Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение

«Академия управления городской средой, градостроительства и печати»

УТВЕРЖДАЮ

Заместитель директора

по учебно-методической работе

_____О.В. Фомичёва

«____» _____ 20_____г.

Методические рекомендации по организации и проведению практических занятий

ПМ.01 «РАЗРАБОТКА КОДА ДЛЯ ОБУЧЕНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»
МДК.01.03. Тестирование программных модулей

для специальности

специальности 09.02.13 Интеграция решений с применением технологий искусственного интеллекта

Форма обучение –очная

Санкт-Петербург
2025

Разработчик: Ипатова С.В./Оболенская Е.Г., методисты СПб ГБПОУ АУГСГиП

Одобрены на заседании цикловой комиссии

Общетехнических дисциплин и компьютерных технологий

Протокол № 4

09.12.2025 г.

Председатель цикловой комиссии:

Шурухина И.Е.

В результате изучения профессионального модуля обучающихся должен освоить основной вид деятельности ВД1. «Разработка кода для обучения искусственного интеллекта» и соответствующие ему общие компетенции и профессиональные компетенции:

1.1.1. Перечень общих компетенций

Код	Наименование общих компетенций
ОК 01	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях
ОК 04	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения
ОК 07	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
ОК 08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках

1.1.2. Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций
ВД 1	Разработка кода для искусственного интеллекта
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием.
ПК 1.3	Оформлять программный код в соответствии с техническим заданием.
ПК 1.4	Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки.
ПК 1.5	Выполнять отладку программных модулей с использованием специализированных программных средств.
ПК 1.6	Выполнять тестирование программного кода.
ПК1.7.	Составлять тестовые сценарии.

1.1.3. В результате освоения профессионального модуля обучающийся должен:

Иметь практический опыт	– Разработки, оптимизации и оценки сложности алгоритмов для ИИ-программ. – Использования библиотек и инструментов для работы с алгоритмами и данными (например: Pandas, NumPy, Scikit-learn).

	– Применения структур данных (деревья, графы, списки) для реализации алгоритмов. – Разработки модульных ИИ-систем, соответствующих требованиям производительности и безопасности. – Внедрения разработанных ИИ-модулей в комплексные программные системы. – Оптимизации кода и работы с интерфейсами для взаимодействия между модулями. – Оформления, документирования и структурирования кода для последующей поддержки. – Использования инструментов статического анализа кода для выявления ошибок и улучшения качества. – Работы с системами документирования кода (например, Doxygen, Sphinx). – Управления проектами с использованием систем контроля версий для организации командной работы. – Разрешения конфликтов при слиянии веток и использования pull request для рецензирования кода. – Настройки процессов CI/CD для автоматического тестирования и развертывания кода. – Отладки программных модулей с использованием пошаговой проверки. – Применения методов логирования и профилирования производительности. – Использования специальных средств для отладки многопоточных программ. – Выполнения статического тестирования программного кода на предмет выявления ошибок/дефектов алгоритмов, в том числе – на наличие обработки исключений – Выполнения тестирования программных модулей в соответствии в тест-планом – Генерирования тестовых данных – Выполнения интеграционного тестирования в соответствии с заданием – Выполнения регрессионного тестирования в соответствии с заданием. – Работы с CI/CD пайплайнами для автоматизации тестирования. – Разработки тестовых сценариев в соответствии с тестовым планом (тестирование производительности, надежности, UI-тестирование), в том числе с применением средств автоматизации проектирования. – Разработки тестовых пакетов и заданий на выполнение тестирования. – Оценки тестовых данных на предмет покрытия строк и покрытия ветвей, выполнения валидации данных. – Автоматизации создания и выполнения тестовых сценариев.
Уметь	– Анализировать технические задания и выявлять требования к алгоритмам. – Применять методы алгоритмизации для решения задач программирования.

	– Разрабатывать оптимальные алгоритмы для решения задач в области ИИ. – Реализовывать программные модули на основе требований технического задания. – Соблюдать при разработке принципы «чистого кода». – Использовать стандартные библиотеки и фреймворки для ускорения разработки. – Оформлять код в соответствии с принятыми стандартами и требованиями. – Документировать разработанный программный код. – Соблюдать соглашения о наименованиях переменных, функций и классов (например, PEP8 для Python). – Работать с системами контроля версий для управления проектами. – Организовывать совместную работу над проектом через ветки разработки и слияние изменений. – Разрешать конфликты при слиянии кода. – Использовать инструменты для отладки программного кода. – Идентифицировать и исправлять ошибки в программе. – Применять методы логирования для анализа выполнения программ. – Проводить различные виды тестирования (юнит-тестирование, интеграционное тестирование). – Выполнять настройки окружения т подготовку тестовых данных – Фиксировать результаты выполнения тестов и подготавливать отчеты о результатах тестов. – Определять уровень критичности дефектов. – Разрабатывать автоматизированные тесты для тестирования модулей и/или отдельных функций – Восстанавливать окружение и тесты после сбоя – Проектировать тестовые сценарии на основе тестовых планов. – Разрабатывать тестовые пакеты и задания на выполнение тестирования. – Использовать шаблоны для написания тест-кейсов. – Оценивать риски при отборе тестов для регрессионного тестирования. – Оценивать тесты на соответствие целям тестирования.
Знать	– Основные методы и подходы к построению алгоритмов (типовые поисковые алгоритмы, жадные алгоритмы, динамическое программирование, рекурсивные подходы). – Принципы эффективной обработки данных. – Языки программирования, применяемые для разработки алгоритмов. – Принципы модульного программирования. – Языки программирования для разработки модулей. – Стандартные фреймворки и библиотеки для работы с ИИ. – Основные принципы чистого кода (Clean Code). – Стандарты и практики документирования программного обеспечения. – Инструменты для автоматической проверки качества кода (например, PyLint, ESLint).

