloud Console. • Создание виртуальной машины (Yandex Compute Cloud) с предустановленными Docker и JDK/Python. • Подключение к ВМ по SSH, базовые команды Linux.	Опрос по теоретическому материалу. Отчет по лабораторной работе.
2.	Распределенная файловая система HDFS и объектное хранение. Hadoop и экосистема	**Работа с системами хранения в Yandex Cloud.** • Создание бакета в Yandex Object Storage. • Загрузка, скачивание и управление данными через Console и CLI. • Развертывание HDFS в Docker-контейнере (используя docker-compose). • Практика базовых команд HDFS (hdfs dfs -put, -get, -ls).	Опрос по теоретическому материалу. Отчет по лабораторной работе.
3.	Модели вычислений. MapReduce и его эволюция.	**Реализация алгоритма WordCount на MapReduce (Java).** • Написание Mapper и Reducer классов на Java. • Сборка проекта с помощью Maven. • Запуск джобы на локально развернутом Hadoop/YARN кластере (в Docker). • Анализ логов и выходных данных.	Опрос по теоретическому материалу. Отчет по лабораторной работе.
4.	Введение в Apache Spark. Архитектура и RDD.	**Основы работы с RDD в PySpark.** • Запуск Spark Shell (PySpark) в локальном режиме. • Создание RDD из коллекции и из файла в HDFS/Object Storage. • Применение основных трансформаций (map, filter, flatMap) и действий (count, collect, take). • Реализация WordCount с использованием RDD API.	Контрольная работа №2 Проверка выполнения домашних работ.

5.	Spark SQL и DataFrames.	**Обработка структурированных данных с помощью Spark SQL.** • Создание SparkSession. • Загрузка CSV/JSON данных в DataFrame. • Выполнение запросов с использованием DataFrame API (select, filter, groupBy, agg). • Использование Spark SQL для выполнения SQL-подобных запросов.	Опрос по теоретическому материалу. Отчет по лабораторной работе.
6.	Оптимизация в Spark.	**Оптимизация Spark-приложений.** • Анализ плана выполнения сложного запроса. • Изменение стратегии партиционирования (repartition, coalesce). • Применение кэширования для итеративных операций. • Сравнение времени выполнения до и после оптимизации.	Опрос по теоретическому материалу. Отчет по лабораторной работе.
7.	Работа с Spark в облаке и кластерном режиме.	**Запуск Spark-задачи на кластере Yandex Data Proc.** • Создание кластера Data Proc. • Подготовка скрипта на PySpark и загрузка его в Object Storage. • Запуск задачи с помощью Yandex Data Proc UI и CLI. • Мониторинг выполнения задачи через Spark UI.	Опрос по теоретическому материалу. Отчет по лабораторной работе.
8.	Потоковая обработка данных со Structured Streaming.	**Обработка потока данных с Kafka и Spark Structured Streaming.** • Развертывание Kafka в Docker. • Создание продюсера, генерирующего тестовые данные. • Написание Spark-приложения, которое читает данные из Kafka, агрегирует их в скользящем окне и выводит результат. • (Опционально) Запись результатов в ClickHouse.	Опрос по теоретическому материалу. Отчет по лабораторной работе.
9.	ETL-пайпланы на Spark. Best Practices.	**Построение сквозного ETL-пайплана.** • Написание PySpark-скрипта, который: 1. Читает "сырые" JSON-данные из Object Storage. 2. Проводит очистку и валидацию (обработка пропусков, приведение типов). 3. Выполняет обогащение данных (джойн со справочником). 4. Записывает обработанные данные в колонный формат (Parquet) в другую папку Object Storage.	-//-
10.	OLAP vs OLTP. Колоночные базы данных.	**Установка и настройка ClickHouse.** • Развертывание ClickHouse в Docker. • Знакомство с CLI-клиентом clickhouse-client. • Создание первой базы данных и таблицы. Изучение основных типов данных	-//-
11.	Глубокое погружение в ClickHouse. Движки таблиц и партиционирование.	**Создание оптимизированных таблиц в ClickHouse.** • Создание таблицы с движком MergeTree, указание ключа партиционирования и первичного ключа. • Загрузка данных из CSV-файла. • Сравнение производительности запросов к одной и той же таблице с разными первичными ключами.	-//-

		• Создание реплицируемой таблицы (ReplicatedMergeTree).	
12.	Оптимизация запросов в ClickHouse.	**Анализ и оптимизация запросов в ClickHouse.** • Написание сложных аналитических запросов (оконные функции, GROUP BY с модификаторами WITH TOTALS). • Использование EXPLAIN и EXPLAIN PIPELINE для анализа плана запроса. • Создание материализованного представления для ускорения часто используемых агрегаций.	-//-
13.	Интеграция Spark и ClickHouse.	**Загрузка данных из Spark в ClickHouse.** • Написание PySpark-скрипта, который читает данные из Parquet-файлов в Object Storage. • Преобразование данных и загрузка их в таблицу ClickHouse с использованием JDBC-коннектора. • Реализация стратегии "upsert" через использование движка `ReplacingMergeTree`.	-//-
14.	Инструменты оркестрации данных. Apache Airflow.	**Создание DAG в Apache Airflow.** • Развертывание Airflow в Docker. • Написание простого DAG, который: 1. Запускает Spark-задачу на Data Proc. 2. Ожидает ее успешного завершения. 3. Отправляет уведомление (в консоль или логи) о результате.	-//-
15.	Архитектура Big Data-решений на практике: кейсы индустриальных партнеров	**Финальный проект. Сквозной пайплайн обработки данных.** • **Цель:** Реализовать миниатюрную, но полноценную Big Data-систему в Yandex Cloud. • **Задача:** 1. **Источник данных:** Имитация потока событий (клики, просмотры) с помощью скрипта, пишущего в Kafka/Yandex Message Queue. 2. **Потоковая обработка:** Spark Structured Streaming приложение (на Data Proc) читает из Kafka, агрегирует данные в реальном времени (например, по 5-минутным окнам) и пишет сырые и агрегированные данные в Object Storage (Parquet). 3. **ETL и загрузка в OLAP:** Отдельное Spark-приложение (запускаемое по расписанию через Airflow) читает сырые данные из Parquet, проводит дополнительную очистку и обогащение, и загружает результат в ClickHouse. 4. **Аналитика:** Написание сложных аналитических запросов в ClickHouse для получения бизнес-метрик (DAU, LTV, топ товаров и т.д.). • **Результат:** Отчет с кодом, скриншотами работающих систем (Spark UI, Airflow DAGs, результаты запросов в ClickHouse) и описанием архитектуры.	Защита финального проекта

2.3.4 Примерная тематика курсовых работ (проектов)

Не предусмотрены учебным планом

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Проработка и повторение лекционного материала, материала учебной и научной литературы, подготовка к семинарским занятиям	Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры вычислительных технологий факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №7 от 07.05.2025 г.
2	Подготовка к лабораторным занятиям	Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры вычислительных технологий факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №7 от 07.05.2025 г.
3	Подготовка к решению задач и тестов	Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры вычислительных технологий факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №7 от 07.05.2025 г.
4	Подготовка к текущему контролю	Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры вычислительных технологий факультета компьютерных технологий и прикладной математики ФГБОУ ВО «КубГУ», протокол №7 от 07.05.2025 г.

