


Смоленский колледж телекоммуникаций (филиал)
федерального государственного бюджетного образовательного учреждения
высшего образования
«Санкт-Петербургский государственный университет телекоммуникаций
им. проф. М.А. Бонч-Бруевича»

УТВЕРЖДАЮ

Заместитель директора по УР

 И.А. Овчинникова

« 14 » 05 2025 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТАМИ

по МДК.01.03. Тестирование программных модулей

специальность: 09.02.13 Интеграция решений с применением технологий
искусственного интеллекта

преподаватель: Котяткина Анастасия Николаевна

форма обучения – очная

Составлены в соответствии с рабочей программой, утвержденной «14» 05. 2025 г.

Рассмотрены на заседании методической комиссии
гуманитарных и программно-вычислительных дисциплин

Протокол № 10 от «14» 05. 2025 г.

Председатель МК  Т.Н. Строде

Методист  О.Г. Ряска

г. Смоленск, 2025

Содержание

| | | |
|---|--|---|
| 1 | Пояснительная записка | 3 |
| 2 | Особенности организации внеаудиторной самостоятельной работы студентов | 3 |
| 3 | Виды самостоятельной работы по МДК.01.03. Тестирование программных модулей | 4 |
| | Приложения | 7 |

1. Пояснительная записка

Самостоятельная работа – это планируемая работа студентов, выполняемая по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Она предназначена не только для овладения дисциплиной, но и для формирования навыков самостоятельной работы вообще, в учебной, научной, профессиональной деятельности, способности принимать на себя ответственность, самостоятельно решать проблемы, находить конструктивные решения, выход из кризисной ситуации и т.д. Таким образом, значимость самостоятельной работы студента выходит далеко за рамки отдельной дисциплины, играя существенную роль в развитии самостоятельности как черты характера, личностного качества, выраженного в способности мыслить, анализировать ситуации, вырабатывать собственное мнение, действовать по собственной инициативе, независимо от навязываемых взглядов.

Продумывая формы организации самостоятельной работы по дисциплине, преподаватель должен исходить из нескольких позиций:

- необходимые знания, умения и навыки, которые должен показать студент в результате выполнения всех заданий, выносимых на самостоятельное изучение (в соответствии с целью и задачами изучаемой дисциплины);
- формирование профессиональных компетентностей, которые должны проявиться через ЗУНы (знания, умения и навыки);
- формирование креативности студента в процессе изучения дисциплины и способности нестандартно мыслить при выполнении заданий для самостоятельной работы;
- развитие активной исследовательской позиции студента;
- воспитание чувства ответственности за своевременное выполнение задания.

Методические указания и рекомендации позволяют студенту выявить главное и второстепенное в изучаемой дисциплине, увидеть связь теории и практики, развивают способность к анализу полученных результатов, формируют способность формулировать тактические подходы к выполнению поставленных задач, например, подготовке к сдаче зачетов, экзаменов.

Таким образом, самостоятельная работа студентов способствует развитию у них творческой активности, повышению компетентности, совершенствованию мыслительных навыков, а также воспитывает личность будущего профессионала.

Студент, приступающий к изучению МДК 01.03 «Тестирование программных модулей», получает информацию обо всех видах самостоятельной работы, об объеме и видах самостоятельной работы. Перед выполнением студентами самостоятельной внеаудиторной работы преподаватель проводит инструктаж по выполнению задания, который включает: цель задания, его содержание, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки.

2. Особенности организации внеаудиторной самостоятельной работы студентов

При предъявлении видов заданий на внеаудиторную самостоятельную работу рекомендуется использовать дифференцированный подход к студентам. Перед выполнением студентами внеаудиторной самостоятельной работы преподаватель проводит инструктаж по выполнению задания, который включает:

- цель задания,
- содержание,
- сроки выполнения,
- ориентировочный объем работы,
- основные требования к результатам работы,
- критерии оценки.

В процессе инструктажа преподаватель предупреждает студентов о возможных типичных ошибках, встречающихся при выполнении задания. Инструктаж проводится преподавателем за счет объема времени, отведенного на изучение дисциплины.

Во время выполнения студентами внеаудиторной самостоятельной работы и при необходимости преподаватель может проводить консультации за счет общего бюджета времени, отведенного на консультации.

Самостоятельная работа может осуществляться индивидуально или группами студентов в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня умений студентов.

Контроль результатов внеаудиторной самостоятельной работы студентов может осуществляться в пределах времени, отведенного на обязательные учебные занятия по дисциплине и внеаудиторную самостоятельную работу студентов по дисциплине, может проходить в письменной, устной или смешанной форме, с представлением изделия или продукта творческой деятельности студента.

В качестве форм и методов контроля внеаудиторной самостоятельной работы студентов могут быть использованы: тестирование, защита практических и лабораторных занятий, письменная проверка и др.

3. Виды самостоятельной работы по МДК.01.03. Тестирование программных модулей.

На самостоятельную работу студентов по МДК.01.03. Тестирование программных модулей РУП выделено 8 часов.