	– Принципы работы распределенных систем контроля версий. – Основные команды и операции в системах контроля версий (например: commit, pull, push, merge). – Методы разрешения конфликтов в ходе групповой разработки. – Принципы работы отладчиков и логирования. – Способы выявления ошибок в программе (отладка по шагам, точки останова). – Инструменты для отладки кода (например, PyCharm, Visual Studio Debugger). – Техники выполнения тестовых прогонов. – Инструменты и среды выполнения тестирования – Языки разработки автоматизированных тестов – Инструменты для тестирования программного кода. – Правила выполнения отчетов о тестировании – Цели, задачи и виды тестирования. Понятие стратегии тестирования. – Жизненный цикл дефекта. – Основы тест-дизайна: тестовый сценарий, тестовый пакет, чек-лист, основные шаблоны. – Основные инструменты проектирования тестов. – Методы и подходы к написанию тестов (Test-Driven Development, Behavior-Driven Development).

Практические работы

тема	название ПР	часы
МДК.01.03. Тестирование программных модулей		
Тема 3.1. Основы тестирования программных приложений	**Практическая работа №1.** Определение целей тестирования для каждого уровня и вида тестирования	8
Тема 3.2. Основы тест-дизайна	**Практическая работа № 2.** Подготовка тестового пакета и задания на тестирование	8
	Практическая работа № 3. Подготовка тестового сценария	10
Тема 3.3. Особенности тестирования ИИ-систем	**Практическая работа №.4.** Обучение и прогноз модели логистической регрессии	10
	Практическая работа №.5. Построение и визуализация матрицы ошибок	10
	Практическая работа №.6. Оценка качества нейронной сети с использованием ROC-кривой.	10
Тема 3.5. Тестирование ИИ-приложений	**Практическая работа № 7.** Юзабилити-тестирование приложения после интеграции.	10
	Практическая работа № 8. Тестирование безопасности ИИ-приложений. Тестирование совместимости с браузерами	10
	Практическая работа № 9. Тестирование API	10
	Практическая работа № 10. Мониторинг производительности ИИ-модели с использованием систем мониторинга и оповещения и мониторинга и визуализации данных.	10
		116

МДК.01.03. Тестирование программных модулей

Практическая работа №1. Определение целей тестирования для каждого уровня и вида тестирования

Цель работы: определение целей тестирования.

Задачи:

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

Оборудование, технические средства и инструменты:

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

Ход практического занятия:

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями, студенты выполняют практическую работу.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

Теоретические и учебно-методические материалы по теме практической работы

План создания тестового сценария:

1. **Определение целей тестирования**. Что именно нужно проверить: функциональность, производительность, безопасность или совместимость.
2. **Выбор тестовых данных**. Нужно убедиться, что они репрезентативны и соответствуют реальным условиям использования.
3. **Описание шагов тестирования**. Необходимо описать последовательность действий, которые должен выполнить тестировщик.
4. **Ожидаемые результаты**. Нужно определить, чего ожидают после выполнения тестовых шагов. Это поможет понять, успешно ли прошёл тест.
5. **Фактические результаты и оценка**. Необходимо записать, что произошло в результате тестирования, и сравнить это с ожидаемыми результатами.
6. **Документирование и анализ**. Все результаты нужно задокументировать. Если тест не пройден, следует проанализировать причины и определить шаги для устранения проблем.

ncsa.ru

Выбор видов тестирования зависит от поставленных требований, этапа разработки, условий использования системы. Некоторые виды тестирования и их цели:

- **Компонентное (модульное)**. Цель — выявление локализованных в модуле ошибок в реализации алгоритмов, а также определение степени готовности системы к переходу на следующий уровень разработки и тестирования.
- **Интеграционное**. Предназначено для проверки связи между компонентами, а также взаимодействия с различными частями системы (операционной системой, оборудованием либо связи между различными системами).
- **Системное**. Проверяет как функциональные, так и нефункциональные требования к системе в целом.
- **Позитивное**. Основная цель — проверка того, что при помощи системы можно делать то, для чего она создавалась.
- **Негативное**. Основной целью является проверка устойчивости системы к воздействиям различного рода, валидация неверного набора данных, проверка обработки исключительных ситуаций.

1. Цели по уровням тестирования

1.1. Модульное (компонентное) тестирование

- Найти максимальное число сбоев на ранних стадиях разработки.
- Выявить и устранить основные дефекты отдельных модулей, классов, функций.
- Проверить корректность работы изолированных компонентов в соответствии с их спецификациями.
- Обеспечить надёжность базовых строительных блоков системы.
- Подготовить компоненты к последующему интеграционному тестированию.

1.2. Интеграционное тестирование

- Проверить взаимодействие между компонентами/модулями системы.
- Убедиться, что интерфейсы соответствуют проектным требованиям (функциональным и нефункциональным).
- Обнаружить дефекты в механизмах обмена данными и вызовах между компонентами.
- Снизить риск пропуска ошибок на более высокие уровни тестирования.
- Повысить уверенность в корректности межкомпонентных связей.

1.3. Системное тестирование

- Проверить выполнение всех функциональных и нефункциональных требований к системе в целом.
- Выявить дефекты, связанные с использованием ресурсов, комбинациями данных, сценариями использования.
- Оценить совместимость системы с окружением (ОС, браузеры, устройства).
- Проверить удобство использования и соответствие пользовательским ожиданиям.
- Подтвердить готовность системы к приемочному тестированию.