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа,

– в форме аудиофайла,

– в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа,

– в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии, применяемые при освоении дисциплины (модуля)

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с

использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

1. Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.
2. Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.
3. Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.
4. Информационно-коммуникационные технологии (ИКТ) - расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:
5. Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.
6. Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.
7. Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.
8. Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.
9. Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.
10. Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.
11. Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.
12. Комплексное использование в учебном процессе всех вышеназванных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

13. работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;
14. проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;
15. анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

16. развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

При проведении лабораторных занятий участники закрепляют пройденный материал путем обсуждения вопросов, требующих особого внимания и понимания, отвечают на вопросы преподавателя и других слушателей, осуществляют решения тестов, направленных на повторение лекционного материала и нормативных документов по изучаемой тематике, выполняют решение задач, которые способствуют развитию практических навыков в области изучаемой дисциплины.

В число видов работы, выполняемой слушателями самостоятельно, входят:

1) поиск и изучение литературы по рассматриваемой теме;

2) поиск и анализ научных статей, монографий по рассматриваемой теме.

Интерактивные образовательные технологии, используемые в аудиторных занятиях: при реализации различных видов учебной работы (лекций и практических занятий) используются следующие образовательные технологии: дискуссии, презентации, конференции. В сочетании с внеаудиторной работой они создают дополнительные условия формирования и развития требуемых компетенций обучающихся, поскольку позволяют обеспечить активное взаимодействие всех участников. Эти методы способствуют личностно-ориентированному подходу.

Все перечисленные виды и формы учебной работы и текущего контроля направлены на формирование у обучающихся профессиональных компетенций, предусмотренных при планировании результатов обучения по дисциплине и соотнесенных с планируемыми результатами освоения образовательной программы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты и устанавливается особый порядок освоения указанной дисциплины. В образовательном процессе используются социально-активные и рефлексивные методы обучения, технологии социально-культурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе.

Вышеозначенные образовательные технологии дают наиболее эффективные результаты освоения дисциплины с позиций актуализации содержания темы занятия, выработки продуктивного мышления, терминологической грамотности и компетентности обучаемого в аспекте социально направленной позиции будущего бакалавра, и мотивации к инициативному и творческому освоению учебного материала.

4. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «название дисциплины».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме отчетов по лабораторным работам и **промежуточной аттестации** в форме вопросов и заданий к экзамену.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
	Введение в Big Data. Проблематика и базовые концепции.	BD-3, BD-4	*Лабораторная работа №1*	*Вопросы к экзамену*
	Распределенная файловая система HDFS и объектное хранение. Hadoop и экосистема	BD-3, BD-4, BD-5	*Лабораторная работа №2*	*Вопросы к экзамену*
	Модели вычислений. MapReduce и его эволюция.	BD-3, BD-4, BD-5, PL-1	*Лабораторная работа №3*	*Вопросы к экзамену*
	Введение в Apache Spark. Архитектура и RDD.	BD-3, BD-4, BD-5, PL-1	*Лабораторная работа №4*	*Вопросы к экзамену*
	Spark SQL и DataFrames.	BD-3, BD-4, BD-5, PL-1	*Лабораторная работа №5*	*Вопросы к экзамену*
	Оптимизация в Spark.	BD-3, BD-4, BD-5, PL-1	*Лабораторная работа №6*	*Вопросы к экзамену*

	Работа с Spark в облаке и кластерном режиме.	BD-3, BD-4, BD-5, PL-1	*Лабораторная работа №7*	*Вопросы к экзамену*
	Потоковая обработка данных со Structured Streaming.	BD-**3**, BD-**4, BD-5, PL-1**	*Лабораторная работа №8*	*Вопросы к экзамену*
9	ETL-пайпланы на Spark. Best Practices.	BD-3, BD-4, BD-5, PL-1	*Лабораторная работа №9*	*Вопросы к экзамену*
10	OLAP vs OLTP. Колоночные базы данных.	BD-3, BD-4, BD-5, PL-1, ML-1	*Лабораторная работа №10*	*Вопросы к экзамену*
11	Глубокое погружение в ClickHouse. Движки таблиц и партиционирование.	BD-3, BD-4, BD-5, PL-1, ML-1	*Лабораторная работа №11*	*Вопросы к экзамену*
12	Оптимизация запросов в ClickHouse.	BD-3, BD-4, BD-5, PL-1, ML-1	*Лабораторная работа №12*	*Вопросы к экзамену*
13	Интеграция Spark и ClickHouse.	BD-3, BD-4, BD-5, PL-1, ML-1	*Лабораторная работа №13*	*Вопросы к экзамену*
14	Инструменты оркестрации данных. Apache Airflow.	BD-3, BD-4, BD-5, PL-1, ML-1	*Лабораторная работа №14*	*Вопросы к экзамену*
15	Архитектура Big Data-решений на практике: кейсы индустриальных партнеров	BD-3, BD-4, BD-5, PL-1, ML-1	*Лабораторная работа №15-16 -финальный проект*	*Вопросы к экзамену*

Показатели, критерии и шкала оценки сформированных компетенций

BD-3	**Способен организовывать хранения данных, выбирая адекватные технологические решения**
BD-3.1	Разрабатывает, отлаживает и тестирует прикладные решения с элементами ИИ с применением различных технологий хранения структурированных данных, оценивает качество. Пишет аналитические запросы к данным и анализирует план запроса. Умеет создавать представления, хранимые процедуры, функции и триггеры. Знание типов СУБД: реляционные, NoSQL, колоночные, документные - Архитектура распределенных систем хранения - Принципы ACID, CAP-теорема - Методы индексации и партиционирования - Принципы транзакционности Уметь: Выбирать тип СУБД под задачи ИИ - Проектировать схемы данных для ML-моделей - Анализировать и оптимизировать планы запросов - Разрабатывать сложные аналитические запросы - Создавать database objects (views, procedures, functions) Владение SQL (оконные функции, CTE, сложные джойны) - Навык чтения explain plan - Оптимизация запросов через индексы - Создание хранимых процедур для ETL - Работа с триггерами для поддержания целостности