Тема 1. Тестирование ИИ-модулей и систем

Цель ВСП: сформировать у студентов практические навыки планирования, проведения и анализа результатов комплексного тестирования ИИ-модели. Студенты научатся выявлять не только стандартные метрики качества, но и оценивать надежность, устойчивость и этические аспекты работы ИИ-системы.

Трудоемкость

| Количество заданий (задач, упражнений) | Характер задачи (обязательный/рекомендательный) | Норма времени (в часах по рабочей программе) | Срок выполнения (в неделях) | Форма представления материала | Форма контроля каждого задания |
|--|---|--|-----------------------------|---|------------------------------------|
| Задание | Обязательный | 8 | 2 недели | Аналитический отчет в формате pdf; Jupyter Notebook со всем исходным кодом, комментариями и результатами выполнения ячеек; наборы сгенерированных данных (например, .csv с adversarial-примерами), если они были созданы в процессе работы. | Устный опрос, демонстрация проекта |

Задание. Разработать систему для автоматического анализа тональности отзывов о товарах в интернет-магазине. Ваша задача — протестировать новую модель классификации текста, которая присваивает отзывам одну из трех категорий:

- POSITIVE (положительный);
- NEUTRAL (нейтральный);
- NEGATIVE (отрицательный).

Для работы предоставляется:

Обученная модель: Доступ к API эндпоинту или Python-скрипт с предобученной моделью (например, на основе BERT или RandomForest с TF-IDF). Для упрощения, модель может быть заранее подготовлена преподавателем.

Тестовый датасет: Набор из 500 размеченных отзывов (test_data.csv), который не использовался при обучении модели.

Набор для проверки устойчивости: Дополнительный небольшой набор текстов, специально подобранный для проверки на смещения и уязвимости.

Этапы выполнения работы:

Этап 1: Планирование и подготовка.

Задача 1.1: Ознакомьтесь с техническим заданием на модель. Определите и задокументируйте в отчете ключевые требования к системе (например, целевая точность >85%, время отклика < 1 секунды).

Задача 1.2: Разработайте План тестирования, который будет включать следующие виды тестов: Функциональное тестирование (точность, полнота, F1-мера).

Тестирование производительности (время инференса, нагрузочное тестирование).

Тестирование надежности и устойчивости (проверка на adversarial-атаках, данных со сдвигом).

Этическое тестирование (проверка на смещения - bias).

Этап 2: Функциональное и метрическое тестирование.

Задача 2.1: Напишите скрипт, который отправит тестовый датасет на модель и получит предсказания.

Задача 2.2: Рассчитайте стандартные метрики классификации:

Матрица ошибок (Confusion Matrix)

Accuracy, Precision, Recall, F1-мера для каждого класса.

Задача 2.3: Проанализируйте результаты. На каких классах модель ошибается чаще всего? Сформулируйте гипотезы, почему это происходит.

Этап 3: Тестирование производительности.

Задача 3.1: Проведите замер времени отклика модели на одном примере и на батче из 100 примеров.

Задача 3.2: (Опционально, если позволяет инфраструктура) Напишите простой скрипт для нагрузочного тестирования, отправляющий 1000 последовательных запросов. Постройте график времени ответа и определите, есть ли деградация производительности.

Этап 4: Тестирование надежности и устойчивости.

Задача 4.1: Adversarial-атаки. Создайте модифицированные версии корректных отзывов из тестового набора, чтобы "обмануть" модель. Используйте простые методы:

Опечатки (очень вместо очень).

Синонимизацию ключевых слов.

Добавление случайных символов.

Протестируйте модель на этих примерах. Насколько она устойчива?

Задача 4.2: Data Shift. Протестируйте модель на данных из другой предметной области (например, отзывы о ресторанах вместо электроники). Сравните метрики. Сделайте вывод о способности модели к обобщению.

Этап 5: Этическое тестирование и проверка на смещения (Bias).

Задача 5.1: Проанализируйте, есть ли в вашем тестовом наборе признаки, которые могут привести к смещению модели (например, имена людей, упоминание определенных брендов, гендерные или культурные маркеры).

Задача 5.2: Создайте парные примеры (Parity Testing). Например:

"Этот телефон купил [Имя_1]. Он отличный."

"Этот телефон купил [Имя_2]. Он отличный."

Замените [Имя_1] и [Имя_2] на имена, ассоциируемые с разными этническими группами.

Предсказывает ли модель одинаковый sentiment для обеих фраз? Задокументируйте найденные случаи смещения.

Этап 6: Оформление отчета и выводы.

Задача 6.1: Оформите итоговый отчет, который должен содержать:

Титульный лист.

Введение (постановка задачи).

План тестирования.

Результаты по каждому этапу с графиками, таблицами и кодом (где уместно).

Сводную таблицу с найденными "багами" и уязвимостями модели, оцененной их критичности.

Заключение с общим вердиктом о готовности модели к промышленной эксплуатации и рекомендациями по ее улучшению.