1.4. Приемочное тестирование

- Подтвердить, что система работает как ожидается и удовлетворяет требованиям.
- Проверить завершенность системы и её готовность к эксплуатации.
- Оценить соответствие функциональному и нефункциональному поведению установленным требованиям.

- Получить подтверждение готовности системы к развертыванию от заказчика/пользователей.
- Выявить дефекты, специфичные для реальной среды эксплуатации.

2. Цели по видам тестирования

2.1. Функциональное тестирование

- Проверить, что система выполняет заявленные функции согласно требованиям.
- Убедиться в корректности обработки входных данных и формировании выходных результатов.
- Валидировать сценарии использования и бизнес-процессы.
- Обнаружить несоответствия между реализованной функциональностью и спецификациями.

2.2. Тестирование пользовательского интерфейса (GUI)

- Проверить соответствие интерфейса дизайн-спецификациям (размеры, шрифты, цвета).
- Убедиться в согласованности поведения элементов интерфейса.
- Оценить удобство навигации и интуитивность взаимодействия.
- Выявить визуальные дефекты и несоответствия стандартам UX/UI.

2.3. Тестирование безопасности

- Проверить устойчивость системы к внешним угрозам и атакам.
- Выявить уязвимости в механизмах аутентификации, авторизации, шифрования.
- Оценить защиту конфиденциальных данных от несанкционированного доступа.
- Проверить соответствие требованиям безопасности и нормативам.

2.4. Тестирование производительности

- Оценить время отклика системы при различных нагрузках.
- Определить пределы масштабируемости и точки насыщения.
- Выявить узкие места, влияющие на производительность.
- Проверить стабильность работы при длительной нагрузке.
- Убедиться, что система соответствует SLA по производительности.

2.5. Нагрузочное тестирование

- Имитировать работу множества пользователей одновременно.
- Проверить поведение системы при плановой и пиковой нагрузке.
- Оценить использование ресурсов (CPU, память, дисковое пространство).
- Выявить проблемы, возникающие при высокой интенсивности операций.

2.6. Стрессовое тестирование

- Проверить работоспособность системы в экстремальных условиях.
- Оценить способность к восстановлению после критических сбоев.
- Выявить точки отказа и механизмы деградации производительности.
- Проверить устойчивость к аварийным изменениям конфигурации.

2.7. Тестирование стабильности (надежности)

- Проверить долгосрочную работоспособность системы при средней нагрузке.
- Выявить утечки памяти, накопления ошибок, деградации производительности.

- Оценить время безотказной работы между перезапусками.
- Проверить механизмы автоматического восстановления.

2.8. Тестирование установки

- Проверить корректность инсталляции, настройки и обновления ПО.
- Убедиться в возможности полного удаления продукта без последствий.
- Оценить совместимость с различными конфигурациями окружения.
- Выявить проблемы, возникающие при миграции данных.

2.9. Тестирование удобства пользования (Usability)

- Оценить интуитивность интерфейса и легкость освоения.
- Проверить доступность для пользователей с ограниченными возможностями.
- Выявить барьеры, мешающие эффективному взаимодействию.
- Собрать обратную связь о пользовательском опыте (UX).

2.10. Тестирование на отказ и восстановление

- Проверить способность системы противостоять сбоям.
- Оценить механизмы резервного копирования и восстановления данных.
- Проверить время восстановления после отказа (RTO).
- Убедиться в сохранности данных при аварийных ситуациях.

2.11. Конфигурационное тестирование

- Проверить работу ПО на различных платформах и версиях ОС.
- Убедиться в совместимости с заявленными драйверами и оборудованием.
- Оценить поведение при разных настройках окружения.
- Выявить зависимости от конкретных конфигураций.

2.12. Дымовое тестирование (Smoke Testing)

- Быстро проверить базовую работоспособность после сборки.
- Убедиться, что основные функции запускаются и выполняются.
- Принять решение о целесообразности дальнейшего тестирования.
- Выявить критические дефекты на раннем этапе.

2.13. Регрессионное тестирование

- Проверить, что внесённые изменения не нарушили существующую функциональность.
- Убедиться в сохранении работоспособности ранее исправленных дефектов.
- Оценить влияние обновлений на смежные модули.
- Поддерживать стабильность системы при эволюционном развитии.

2.14. Повторное тестирование (Re-testing)

- Подтвердить исправление конкретных дефектов.
- Проверить воспроизводимость ранее обнаруженных ошибок.
- Убедиться, что исправления не создали новых проблем.
- Документировать статус устранения дефектов.

2.15. Тестирование сборки (Build Verification Test)

- Проверить целостность и корректность новой сборки.
- Убедиться в наличии всех необходимых компонентов.
- Выполнить минимальные проверки работоспособности.
- Принять решение о передаче сборки на дальнейшее тестирование.

2.16. Санитарное тестирование (Sanity Testing)

- Быстро оценить работоспособность ключевых функций после изменений.

- Проверить логическую целостность системы.
- Выявить очевидные дефекты, требующие немедленного внимания.
- Принять решение о глубине последующего тестирования.

3. Общие цели тестирования (для всех уровней и видов)

- **Обеспечение качества:** Создать уверенность в уровне качества продукта.
- **Предотвращение дефектов:** Выявить потенциальные проблемы до их проявления в эксплуатации.
- **Снижение рисков:** Минимизировать вероятность сбоев и ущерба от ненадлежащего качества.
- **Соответствие требованиям:** Проверить выполнение функциональных и нефункциональных требований.
- **Информационная поддержка:** Предоставить заинтересованным сторонам данные для принятия решений.
- **Оптимизация затрат:** Снизить стоимость исправления дефектов за счёт их раннего обнаружения.

Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

Содержание отчёта

1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

Ответить на контрольные вопросы.(устно)

Сформулировать выводы по результатам работы.

Сдать и защитить работу.

Практическая работа № 2. Подготовка тестового пакета и задания на тестирование

Цель работы: изучить основы разработки тестовых пакетов, которые проверяют определённую функциональность или компонент программного обеспечения

Задачи:

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

Оборудование, технические средства и инструменты:

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

Ход практического занятия:

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

• Руководствуясь методическими рекомендациями, студенты выполняют практическую работу.

• Предъявить преподавателю результаты работы.

• Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).

• Подготовить ответы на контрольные вопросы.

Теоретические и учебно-методические материалы по теме практической работы

Шаги, которые помогают разработать тестовые пакеты:

1. **Определение области тестирования**. Нужно решить, какие функции или модули программы будут тестироваться.
2. **Разработка тестовых случаев**. Необходимо создать список тестовых случаев, которые покрывают все возможные сценарии использования функций или модулей.
3. **Подготовка тестовых данных**. Нужно собрать или сгенерировать данные, которые будут использоваться в тестах.
4. **Настройка тестового окружения**. Следует убедиться, что есть всё необходимое для выполнения тестов (например, доступ к серверам, настроенные инструменты).
5. **Выполнение тестов**. Нужно запустить тесты и зафиксировать результаты.
6. **Анализ результатов**. Следует оценить результаты тестов и определить, соответствуют ли они ожиданиям.
7. **Документирование**. Нужно записать процесс тестирования и результаты в отчёты.
8. **Улучшение и исправление**. Если были найдены ошибки, следует разработать план их исправления.

ncsa.ru

Одно из возможных заданий — подготовить набор тестовых вариантов для обнаружения ошибок в программе. Ещё одно задание — на основании проведённых тестов составить рекомендации по исправлению ошибок, выявленных в ходе тестирования, в виде отчёта. multiurok.ru

Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

Содержание отчёта

1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

Ответить на контрольные вопросы.(устно)

Сформулировать выводы по результатам работы.

Сдать и защитить работу.

Практическая работа № 3. Подготовка тестового сценария

Цель работы: описать последовательность действий, которые нужно выполнить для проверки конкретной функции или аспекта системы.

Задачи:

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

Оборудование, технические средства и инструменты:

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

Ход практического занятия:

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями, студенты выполняют практическую работу.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

Теоретические и учебно-методические материалы по теме практической работы

Структура тестового сценария (минимально) включает: qatools.ru

- **Название** — краткое описание тестируемого процесса.
- **Описание** — что именно тестируется и в каком контексте.
- **Предусловия** — условия, которые должны быть выполнены перед тестом.
- **Шаги** — основные действия, описывающие сценарий.
- **Ожидаемый результат** — критерий успешного выполнения сценария.

qatools.ru

Методика

Разработка тестового сценария включает несколько этапов: practicum.yandex.ru

1. **Сбор требований** к программе: изучить спецификацию, обсудить результаты работы ПО, проанализировать пользовательские сценарии.
2. **Определение цели** — что именно будет проверяться сценарием (проверка функциональности или поиск ошибок).
3. **Описание шагов** — шаги должны быть ясными, понятными и охватывать все необходимые действия для достижения цели.
4. **Описание ожидаемых результатов** — для каждого шага необходимо прописать конкретные результаты, к которым он должен привести.

5. **Документирование** — из последовательности действий формируют документ формата, принятого в компании.

practicum.yandex.ruru.hexlet.io

Рекомендации:

- Избегать неоднозначных формулировок.
- Учитывать разные сценарии использования — предусмотреть нажатия не на те кнопки, ошибочный ввод и другие ситуации, которые выходят за рамки стандартного использования программы.
- Определить начальные условия — например, задать, должен ли тестировщик быть зарегистрированным или какой браузер ему следует использовать для тестирования.

practicum.yandex.ru

Пример

Пример тестового сценария для проверки регистрации пользователя на сайте: tquality.ru

- **Что нужно сделать**: открыть страницу регистрации, ввести имя пользователя, адрес электронной почты, пароль, нажать кнопку «Зарегистрироваться».
- **Ожидаемый результат**: появляется сообщение «Регистрация прошла успешно», и пользователь переходит на страницу входа.

tquality.ru

Отчет

В отчёте по практической работе №3 рекомендуется:

- **Описать тестовый сценарий** — указать, что он описывает, и привести примеры шагов и ожидаемых результатов. practicum.yandex.rutquality.ruqatools.ru
- **Указать, как сценарий может быть использован** — например, для ручной проверки или для вариантов использования инструментов автоматизации. ru.hexlet.io
- **Описать, как тестовые сценарии следует регулярно обновлять** по мере изменения требований или функциональности ПО, чтобы они оставались актуальными. tquality.ru
- ***Оформление результатов работы***
- Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:
- ***Содержание отчёта***
- 1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

- **Ответить на контрольные вопросы. (устно)**
- ***Сформулировать выводы по результатам работы.***
- ***Сдать и защитить работу.***

Практическая работа №.4. Обучение и прогноз модели логистической регрессии

Цель работы: оценить вероятность того, что объект принадлежит к определённому классу, на основе значений независимых переменных (предикторов).