BD-4	**Способен применять различные модели и (или) технологии обработки данных**
BD-4.1	Осуществляет выбор технологий обработки больших данных, приемлемых для создания прикладной системы ИИ с заданными требованиями Способен организовывать распределенное хранилище и параллельную обработку на базе современных технологий (Hadoop, Spark) больших данных: Знать: Архитектуру Hadoop ecosystem (HDFS, YARN) - Модели вычислений: MapReduce, DAG - Принципы RDD и DataFrame в Spark – принципы Стриминговой обработки vs batch processing - Паттерны Lambda/Kappa архитектур Уметь: Выбирать стек технологий под задачи ИИ - Проектировать распределенные ETL-пайплайны - Оптимизировать производительность Spark-приложений - Организовывать шардирование и репликацию данных Владеть PySpark API - Настройка и администрирование Hadoop/Spark кластеров - Оптимизация через партиционирование, кэширование - Мониторинг производительности распределенных систем
BD-5	**Способен применять технологии организации инфраструктуры БД**
BD-5.1	Осуществляет выбор направления вспомогательных технологических решений для формирования единого стека работы с большими данными для решения поставленной задачи. Руководит проектами по организации инфраструктуры БД: Знать: принципы построения data lake, data warehouse - Методы оркестрации пайплайнов (Airflow, Prefect) - CI/CD для data projects - Мониторинг и observability в data-системах - Методологии управления data projects. Уметь: Формировать единый технологический стек - Управлять жизненным циклом data infrastructure - Выбирать инструменты мониторинга и оркестрации - Оценивать TCO инфраструктурных решений. Владеть навыками проектного управления в data-проектах - Составление ТЗ на инфраструктуру - Ведение технической документации - Оценка рисков инфраструктурных решений.
ML-1	**Способен применять знания об истории развития и трендах современного ИИ для формулирования корректных постановок задач и поиска перспективных способов решения проблем с помощью ИИ**
ML-1.2	Определяет тенденции развития, оценивает новизну и практическую значимость своих решений с точки зрения современного искусственного интеллекта. Проектирует и внедряет комплексные пайплайны предварительной обработки данных с использованием современных методов ИИ, автоматизации и feature engineering в различных предметных областях. Знать: современные архитектуры ML-моделей (трансформеры, GAN, RL) - Методы feature engineering и feature selection - MLOps принципы и best practices - Современные фреймворки автоматического ML - Тренды в области ИИ (LLM, мультимодальные модели) Уметь: Проектировать end-to-end ML пайплайны - Применять автоматизированный feature engineering - Оценивать бизнес-ценность ML-решений - Адаптировать state-of-the-art подходы под задачи

	- Владение MLflow, Kubeflow - Создание воспроизводимых ML-экспериментов - Автоматизация пайплайнов предобработки - A/B тестирование ML-моделей
PL-1	**Способен применять язык программирования Python для решения задач в области ИИ**
PL-1.3	Разрабатывает и поддерживает системы обработки больших данных различной степени сложности. Способен строить архитектуру вычислений с использованием cloud-native инструментов, в том числе бессерверных решений (Yandex Cloud Functions): Знает: архитектурные паттерны big data систем - Принципы serverless computing - Cloud-native подходы (контейнеризация, orchestration) - Асинхронное программирование в Python - Мониторинг и отладка распределенных систем. Умеет: Проектировать масштабируемые data-приложения - Использовать бессерверные архитектуры для ETL - Оптимизировать производительность Python-кода - Интегрировать различные cloud-сервисы. Владеет – навыками разработки на PySpark, Dask, Ray - Созданием и деплем cloud functions - Контейнеризацией приложений (Docker) - Настройкой автоматического масштабирования - Оптимизация costs в cloud-среде.

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

4.1.2.Примеры лабораторных работ и контрольных заданий по разделам учебной дисциплины

Примеры лабораторных работ

Лабораторная работа: Оптимизация аналитических запросов в колоночной СУБД

Цель: Освоить techniques оптимизации сложных аналитических запросов в ClickHouse, анализ планов выполнения и создание оптимизированных структур хранения.

Компетенции: BD-3.1, BD-5.1, PL-1.3

Теоретическая часть

Задача: Компания собирает данные о кликах пользователей (10+ млн записей). Необходимо оптимизировать запросы для аналитики в реальном времени.

Исходная неоптимизированная таблица:

sql

```
CREATE TABLE default.clicks (
    user_id UInt32,
    session_id String,
    page_url String,
    click_element String,
```

```
    event_time DateTime,
    device_type String,
    country String,
    region String,
    duration_sec UInt32
) ENGINE = MergeTree()
ORDER BY (user_id, event_time);
```

Этап 1: Анализ проблем производительности
Задание 1.1:

- Загрузите тестовые данные (10 млн записей)
- Выполните запрос, замерьте время выполнения
- Проанализируйте EXPLAIN PIPELINE, определите узкие места

Этап 2: Редизайн структуры данных
Этап 3: PySpark обработка на Data Proc
Этап 4: Оркестрация через Yandex Data Proc

Критерии оценки:

- End-to-end работоспособность пайплайна
- Обработка ошибок и валидация данных
- Эффективное использование ресурсов cloud
- Мониторинг и логирование всех этапов

Лабораторная работа: Основы работы с RDD в PySpark
Цель:Освоить базовые операции с Resilient Distributed Datasets (RDD) - фундаментальной структурой данных в Apache Spark. Научиться создавать RDD из различных источников и применять основные трансформации и действия для обработки данных.
Теоретическая справка
RDD (Resilient Distributed Dataset) - неизменяемая распределенная коллекция объектов, являющаяся основой Spark. Ключевые характеристики:
Распределенность: данные разделены на партиции и распределены по кластеру
Отказоустойчивость: lineage (родословная) позволяет восстанавливать потерянные партиции
Ленивые вычисления: трансформации выполняются только при вызове действий
Задания
Часть 1: Настройка окружения и создание RDD

Задание 1.1: Запуск PySpark в локальном режиме

```
python
# Запустите PySpark shell
pyspark --master local[2]

# Или создайте SparkContext в Python скрипте
from pyspark import SparkContext, SparkConf

conf = SparkConf().setAppName("RDD_Basics").setMaster("local[2]")
sc = SparkContext(conf=conf)
```

Задание 1.2: Создание RDD из коллекции
python

```
# Создайте RDD из списка чисел
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
numbers_rdd = sc.parallelize(numbers)

# Создайте RDD из списка строк
text_data = ["Hello World", "Apache Spark", "Big Data Processing", "Python Programming"]
text_rdd = sc.parallelize(text_data)
```

Задание 1.3: Создание RDD из файла
python
```
# Создайте текстовый файл для работы
# sample_text.txt содержимое:
# Apache Spark is a unified analytics engine
# for large-scale data processing
# It provides high-level APIs in Java Scala Python and R

file_rdd = sc.textFile("sample_text.txt")
```

Часть 2: Трансформации (Transformations)
Задание 2.1: Операция map
python
```
# Увеличьте каждое число в 2 раза
doubled_numbers = numbers_rdd.map(lambda x: x * 2)

# Преобразуйте строки в верхний регистр
uppercase_text = text_rdd.map(lambda x: x.upper())
```

Задание 2.2: Операция filter
python
```
# Отфильтруйте четные числа
even_numbers = numbers_rdd.filter(lambda x: x % 2 == 0)

# Отфильтруйте строки, содержащие слово "Spark"
spark_lines = text_rdd.filter(lambda x: "Spark" in x)
```

Задание 2.3: Операция flatMap
python
```
# Разбейте строки на отдельные слова
words_rdd = text_rdd.flatMap(lambda x: x.split(" "))

# Для файла: разбейте каждую строку на слова и преобразуйте в нижний регистр
file_words = file_rdd.flatMap(lambda line: line.lower().split(" "))
```

Часть 3: Действия (Actions)

Задание 3.1: Базовые действия
python
```
# Подсчет элементов
total_numbers = numbers_rdd.count()
total_words = words_rdd.count()

# Получение всех элементов (осторожно с большими данными!)
```

```
all_numbers = numbers_rdd.collect()
first_three = words_rdd.take(3)

# Получение первых n элементов
first_five_words = words_rdd.take(5)
```

Задание 3.2: Дополнительные действия
python
```
# Подсчет по каждому элементу
word_counts = words_rdd.countByValue()

# Сумма всех чисел
total_sum = numbers_rdd.reduce(lambda a, b: a + b)

# Максимальное значение
max_number = numbers_rdd.reduce(lambda a, b: a if a > b else b)
```

Часть 4: Реализация WordCount

Задание 4.1: Классический WordCount
python
```
# Реализуйте WordCount для файла
word_count = (file_rdd
        .flatMap(lambda line: line.lower().split(" "))
        .map(lambda word: (word, 1))
        .reduceByKey(lambda a, b: a + b))

# Отсортируйте по убыванию частоты
sorted_word_count = word_count.sortBy(lambda x: x[1], ascending=False)
```

Пример задания и проверяемые индикаторы

Задание:

- Загрузить граф социальной сети (например, данные из VK API или датасета LiveJournal).
- Найти всех друзей пользователя (1 уровень связей).
- Применить PageRank для определения влиятельных узлов.
- Визуализировать граф (например, с помощью NetworkX или Gephi).

Проверяемые индикаторы:

- BD 4.1 (работа с GraphX).
- BD 5.1 (Оптимизация запросов, развертывание кластеров).

Вывод

Задание «Анализ социального графа» в первую очередь проверяет:

- BD 4.1 (работа с распределёнными графами в Spark).

Дополнительно могут затрагиваться BD 5.1 (инфраструктура, нестандартные данные).

Для максимального балла студент должен показать не только техническое выполнение, но и интерпретацию результатов (например, как выявленные связи можно использовать в рекомендательной системе).

Экзаменационные материалы для промежуточной аттестации (экзамен)

Вопросы для подготовки к экзамену

1. Что такое Big Data? Основные характеристики (3V).
2. Сравнение Hadoop и Spark.
3. Принципы работы HDFS.
4. Архитектура MapReduce.
5. Spark RDD vs DataFrame.
6. Какие существуют методы оптимизации в Spark?
7. Особенности потоковой обработки данных.
8. Разница между Kafka и RabbitMQ.
9. Типы NoSQL баз данных.
10. Особенности колоночных СУБД (Cassandra).
11. Как работает шардинг в распределенных БД?
12. Облачные решения для Big Data (AWS, GCP).
13. Методы машинного обучения для больших данных.
14. Как работает рекомендательная система на основе коллаборативной фильтрации?
15. Инструменты визуализации больших данных.
16. Методы обеспечения безопасности в Big Data.
17. Что такое GDPR и как он влияет на обработку данных?
18. Применение Big Data в финансах.
19. Как работает алгоритм PageRank?
20. Проблемы этики при работе с персональными данными.
21. Разница между Lambda и Kappa архитектурами.
22. Как работает Elasticsearch?
23. Методы борьбы с перекосом данных (skewness).
24. Что такое Data Lake?
25. Как устроена графовая БД (Neo4j)?

Примерные практические задания (к экзамену)

1. Написать MapReduce-программу для подсчета слов.
2. Создать DataFrame в Spark и выполнить агрегацию.
3. Настроить Kafka producer и consumer.
4. Написать SQL-запрос в Hive.
5. Построить гистограмму распределения данных.
6. Обучить модель классификации в MLlib.
7. Загрузить данные в Google BigQuery.
8. Оптимизировать Spark-запрос.
9. Построить граф в Neo4j.
10. Настроить мониторинг логов в Kibana.

Перечень компетенций (части компетенции), проверяемых оценочным средством **BD-3.1, BD-4.1, BD-5.1, ML-1.2, PL-1.3** (см. таблица **Структура оценочных средств для текущей и промежуточной аттестации**).

4.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Методические рекомендации, определяющие процедуры оценивания лабораторной работы.

Текущая аттестация проводится по лабораторным работам, и может принести в копилку максимум **40 баллов**. В соответствие с критериями оценки выполнения лабораторных работ Пороговый уровень 10 баллов, базовый 20 баллов, продвинутый уровень 40 баллов. Текущий бал определяется усреднением баллов по всем (16) лабораторным работам и как результат будет принадлежать отрезку от 0 до 40 баллов.

Промежуточной аттестацией по дисциплине «Технологии обработки больших данных» является экзамен. Максимальная оценка, которую можно получить в качестве оценки экзамена 60 баллов.

Методические рекомендации, определяющие процедуры оценивания на экзамене.

В экзаменационном билете два теоретических вопроса и одно практическое задание. Каждый раздел билета оценивается в 20 баллов.

Пример: В билете вопрос №2 из списка экзаменационных вопросов. Этому вопросу соответствует два индикатора компетенций BD 3.1 и BD 4.1, предположим, что по индикатору BD 3.1 достигнут пороговый уровень (10 баллов), по индикатору BD 4.1 достигнут продвинутый уровень (20 баллов), тогда ответ данного вопроса в билете будет оценен в 15 баллов. Аналогично второй вопрос.

По теоретическому материалу балл определяется усреднением уровня усвоения компетенций по индикаторам соответствующего раздела.

Практическое задание оценивается следующим образом:

- 18-20 баллов: Код работает корректно, использованы оптимальные методы, решение эффективно.
- 11-17 баллов: Код работает, но есть недочёты в оптимизации.
- 1-10 балла: Код требует доработок, но логика верна.
- 0: Код нерабочий или решение неверное.

Экзаменационная оценка в баллах формируется простым суммированием оценок (баллов) за разделы экзамена и оценки (баллы) текущей аттестации за работу в семестре (лабораторные работы).

В стандартной форме экзаменационная оценка определяется следующим соответствием:

0 – 49 баллов «неудовлетворительно»;

50 – 70 баллов «удовлетворительно»;

71 – 85 баллов «хорошо»;

86 – 100 баллов «отлично».