Задачи:

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

Оборудование, технические средства и инструменты:

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

Ход практического занятия:

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями, студенты выполняют практическую работу.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

Теоретические и учебно-методические материалы по теме практической работы

Работа может включать подготовку данных, построение модели, оценку качества и интерпретацию результатов. colab.research.google.comsky.prozhanibekov.edu.kz

Подготовка данных

Некоторые этапы подготовки данных для логистической регрессии:

- **Исследовательский анализ** — исследование распределений переменных, выявление выбросов.
- **Обработка пропущенных значений** — заполнение или удаление соответствующих наблюдений.
- **Кодирование категориальных переменных** — преобразование текстовых переменных в числовой формат (например, горячее кодирование или кодирование меток).
- **Масштабирование числовых признаков** — приведение переменных к единой шкале через стандартизацию или нормализацию.
- **Работа с несбалансированными классами** — применение методов ресэмплинга для уравновешивания классов.

sky.profastercapital.com

Построение модели

Для построения модели логистической регрессии можно использовать библиотеки машинного обучения, например **Scikit-learn** в Python. Некоторые шаги: sky.prosimplilearn.com

- **Создание объекта класса LogisticRegression**.
- **Обучение модели** на обучающих данных и метках.

- **Прогнозирование** — использование обученной модели для предсказания значений для тестовой выборки.

simplilearn.comeducative.io

Оценка качества

После прогнозирования на тестовой выборке предсказанные значения нужно сравнить с реальными. Для этого используются специальные метрики, поскольку модель предсказывает вероятность принадлежности к классу, а не конкретное значение. Некоторые метрики: cloud.ruzhanibekov.edu.kz

- **Матрица ошибок (Confusion Matrix)** — таблица, в которой показаны правильные и неправильные предсказания для каждого класса.
- **Точность (Accuracy)** — доля правильных предсказаний среди общего числа предсказаний.
- **Полнота (Recall)** — доля правильно предсказанных положительных примеров среди всех истинных положительных примеров.

zhanibekov.edu.kz

Интерпретация результатов

Коэффициенты логистической регрессии могут быть интерпретированы как изменение логарифма шансов на единицу изменения независимой переменной. Это позволяет оценивать значимость различных признаков в модели. zhanibekov.edu.kz

Однако интерпретация коэффициентов может быть сложной из-за нелинейности, эффектов взаимодействия между предикторами и других факторов. geeksforgeeks.org

Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

Содержание отчёта

1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

Ответить на контрольные вопросы. (устно)

Сформулировать выводы по результатам работы.

Сдать и защитить работу.

Практическая работа №.5. Построение и визуализация матрицы ошибок регрессии

Цель работы: визуализировать эффективность алгоритма классификации, выявить не только количество правильных прогнозов, но и типы ошибок, которые совершает модель.

Задачи:

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ

- Показать основные приемы эффективного использования возможностей ИИ

Оборудование, технические средства и инструменты:

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

Ход практического занятия:

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями, студенты выполняют практическую работу.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

Теоретические и учебно-методические материалы по теме практической работы

Методика

- **Создание матрицы ошибок**. В матрице строки представляют истинные классы (реальные метки), а столбцы — предсказанные классы (метки, которые предсказала модель). Размер матрицы соответствует количеству классов. work-zilla.comhabr.com
- **Интерпретация элементов матрицы**. Например, для бинарной классификации:
  - **True Positive (TP)** — модель правильно предсказала позитивный класс; sky.prohabr.com
  - **False Positive (FP)** — модель ошибочно предсказала позитивный класс (ошибка I рода); sky.pro
  - **False Negative (FN)** — модель ошибочно предсказала негативный класс (ошибка II рода); sky.pro
  - **True Negative (TN)** — модель правильно предсказала негативный класс. sky.pro
- **Визуализация матрицы**. Например, с помощью **тепловой карты (heatmap)**, где цветом выделяются ячейки: чем ярче цвет на диагонали (где реальный класс совпадает с предсказанным), тем лучше работает модель. habr.comcloud.ru

Примеры

- **Задача многоклассовой классификации**. Например, анализ эмоциональной окраски текстов для сервиса обратной связи. Нужно построить матрицу ошибок для позитивных, нейтральных и негативных отзывов и выявить закономерность: нейтральные отзывы регулярно классифицировались как позитивные, но практически никогда — как негативные.

- **Задача прогнозирования спроса на скоропортящуюся продукцию**.

Например, нужно построить матрицу ошибок для прогноза, что по цене k будут полностью раскуплены все цветы в количестве n букетов, или что по цене k будут полностью раскуплены не все цветы, останется m не проданных букетов.

sky.probigdataschool.ru

Инструменты

Для построения и визуализации матрицы ошибок в практических работах по анализу данных можно использовать:

- **Библиотеку Scikit-learn**. В ней матрица ошибок создаётся с помощью функции

confusion_matrix

из модуля metrics. Важно корректно задать true и predicted labels, чтобы не перепутать строки и столбцы, и учитывать порядок классов с помощью параметра labels, особенно при наличии несбалансированных данных. work-zilla.comlabex.io

- **Библиотеки Matplotlib и Seaborn**. Они помогают визуализировать матрицу ошибок, например, с помощью тепловых карт. cloud.ru

Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

Содержание отчёта

1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

Ответить на контрольные вопросы. (устно)

Сформулировать выводы по результатам работы.

Сдать и защитить работу.

Практическая работа №.6. Оценка качества нейронной сети с использованием ROC-кривой.

Цель работы: понять, как ROC-кривая показывает зависимость между True Positive Rate (TPR, чувствительностью) и False Positive Rate (FPR, специфичностью) для различных пороговых значений классификационных баллов.

Задачи:

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

Оборудование, технические средства и инструменты:

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

Ход практического занятия:

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

• Руководствуясь методическими рекомендациями, студенты выполняют практическую работу.