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,
– в форме электронного документа.
Для лиц с нарушениями опорно-двигательного аппарата:
– в печатной форме,
– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4.3 Методические указания по организации лабораторных работ по дисциплине "Технологии обработки больших данных"

1. Общие сведения

Образовательная программа: «Современные методы машинного обучения и компьютерного зрения», Дисциплина "Технологии обработки больших данных".
Вид обеспечения: Проведение лабораторных работ.
Условия применения:
Для успешного выполнения лабораторных работ требуется:

Программное обеспечение:

Python (PySpark, Pandas, Dash/Plotly, MLlib, Kafka, GraphX)
Jupyter Notebook / Google Colab
Apache Spark, Hadoop (Yandex Data Proc)
MongoDB, Kafka
Yandex Cloud CLI и SDK

Аппаратное обеспечение:

Доступ к облачным ресурсам (Yandex Cloud)
Достаточные вычислительные мощности для обработки данных (виртуальные машины, кластеры)

Облачная инфраструктура:

Yandex Cloud (Compute Cloud, Object Storage, Managed MongoDB, Yandex Data Proc, Yandex Metrica API, SpeechKit)
Доступ к S3-совместимому хранилищу.

2. Цели, задачи и ожидаемые результаты

Цель:

- Закрепить теоретические знания на практике.
- Отработать навыки работы с современными инструментами обработки больших данных (Spark, NoSQL, потоковая обработка, ML).
- Обеспечить понимание облачных технологий и распределенных вычислений.
- Подготовить студентов к решению реальных задач в индустрии (ETL, аналитика, визуализация).

Задачи:

1. *Организация инфраструктуры:*
   Настройка облачного окружения (Yandex Cloud, Colab).
   Работа с виртуальными машинами, кластерами (Data Proc).
2. *Обработка данных:*
   Освоение PySpark (RDD, DataFrame).
   Работа с NoSQL (MongoDB).
   Потоковая обработка (Kafka + Spark Streaming).
3. *Аналитика и ML:*
   Применение MLlib для классификации.
   Прогнозирование временных рядов (ARIMA).

4. *Визуализация и интеграция:*
   Построение дашбордов (Dash/Plotly).
   Работа с API (Yandex Metrica, SpeechKit).
5. *Оптимизация и масштабирование:*
   Кэширование, партиционирование в Spark.
   Развертывание кластера (Yandex Data Proc).

Ожидаемые результаты:
После выполнения лабораторных работ студенты смогут:
- Самостоятельно настраивать облачную инфраструктуру для обработки данных.
- Применять Spark для ETL, агрегации и анализа больших данных.
- Работать с NoSQL и потоковыми данными.
- Строить ML-модели и визуализировать результаты.
- Оптимизировать запросы и развертывать распределенные системы.
- Интегрироваться с внешними API (Yandex Cloud, SpeechKit).

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

5.1 Основная литература:

1. Бутаков, Н. А. Обработка больших данных с Apache Spark : учебно-методическое пособие : [16+] / Н. А. Бутаков, М. В. Петров, Д. Насонов. – Санкт-Петербург : Университет ИТМО, 2019. – 52 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=566771 (дата обращения: 02.11.2025). – Библиогр. в кн. – Текст : электронный.
2. Параскевов, А. В. Большие данные : учебник : [12+] / А. В. Параскевов, А. Э. Сергеев. – Москва ; Вологда : Инфра-Инженерия, 2024. – 148 с. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=725632 (дата обращения: 02.11.2025). – Библиогр. в кн. – ISBN 978-5-9729-2120-1. – Текст : электронный.
3. Шкодина, Т. А. Статистический анализ данных в Python : лабораторный практикум : учебное пособие для направления 01.03.05 «Статистика» : учебное пособие : [16+] / Т. А. Шкодина, С. М. Щербаков ; Ростовский государственный экономический университет (РИНХ). – Ростов-на-Дону : Издательско-полиграфический комплекс РГЭУ (РИНХ), 2024. – 104 с. : ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=718683 (дата обращения: 02.11.2025). – Библиогр.: с. 100. – ISBN 978-5-7972-3232-2. – Текст : электронный.
4. Машинное обучение : учебник : [16+] / Е. Ю. Бутырский, В. В. Цехановский, Н. А. Жукова [и др.]. – Москва : Директ-Медиа, 2023. – 368 с. : ил., табл., схем., граф. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=701807 (дата обращения: 02.11.2025). – Библиогр. в кн. – ISBN 978-5-4499-3778-0. – DOI 10.23681/701807. – Текст : электронный.
5. Кревецкий, А. В. Основы технологий искусственного интеллекта : учебное пособие : [16+] / А. В. Кревецкий, Ю. А. Ипатов, Н. И. Роженцова ; под общ. ред. А. В. Кревецкого ; Поволжский государственный технологический университет. – Йошкар-Ола : Поволжский государственный технологический университет, 2023. – 272 с. : ил., табл., схем. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=714624 (дата обращения: 02.11.2025). – Библиогр.: с. 264-267. – ISBN 978-5-8158-2358-7. – Текст : электронный.
6. Уржумов, Д. В. Системы распознавания образов. Компьютерное зрение : практикум : [16+] / Д. В. Уржумов, А. В. Кревецкий ; Поволжский государственный технологический

университет. – Йошкар-Ола : Поволжский государственный технологический университет, 2024. – 36 с. : ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=718735 (дата обращения: 02.11.2025). – Библиогр.: с. 34. – ISBN 978-5-8158-2386-0. – Текст : электронный.

5.2 Дополнительная литература:

1. Официальная документация Apache Spark, Kafka.
2. Yandex Cloud – руководства по Data Proc.
3. Харрисон Д. – "Python и анализ данных".
4. Колдаев, В. Д. Структуры и алгоритмы обработки данных : учебное пособие / В. Д. Колдаев. - Москва : РИОР : ИНФРА-М, 2021. - 296 с. - ЭБС ZNANIUM.com. - URL: https://znanium.com/catalog/product/1230215 (дата обращения: 07.12.2023). – Текст : электронный.
5. Сенько, А. Работа с BIGDATA в облаках.Обработка и хранение данных с примерами из Microsoft / А. Сенько. - СПб.: Питер, 2019. - 448 с. - ЭБС ZNANIUM.com. - URL: https://znanium.com/catalog/product/1230215 (дата обращения: 07.12.2023). – Текст : электронный.
6. Нархид, Н. Apache Kafka. Потоковая обработка и анализ данных / Н. Нархид. - СПб.: Питер, 2019. - 320 с. ЭБС ZNANIUM.com. - URL: https://znanium.com/catalog/product/1230215 (дата обращения: 07.12.2023). – Текст: электронный.
7. Сенько, А Работа с BigData в облаках. Обработка и хранение данных с примерами из MIcrosoft Azure / А Сенько. - СПб.: Питер, 2019. - 448 с. ЭБС Университетская библиотека ONLINE. – URL: https://biblioclub.ru/index.php?page=book&id=598404 (дата обращения: 07.12.2023). – Текст : электронный.
8. Sun, X., Li, J., Kovalenko, A.V., Feng, W., Ou, Y. Integrating Reinforcement Learning and Learning From Demonstrations to Learn Nonprehensile Manipulation //IEEE Transactions on Automation Science and Engineering, 2023, 20(3), 1735–1744, DOI: 10.1109/TASE.2022.3185071, Q1
9. Petukhova, A.V.; Kovalenko, A.V.; Ovsyannikova, A.V. Algorithm for Optimization of Inverse Problem Modeling in Fuzzy Cognitive Maps. Mathematics 2022, 10, 3452. DOI: 10.3390/math10193452, Q1
10. Kirillova, E.; Kovalenko, A.; Urtenov, M. Study of the Current–Voltage Characteristics of Membrane Systems Using Neural Networks. AppliedMath 2025, 5, 10. https://doi.org/10.3390/appliedmath5010010
11. Kadurin, Artur, et al. "The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology." Oncotarget 8.7 (2016): 10883.
12. Kadurin, Artur, et al. "druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico." Molecular pharmaceutics 14.9 (2017): 3098-3104.
13. Polykovskiy, Daniil, et al. "Molecular sets (MOSES): a benchmarking platform for molecular generation models." Frontiers in pharmacology 11 (2020): 565644.
14. Khrabrov, Kuzma, et al. "$\nabla^ 2$ DFT: A Universal Quantum Chemistry Dataset of Drug-Like Molecules and a Benchmark for Neural Network Potentials." Advances in Neural Information Processing Systems 37 (2024): 36869-36889.
15. Polykovskiy, Daniil, et al. "Entangled conditional adversarial autoencoder for de novo drug discovery." Molecular pharmaceutics 15.10 (2018): 4398-4405.
16. Николенко, Сергей, Кадурин, Артур и Архангельская Екатерина. Глубокое обучение. Издательский дом" Питер", 2017.
17. https://yandex.cloud/ru/blog/posts/2022/10/nosql?ysclid=mcv4j5m94v759899232&utm_referrer=https%3A%2F%2Fyandex.ru%2F
18. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL | Уилсон Джим Р., Редмонд Эрик, ДМК Пресс, 2018.

19. Марин И., Шукла А., ВК С. Big Data Analysis with Python: Combine Spark and Python to unlock the powers of parallel computing and machine learning. — Packt Publishing, 2019. — 276 с. — ISBN 978-1789955286
20. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. — М.: ДМК, 2019. — 480 с. — ISBN 978-5-97060-660-5

5.3. Периодические издания и конференции (A*):

1. IEEE Transactions on Big Data – научные статьи по обработке больших данных.
2. Journal of Big Data (SpringerOpen) – открытый журнал с исследованиями в области Big Data.
3. Big Data Research (Elsevier) – публикации по анализу, управлению и визуализации данных.
4. Data Science Journal (CODATA) – междисциплинарные исследования данных.
5. ACM Transactions on Knowledge Discovery from Data (TKDD) – методы извлечения знаний из больших данных.
6. https://openreview.net/forum?id=FMMF1a9ifL
7. https://openreview.net/forum?id=ElUrNM9U8c#discussion
8. https://openreview.net/forum?id=JoO6mtCLHD
9. https://aclanthology.org/2024.findings-emnlp.760/
10. https://aclanthology.org/2020.coling-main.588/
11. https://link.springer.com/chapter/10.1007/978-3-030-72113-8_30
12. https://link.springer.com/chapter/10.1007/978-3-031-42448-9_10
13. https://aclanthology.org/2024.findings-naacl.288/

5.4. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Базы данных и аналитические платформы

1. Google BigQuery – облачная аналитика больших данных.
2. Apache Hadoop & Spark – официальная документация и ресурсы.
3. Kaggle – датасеты, соревнования и учебные материалы.
4. Cloudera – платформа для работы с Big Data.
5. Databricks – решения на основе Apache Spark.

Справочные системы и блоги

1. Towards Data Science (Medium) – статьи по Data Science и Big Data.
2. KDnuggets – новости, обучающие материалы и обзоры инструментов.
3. O’Reilly Data & AI – книги и статьи по Big Data и машинному обучению.
4. IBM Big Data Hub – кейсы и руководства по Big Data.

Ресурсы свободного доступа

1. Apache Spark Documentation
2. Yandex Cloud Big Data
3. Kaggle Datasets

Собственные электронные образовательные и информационные ресурсы КубГУ

1. Электронный каталог Научной библиотеки КубГУ http://megapro.kubsu.ru/MegaPro/Web
2. Электронная библиотека трудов ученых КубГУ http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6
3. Среда модульного динамического обучения http://moodle.kubsu.ru
4. База учебных планов, учебно-методических комплексов, публикаций и конференций http://infoneeds.kubsu.ru/

5. Библиотека информационных ресурсов кафедры информационных образовательных технологий http://mschool.kubsu.ru;
6. Электронный архив документов КубГУ http://docspace.kubsu.ru/
7. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" http://icdau.kubsu.ru/

6. Методические указания для обучающихся по освоению дисциплины (модуля)

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

По курсу предусмотрено проведение лекционных занятий, на которых дается систематизированный материал по технологиям обработки больших данных. В ходе лекций рассматриваются ключевые концепции.

Лабораторные занятия курса посвящены практическому освоению технологиям обработки больших данных

При самостоятельной работе студентам необходимо изучать рекомендованную литературу в виде официальной документации к используемым открытым программным продуктам, облачным платформам.

Важнейшим компонентом курса является самостоятельная проектная работа, в ходе которой студент разрабатывает законченное решение для решения задач (кейсов) индустриальных партнеров. Допускается выполнение проектов в командах.

Подход, определяющий установление соответствия кейсов ИП и УГТ (5-7), позволяет четко соотносить этапы развития технологии с вовлеченностью партнера и снижать риски при переходе от лабораторных испытаний к промышленному внедрению.

Ключевые аспекты взаимодействия с индустриальными партнерами:

- Для УГТ 5 – ИП помогает определить реалистичные условия тестирования, но не рискует своей инфраструктурой.
- Для УГТ 6 – ИП предоставляет "песочницу" или изолированную среду, где можно выявить скрытые проблемы.
- Для УГТ 7 – ИП становится соразработчиком, так как технология адаптируется под его конкретные процессы.

А. Применение технологий обработки больших данных в кейсах ПАО «Сбербанк»

1. Прогнозирование оттока клиентов (Churn Prediction)

Описание:

Анализ поведения клиентов для выявления тех, кто с высокой вероятностью может уйти к конкурентам.