• Предъявить преподавателю результаты работы.

• Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).

• Подготовить ответы на контрольные вопросы.

Теоретические и учебно-методические материалы по теме практической работы

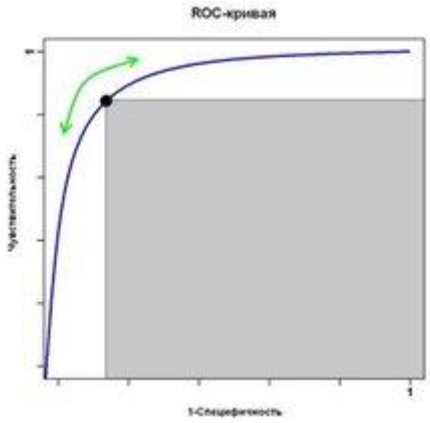
Теория

ROC-кривая — графическое представление компромисса между чувствительностью и специфичностью при различных порогах классификации. Каждая точка на кривой соответствует паре значений TPR и FPR для определённого порогового значения. habr.commathworks.com

Интерпретация кривой:

- **Идеальный классификатор** — кривая проходит через верхний левый угол (TPR = 1, FPR = 0).
- **Диагональная линия** — кривая для классификатора, который не лучше случайного угадывания.
- **Оптимальное пороговое значение** зависит от контекста задачи и баланса между TPR и FPR, который считается приемлемым.

habr.comcyberleninka.ru



Алгоритм

Для построения ROC-кривой используется мягкая классификация: вместо того, чтобы чётко отнести объект к классу, классификатор возвращает вероятности принадлежности объекта к различным классам. Эта уверенность сравнивается с порогом (какой уверенности «достаточно», чтобы отнести объект к положительному классу). В зависимости от значения этого порога меняются значения TPR и FPR. neerc.ifmo.ru

Реализация

Для построения ROC-кривой можно использовать библиотеки, например:

- **Scikit-learn** — функция

roc_curve

, которая вычисляет значения TPR и FPR.

- **MATLAB** — объект

rocmetrics

, который поддерживает задачи двоичной и многоклассовой классификации.

external.softwaremathworks.com

Важно: ROC-кривая строится на основе вероятностей, а не предсказанных классов. external.software

Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

Содержание отчёта

1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

Ответить на контрольные вопросы. (устно)

Сформулировать выводы по результатам работы.

Сдать и защитить работу.

Практическая работа № 7. Юзабилити-тестирование приложения после интеграции.

Цель работы: выявить проблемы, возникающие при интеграции, и оптимизировать интерфейс для пользователей.

Задачи:

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

Оборудование, технические средства и инструменты:

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

Ход практического занятия:

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями, студенты выполняют практическую работу.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).

- Подготовить ответы на контрольные вопросы.

Теоретические и учебно-методические материалы по теме практической работы

- **Повышение удобства использования** — идентификация элементов интерфейса, которые вызывают затруднения у пользователей, и их оптимизация.
- **Улучшение пользовательского опыта (UX)** — понимание ожиданий пользователей и адаптация дизайна и функциональности продукта под эти ожидания.
- **Снижение затрат на поддержку** — минимизация количества обращений в службу поддержки за счёт более интуитивно понятного интерфейса.
- **Увеличение конверсии** — улучшение ключевых показателей эффективности приложения, таких как время, проведённое на странице, и конверсия.

habr.com

Методика

Некоторые этапы юзабилити-тестирования:

- **Разработка сценариев** — описание шагов, которые должны выполнить участники. Сценарии должны быть контекстными и отражать реальные потребности пользователей. practicum.yandex.rusky.pro
- **Проведение тестирования** — участники выполняют задания, действия и комментарии фиксируются. Важно создать условия, которые максимально похожи на реальную среду пользователя. habr.comSEOnews.rupracticum.yandex.ru
- **Фиксация результатов** — результаты записываются, указываются оценка успешности прохождения сценариев для каждого участника (например, «Не выполнил», «Выполнил со сложностями», «Выполнил успешно»). practicum.yandex.ru
- **Анализ результатов** — выявляются проблемы, которые мешают пользователям решать задачи с помощью продукта, и формулируются рекомендации по их устранению. practicum.yandex.ruSEOnews.ru

Инструменты

Для проведения юзабилити-тестирования могут использоваться:

- **Модерируемое тестирование** — проводится в присутствии модератора, который наблюдает за пользователем в реальном времени и задаёт вопросы.
- **Немодерируемое тестирование** — пользователи выполняют задачи самостоятельно в отсутствие модератора, данные собираются автоматически через специализированное программное обеспечение.
- **Групповое тестирование (фокус-группы)** — обсуждение продукта группой пользователей для получения мнений и предпочтений относительно его юзабилити.

habr.comuplab.ru

Анализ результатов

Некоторые рекомендации по анализу результатов:

- **Определить ключевые показатели** — выбрать важные данные, которые соответствуют целям тестирования (показатели выполнения задач, время на выполнение, частота ошибок и др.).

- **Проанализировать качественные данные** — провести тематический анализ комментариев и наблюдений пользователей на предмет повторяющихся тем, проблем и закономерностей.
- **Группировать и категоризировать результаты** — на основе задач или функций оцениваемого продукта распределить результаты по категориям или темам.
- **Определить приоритетность результатов** — использовать критерии, которые соответствуют целям тестирования, например, степень проблемы, как часто она возникает, как это влияет на пользовательский опыт.

geeksforgeeks.org

Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

Содержание отчёта

1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

Ответить на контрольные вопросы. (устно)

Сформулировать выводы по результатам работы.

Сдать и защитить работу.