Цель: Снизить отток клиентов на 15-20% за счет персональных предложений.

Технологии:

- **Spark MLlib** (CatBoost, XGBoost)

- **Hadoop HDFS** (хранение истории транзакций)
- **Tableau** (визуализация результатов)

Реализация:

a) Сбор данных: история транзакций, активность в приложении, обращения в поддержку.

b) Обучение модели на признаках:
- Снижение активности
- Уменьшение количества операций
- Жалобы в чате поддержки (NLP-анализ)

c) Интеграция с CRM-системой для автоматических предложений.

Результат:

- Точность модели: не менее **87%**
- Снижение оттока: **18%** за 6 месяцев
- Автоматизированные триггеры для маркетинга (например, cashback для "группы риска").

2. Real-time Anti-Fraud для платежей

Описание: Система для мгновенного выявления мошеннических операций.

Цель: Снизить ущерб от мошенничества на 30%.

Технологии:

- **Apache Kafka** (поток транзакций)
- **Spark Streaming** (анализ в реальном времени)
- **GraphX** (поиск связей между счетами)

Реализация:

a) Настройка Kafka-топика для транзакций (100K+ событий/сек).

b) Алгоритмы обнаружения аномалий:
- **Необычные суммы/места операций**
- **Повторяющиеся переводы на новые счета**

c) Автоматическая блокировка подозрительных операций.

Результат:

- **Скорость обработки: <100 мс на операцию**
- **Снижение фрода: 35%**
- **Интеграция с ЦБ РФ для отчетности.**

3. Персонализация предложений в мобильном приложении

Описание: ИИ-система для рекомендации финансовых продуктов.

Цель: Увеличить конверсию в продажах на 25%.

Технологии:

- **Apache Spark (анализ поведения)**
- **Redis (кеширование рекомендаций)**
- **A/B-тестирование (оптимизация алгоритмов)**

Реализация:

a) Сбор данных:
- **История покупок**
- **Геолокация**
- **Время активности**

b) Коллаборативная фильтрация + CatBoost.

c) Динамический интерфейс в приложении.

Результат:

- **Рост продаж кредитных карт: 28%**
- **Увеличение среднего чека: 15%**

4. Оптимизация работы колл-центра с NLP

Описание: Автоматизация обработки обращений клиентов.
Цель: Сократить нагрузку на операторов на 40%.
Технологии:

- **BERT/GPT-3 (классификация запросов)**
- **Kafka (поток аудио/текста)**
- **Yandex SpeechKit (STT/TTS)**

Реализация:

- Транскрипция звонков → текст.
- Классификация интентов (жалобы, вопросы по картам и т.д.).
- Автоответы через чат-бота.

Результат:

- 60% обращений решается без оператора
- Снижение времени ответа: с 5 мин до 30 сек

5. Оптимизация сети банкоматов с геоаналитикой
Описание: Анализ расположения и загрузки банкоматов.
Цель: Сократить затраты на инкассацию на 20%.
Технологии:

- GeoSpark (обработка геоданных)
- H3 Uber Hexagons (кластеризация)
- Kepler.gl (визуализация)

Реализация:

a) Сбор данных:

- Транзакции по координатам
- График инкассации

b) Поиск "мертвых" банкоматов.
c) Оптимизация маршрутов.
Результат:

- Сокращение банкоматов: 12%
- Экономия: 200 млн руб./год

Итоговая таблица эффективности

Кейс	Технологии	Экономический эффект
Прогнозирование оттока	Spark ML, Hadoop	+18% удержание
Anti-Fraud	Kafka, GraphX	-35% фрод
Персонализация	Spark, Redis	+28% продажи
NLP-колл-центр	BERT, SpeechKit	-60% нагрузка
Оптимизация АТМ	GeoSpark, H3	200 млн руб./год

Вывод: Сбербанк использует «Технологии обработки больших данных» для:

- Риск-менеджмента (фрод, скоринг)
- Маркетинга (персонализация)
- Оптимизации (логистика, автоматизация)

Лабораторные работы можно адаптировать под эти кейсы, используя **PySpark, Kafka и ML-библиотеки**.

Б. Применение технологий обработки больших данных в кейсах компании AVA LAB

1. Обнаружение мошеннических транзакций в реальном времени

Описание: Разработка системы для выявления подозрительных платежей в финтех-приложениях.

Цель: Снизить ущерб от мошенничества на 40% с задержкой обработки <100 мс.

Технологии:

- Apache Kafka (потоковая передача транзакций)
- Spark Structured Streaming (анализ в реальном времени)
- GraphX (выявление связанных аккаунтов)
- CatBoost (ML-модель для аномалий)

Реализация:

a) Настройка Kafka-топика для приема транзакций (до 50K событий/сек).
b) Обучение модели на исторических данных с метками "мошенничество/легитимно".
c) Развертывание Spark-джобы для потоковой обработки.
d) Интеграция с графовой БД (Neo4j) для визуализации связей.

Результат:

- Точность детекции: 92%
- Снижение фрода: 45%
- Скорость обработки: 80 мс

2. Оптимизация кредитного скоринга для МФО

Описание: Скоринговая система на основе альтернативных данных (цифровой след, соцсети).

Цель: Увеличить одобрение кредитов надежным заемщикам на 25%.

Технологии:

- PySpark ML (Feature Engineering)
- HDFS (хранение сырых данных)
- SHAP (интерпретируемость модели)

Реализация:

a) Сбор данных: история браузинга, геолокация, активность в соцсетях (с согласия).
b) Обучение Gradient Boosting-модели с учетом регуляризации.
c) Разработка дашборда в **Superset** для анализа решений.

Результат:

- Увеличение approval rate: +28%
- Снижение дефолтов: 15%
- Автоматизированное принятие решений для 80% заявок.

3. NLP-анализ голосовых обращений в колл-центр

Описание: Автоматизация обработки жалоб клиентов через speech-to-text и классификацию интентов.

Цель: Сократить затраты на колл-центр на 35%.

Технологии:

- Yandex SpeechKit (расшифровка аудио)
- BERT (классификация текста)
- Airflow (оркестрация pipeline)

Реализация:

a) Транскрипция звонков в текст (русский язык + диалекты).
b) Обучение BERT-модели на размеченных данных (15 категорий: "жалоба", "запрос информации" и т.д.).
c) Интеграция с CRM для автоматических ответов.

Результат:

- Точность классификации: 89%
- Сокращение ручной обработки: 60%
- Среднее время ответа: 20 сек (было 5 мин).

4. AML-аналитика для криптобирж

Описание: Выявление схем отмывания денег через анализ цепочек транзакций в блокчейне.

Цель: Обнаруживать 95% подозрительных операций.

Технологии:

- Spark GraphFrames (анализ графа транзакций)
- Temporal Graph Networks (учет временных меток)
- Elasticsearch (быстрый поиск паттернов)

Реализация:

a) Парсинг blockchain-данных (Bitcoin/Ethereum) в графовую структуру.
b) Поиск цикличных переводов и "мусорных" кошельков.
c) Визуализация схем в **Gephi**.