Практическая работа № 8. Тестирование безопасности ИИ-приложений. Тестирование совместимости с браузерами

Цель работы: проверка работоспособности приложения в различных средах: браузерах, операционных системах, устройствах и сетевых

Задачи:

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

Оборудование, технические средства и инструменты:

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

Ход практического занятия:

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями, студенты выполняют практическую работу.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

Теоретические и учебно-методические материалы по теме практической работы

Задание: провести тестирование совместимости, используя разные методы и инструменты, а также оценить результаты. mws.rudzen.rufreecodecamp.org

Цели

- **Обеспечить согласованность функционирования** приложения в разных браузерах (Google Chrome, Mozilla Firefox, Apple Safari и других). Устранить визуальные несоответствия, функциональные проблемы. freecodecamp.orgfreecodecamp.org
- **Проверить, будет ли приложение работать** во всех версиях различных браузеров. Например, протестировать на старых версиях популярных браузеров (например, Internet Explorer 11) для поддержки пользователей, которые не обновляли свои браузеры. wiki.merionet.rufreecodecamp.orgdzen.ru
- **Выявить ошибки** в работе приложения в разных средах, например, из-за различий в том, как браузеры интерпретируют код HTML, CSS и JavaScript. freecodecamp.org

Методы

- **Ручное тестирование** — тестировщик проверяет работу приложения на реальных устройствах и браузерах, учитывая особенности отображения, функциональности и производительности.
- **Автоматизация** — с помощью фреймворков (Selenium, Playwright, Cypress, TestCafe) можно автоматизировать повторяющиеся тест-кейсы и быстро протестировать стандартные пользовательские сценарии (авторизация, отправка форм, навигация).
- **Облачные платформы** — сервисы вроде BrowserStack, Sauce Labs и LambdaTest предоставляют доступ к тысячам комбинаций браузеров, ОС и устройств (включая мобильные).
- **Эмуляция и симуляция** — встроенные инструменты браузеров (Chrome DevTools Device Mode, Firefox Responsive Design Mode) имитируют различные разрешения экранов и поведение устройств.

mws.ru

Инструменты

Для тестирования совместимости с браузерами используются, например:

- **Инструменты визуального регрессионного тестирования** — например, Percy, который делает скриншоты в различных браузерах, или Applitools Eyes, использующий алгоритмы компьютерного зрения для выявления визуальных отличий.
- **Инструменты для автоматизации кросс-браузерного тестирования** — Selenium WebDriver, Cypress, Playwright.
- **Инструменты для эмуляции мобильных устройств** — например, Chrome DevTools (режим эмуляции устройств с предустановленными профилями популярных смартфонов и планшетов).

sky.pro

Критерии оценки

Результаты тестирования совместимости с браузерами оценивают по следующим параметрам:

- **Согласованность** — приложение работает одинаково или очень схоже в разных браузерах. freecodecamp.orgfreecodecamp.org
- **Отсутствие сбоев** — например, корректное отображение шрифтов, цветов, макетов и картинок в разных браузерах. dzen.rufreecodecamp.org
- **Отсутствие проблем с производительностью** — измерение времени загрузки страниц, проверка способности приложения к реагированию на разных экранах,

обеспечение, что ресурсоёмкие функции (видео или анимация) не вызывают проблем с производительностью. dzen.ru

-

Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

Содержание отчёта

1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

Ответить на контрольные вопросы. (устно)

Сформулировать выводы по результатам работы.

Сдать и защитить работу.

Практическая работа № 9. Тестирование API

Цель работы: изучение принципов тестирования API, использование специализированных инструментов и решение практических задач.

Задачи:

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

Оборудование, технические средства и инструменты:

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

Ход практического занятия:

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями, студенты выполняют практическую работу.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

Теоретические и учебно-методические материалы по теме практической работы

Основная цель — убедиться, что API работает корректно и соответствует спецификациям, определённым разработчиками. Важно проверить, выполняет ли API свои функции правильно, обрабатывает ли входные данные, возвращает ли ожидаемые результаты. deveducation.com

Задачи тестирования API:

- **Проверка функциональности** — убедиться, что API выполняет все свои функции согласно документации и спецификациям.
- **Тестирование структуры данных** — убедиться, что данные, передаваемые через API, имеют правильную структуру и формат.
- **Проверка граничных условий** — узнать, как API обрабатывает необычные и крайние сценарии, такие как большие объёмы данных или некорректные запросы.
- **Тестирование безопасности** — определить уровень защиты API от угроз, таких как атаки на основе инъекций или обхода авторизации.

deveducation.com

Принципы

Некоторые принципы тестирования API:

- **Использование правильных и разнообразных входных данных** — это помогает убедиться, что API правильно обрабатывает все возможные входные сценарии.
- **Автоматизация тестирования** — например, проверка отдельных функций или обработка ошибок.
- **Непрерывность тестов** — если в компании есть процессы CI/CD, важно включить в них тестирование API, чтобы регулярно проверять его работоспособность и получать обратную связь о проблемах сразу после их возникновения.

practicum.yandex.ruedu.sravni.ru

Инструменты

Для тестирования API используют, например:

- **Postman** — сервис для создания, отправки и тестирования HTTP-запросов, позволяет создавать автоматизированные тесты.
- **REST-assured** — Java-библиотека, которая позволяет создавать запросы, проверять ответы и генерировать автоматизированные тесты с использованием DSL-синтаксиса.
- **JMeter** — сервис по отправке HTTP-запросов для тестирования масштабируемости и производительности API в условиях высокой нагрузки.

edu.sravni.ruru.hexlet.io

Примеры заданий

Некоторые практические задания по тестированию API:

- **Написание тестовых сценариев** — дать спецификацию API и попросить составить набор тест-кейсов, покрывающих функциональность, граничные условия и негативные сценарии.
- **Создание и выполнение запросов в Postman** — задание на создание коллекции запросов для тестирования определённого API, настройку авторизации, переменных окружения и написание тестов.
- **Анализ и отладка API-запросов** — предоставить логи или трассировку API-запросов и попросить найти проблему или неоптимальное взаимодействие.
- **Автоматизация API-тестов** — написание скриптов для автоматизации тестирования API с использованием REST Assured, Pytest или других фреймворков.

sky.pro

Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

Содержание отчёта

1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

Ответить на контрольные вопросы. (устно)

Сформулировать выводы по результатам работы.