Результат:

- Обнаружено 1200+ подозрительных кластеров
- Интеграция с регуляторами (ЦБ, FATF)

5. Персонализация fintech-приложений

Описание: Рекомендательная система для финансовых продуктов на основе поведения пользователей.

Цель: Увеличить конверсию в покупку продуктов на 30%.

Технологии:

- Apache Flink (обработка событий в реальном времени)
- Redis (кеширование рекомендаций)
- Bandit-алгоритмы (A/B-тестирование)

Реализация:

a) Сбор данных: клики, время в приложении, демография.
b) Обучение hybrid-модели (коллаборативная фильтрация + content-based).
c) Динамическое обновление рекомендаций каждые **5 мин**.

Результат:

- Рост продаж: 32%
- Увеличение среднего чека: 18%

Сводная таблица результатов

Кейс	Ключевые технологии	Эффективность
Anti-Fraud	Kafka, Spark, GraphX	-45% фрод, 92% точность
Кредитный скоринг	PySpark, SHAP	+28% одобрений
NLP-колл-центр	BERT, SpeechKit	-60% затрат, 89% accuracy
AML для крипты	GraphFrames, Gephi	1200+ схем обнаружено
Персонализация	Flink, Redis	+32% конверсия

Вывод: AVA LAB использует «Технологии обработки больших данных» для:

- Безопасности (Anti-Fraud, AML)
- Финансовой аналитики (скоринг, рекомендации)
- Автоматизации (NLP, потоковая обработка)

Для лабораторных работ:

1. Реализовать детектор аномалий на синтетических транзакциях.
2. Построить граф связей для AML-анализа.
3. Обучить BERT-модель для классификации текстов.

Инструменты: **Spark, Kafka, Python (PySpark), JupyterHub**.

7. Материально-техническое обеспечение по дисциплине (модулю)

7.1 Перечень информационно-коммуникационных технологий

• Облачные платформы (Google Cloud, AWS, Microsoft Azure, Yandex Cloud)

• Распределённые системы хранения и обработки данных (HDFS, S3, HBase, Cassandra)

• Технологии потоковой обработки данных (Apache Kafka, Apache Flink, Apache Storm)

Системы управления базами данных (СУБД)

- Реляционные (PostgreSQL, MySQL)
- NoSQL (MongoDB, Redis, Elasticsearch)
- Фреймворки для распределённых вычислений (Apache Hadoop, Apache Spark)
- Инструменты визуализации данных (Tableau, Power BI, Apache Superset, Grafana)
- Контейнеризация и оркестрация (Docker, Kubernetes)
- Средства мониторинга и управления инфраструктурой (Prometheus, Grafana, ELK Stack)
- API и веб-сервисы (REST, GraphQL, gRPC)

7.2 Перечень лицензионного и свободно распространяемого программного обеспечения

Лицензионное ПО:

Интегрированные среды разработки (IDE):

JetBrains IntelliJ IDEA (с поддержкой Scala, Python, Java)

PyCharm Professional (для Python-разработки)

Microsoft Visual Studio (с инструментами для Big Data)

Корпоративные решения для Big Data:

Cloudera Data Platform (CDP)

Microsoft SQL Server (с поддержкой Big Data)

Облачные сервисы (платные подписки):

- Google BigQuery
- AWS EMR (Elastic MapReduce)
- Azure HDInsight

Свободно распространяемое (open-source) ПО:

Обработка и анализ данных:

Apache Hadoop (HDFS, MapReduce, YARN)

Apache Spark (для быстрой обработки данных)

Apache Flink (потоковая обработка)

Apache Kafka (распределённый потоковый брокер)

Базы данных и хранилища:

PostgreSQL (+ расширение TimescaleDB для временных рядов)

MongoDB (документоориентированная NoSQL)

Apache Cassandra (высокомасштабируемая NoSQL)

Redis (ключ-значение, кэширование)

Elasticsearch (поиск и аналитика)

Визуализация и BI-инструменты:

Apache Superset (альтернатива Tableau)

Grafana (мониторинг и дашборды)

Metabase (open-source BI)

Разработка и управление инфраструктурой:

Jupyter Notebook / JupyterLab (интерактивная аналитика)

Docker (контейнеризация)

Kubernetes (оркестрация контейнеров)

Apache Airflow (оркестрация ETL-процессов)

Языки программирования и библиотеки:

- **Python** (Pandas, NumPy, SciPy, Scikit-learn, PySpark)
- **R** (для статистического анализа)
- **Scala** (работа с Apache Spark)
- **SQL** (для работы с базами данных)

Дополнительные инструменты

- **Git** (система контроля версий, GitHub/GitLab/Bitbucket)
- **Apache Zeppelin** (аналитика и визуализация в браузере)
- **MLflow** (управление машинным обучением)
- **Apache NiFi** (автоматизация потоков данных)

Виртуальные машины, кластер Managed Kubernetes и ресурсы GPU в облаке предоставляется индустриальным партнером ПАО «Сбербанк».

№	Продукт	Параметры продукта	Кол-во	Кол-во конфигураций	Ед. изм.
1	Виртуальная машина	Виртуальная машина 10% vCPU 2 vCPU 4 RAM	1	60	Шт
		ОС Ubuntu 22.04	1		Шт
		Системный диск SSD	1		Шт
			10		Гб
		Аренда публичного IP	1		Шт
2	Виртуальная машина с GPU	Виртуальная машина с GPU NVIDIA® Tesla® V100 2 GPU 8 vCPU 128 ГБ RAM	1	1	Шт
		ОС Ubuntu_24.04	1		Шт
		Системный диск SSD	1		Шт
			2000		Гб
		Диск SSD	2		Шт
			4096		Гб
		Аренда публичного IP	1		Шт
3	K8S	Master node 8 vCPU 16 RAM	1	1	Шт
		Worker node 10% доля 4 vCPU 32 RAM	5		Шт

		Worker node SSD-NVME	64		Гб
		Аренда публичного IP	1		Шт
4	ML Inference Instance Type GPU	Время работы в месяц	40	1	Ч
		Инстанс 8 x NVIDIA® H100 NVLink PCIe 160 vCPU 1520 GB RAM	1		Шт
		Количество запросов к ML-моделям	1		Млн. Шт
		Кэш ML-моделей	160		Гб
5	LLM	Токены GigaChat 2 Max	50		Млн. Шт
		Токены Embeddings	400		Млн. Шт

Дополнительные облачные ресурсы предоставляются технологическим партнером Yandex Cloud.

№	Вид работ	Наименование учебной аудитории, ее оснащенность оборудованием и техническими средствами обучения
1.	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
2.	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3.	Практические занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
4.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
6.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.