Сдать и защитить работу.

Практическая работа № 10. Мониторинг производительности ИИ-модели с использованием систем мониторинга и оповещения и мониторинга и визуализации данных.

Цель работы: обеспечить надёжность и эффективность развёрнутых систем ИИ, выявить проблемы (например, снижение производительности, дрейф данных) и разработать стратегии их устранения.

Задачи:

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

Оборудование, технические средства и инструменты:

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

Ход практического занятия:

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями, студенты выполняют практическую работу.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

Теоретические и учебно-методические материалы по теме практической работы

- **Специализированные инструменты** для мониторинга моделей ИИ. ultralytics.commlog.uz
- **Механизмы оповещения** об отклонениях показателей производительности. corporatefinanceinstitute.com

- **Инструменты для визуализации данных** при мониторинге. dzen.ru

Системы мониторинга

Некоторые инструменты для мониторинга производительности ИИ-модели:

- **Библиотеки с открытым исходным кодом** — например, TensorFlow Data Validation (TFDV), Evidently AI и Deepchecks. Предоставляют функциональные возможности для валидации данных и моделей, обнаружения дрейфа и мониторинга производительности.
- **Облачные платформы** — предлагают управляемые сервисы для мониторинга моделей, например, Amazon SageMaker Model Monitor, Azure Machine Learning Model Monitoring.
- **Коммерческие платформы** — предоставляют комплексные решения для мониторинга моделей, например, Arize AI, Fiddler AI и WhyLabs.

mlog.uz

Оповещения

Механизмы оповещения при мониторинге производительности ИИ-модели могут включать:

- **Автоматические оповещения** об отклонениях ключевых показателей производительности (точность, отзыв, F1-score). Например, оповещение о снижении точности модели из-за дрейфа данных. ultralytics.comcorporatefinanceinstitute.com
- **Интеллектуальную корреляцию** — систему, которая устанавливает причинно-следственные связи между событиями, группирует схожие оповещения и определяет логические зависимости. artimate.ru

Визуализация данных

Для визуализации данных при мониторинге производительности ИИ-модели можно использовать:

- **TensorBoard** — позволяет отслеживать метрики обучения (потери, точность) на всех этапах обучения, что помогает определить моменты переобучения или недообучения модели.
- **Visdom** — подходит для создания динамичных и интерактивных визуализаций, что упрощает мониторинг процесса обучения моделей в реальном времени.
- **Model Explorer от Google** — инструмент для интуитивного исследования архитектур моделей, позволяет детально исследовать структуру и особенности модели.

dzen.ru

Кэмп-нейросеть для студентов

Платформа отлично подходит как для самостоятельного обучения, так и для развития gptwriter.ru

Промо

Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

Содержание отчёта

1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению

и кодированию.

Ответить на контрольные вопросы. (устно)
Сформулировать выводы по результатам работы.
Сдать и защитить работу.

Критерии оценки

Оценка «отлично» ставится в том случае, если студент:

- свободно применяет полученные знания при выполнении практических заданий;
- выполнил работу в полном объеме с соблюдением необходимой последовательности действий;
- в письменном отчете по работе правильно и аккуратно выполнены все записи;
- при ответах на контрольные вопросы правильно понимает их сущность, дает точное определение и истолкование основных понятий, использует специальную терминологию дисциплины, не затрудняется при ответах на видоизмененные вопросы, сопровождает ответ примерами.

Оценка «хорошо» ставится, если:

- выполнены требования к оценке «отлично», но допущены 2 – 3 недочета при выполнении практических заданий и студент может их исправить самостоятельно или при небольшой помощи преподавателя;
- в письменном отчете по работе делает незначительные ошибки;
- при ответах на контрольные вопросы не допускает серьезных ошибок, легко устраняет отдельные неточности, но затрудняется в применении знаний в новой ситуации, приведении примеров.

Оценка «удовлетворительно» ставится, если:

- практическая работа выполнена не полностью, но объем выполненной части позволяет получить правильные результаты и выводы;
- в ходе выполнения работы студент продемонстрировал слабые практические навыки, были допущены ошибки;
- студент умеет применять полученные знания при решении простых задач по готовому алгоритму;
- в письменном отчете по работе допущены ошибки;
- при ответах на контрольные вопросы правильно понимает их сущность, но в ответе имеются отдельные пробелы и при самостоятельном воспроизведении материала требует дополнительных и уточняющих вопросов преподавателя.

Оценка «неудовлетворительно» ставится, если:

- практическая работа выполнена не полностью и объем выполненной работы не позволяет сделать правильных выводов, у студента имеются лишь отдельные представления об изученном материале, большая часть материала не усвоена;
- в письменном отчете по работе допущены грубые ошибки, либо он вообще отсутствует;
- на контрольные вопросы студент не может дать ответов, так как не овладел основными знаниями и умениями в соответствии с требованиями программы.