Критерии оценки:

| Критерий | Вес, % | Описание |
|--|--------|--|
| Полнота и глубина проведения тестов | 30% | Выполнены все виды тестов из плана, методы атак и проверок реализованы корректно. |
| Качество анализа и интерпретации результатов | 30% | Результаты не просто представлены, но и проанализированы. Сформулированы содержательные гипотезы о причинах ошибок и смещений. |
| Оформление отчета | 20% | Отчет структурирован, логичен, содержит все требуемые разделы. Результаты визуализированы. |
| Техническая реализация | 20% | Код написан, хорошо документирован и работает корректно. Используются адекватные библиотеки (pandas, scikit-learn, matplotlib, transformers и т.д.). |

Рекомендуемые инструменты и технологии

Язык программирования: Python.

Библиотеки: pandas, numpy, scikit-learn (для метрик), matplotlib/seaborn (для визуализации), requests (для работы с API), transformers/torch/tensorflow (если модель локальная).

Для adversarial-атак: библиотеки textattack или foolbox, либо самостоятельная реализация простых методов.

Для оформления: Jupyter Notebook (как исполнительная среда) и PDF-отчет (как финальный результат).

Информационное обеспечение:

1. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение. – М.: ДМК Пресс, 2018. – 652 с.
2. Бринк Х., Ричардс Дж., Феверолф М. Машинное обучение для бизнеса. – СПб.: Питер, 2018. – 256 с.
3. Смирнов И. В., Орлов А. С. Обзор методов повышения устойчивости глубоких нейронных сетей к состязательным атакам // Информатика и системы управления. – 2022. – № 2(72). – С. 76–89.
4. Adversarial-атаки и защита от них: полное руководство [Электронный ресурс] // Хабр. – URL: <https://habr.com/ru/...> (дата обращения: 25.10.2023).
5. Метрики в задачах машинного обучения: полный разбор [Электронный ресурс] // Яндекс.Дзен. – URL: <https://dzen.ru/...> (дата обращения: 25.10.2023).
6. Официальная документация библиотеки scikit-learn [Электронный ресурс]. – URL: https://scikit-learn.org/stable/modules/model_evaluation.html (дата обращения: 25.10.2023).

Требования к оформлению отчета:

1. Итоговый отчет в формате PDF

Рекомендуемая структура отчета

1. Титульный лист:

- Название ВУЗа, факультета, группы.
- ФИО студента.
- Название задания: «Отчет по тестированию ИИ-модуля анализа тональности».
- Дисциплина, дата.

2. Аннотация / Краткое резюме (на 1 абзац):

- Основная цель работы.
- Ключевые выводы: готова ли модель к эксплуатации, ее основные сильные и слабые стороны.

3. Введение:

- Постановка задачи: что тестировали (модуль анализа тональности) и зачем.
- Описание тестируемой системы (API или скрипт модели).
- Цели и объем тестирования.

4. План тестирования:

- Подробное описание всех запланированных видов тестов (как из задания) и использованных методик.

5. Результаты тестирования:

- **Функциональное тестирование:** Привести матрицу ошибок, таблицу с рассчитанными метриками (Accuracy, Precision, Recall, F1 по классам). Графики (например, столбчатая диаграмма метрик). Гипотезы по ошибкам.
- **Тестирование производительности:** Таблица с замерами времени отклика. График нагрузочного тестирования (если проводилось).
- **Тестирование надежности и устойчивости:** Примеры adversarial-атак и результаты модели на них (таблица: "Оригинальный текст", "Модифицированный текст", "Истинный label", "Предсказанный label"). Анализ устойчивости. Результаты теста на Data Shift.
- **Этическое тестирование:** Примеры парных текстов, использованных для проверки на смещения, и ответы модели на них. Выводы о наличии/отсутствии смещений.

6. Сводная таблица дефектов:

Оформить найденные проблемы в виде таблицы.

Например:

| ID дефекта | Критичность (Высокая/Средняя/Низкая) | Описание дефекта | Шаги для воспроизведения | Ожидаемый результат | Фактический результат |
|------------|--------------------------------------|--|--|----------------------|-----------------------|
| BIAS-001 | Средняя | Модель меняет sentiment при замене имени "Андрей" на "Мухаммед" в том же контексте | Подать на вход два парных примера | Одинаковый sentiment | Sentiment различается |
| ADV-005 | Низкая | Модель ошибается при добавлении 2-3 опечаток в слово "отличный" | Подать текст: "Этот отльичный телефон" | POSITIVE | NEUTRAL |

7. Заключение и рекомендации:

Вердикт: Общая оценка готовности модели к промышленному использованию (например: "Модель может быть выпущена с ограничениями...", "Модель требует доработки...").

Рекомендации разработчикам: Конкретные предложения по улучшению модели (например: "Добавить в обучающие данные больше примеров с нейтральными отзывами", "Реализовать предобработку текста для исправления опечаток", "Провести дополнительное обучение на данных из смежной предметной области").

2. Исходный код и скрипты (Исполнительные артефакты)

Форма предоставления: Jupyter Notebook (.ipynb)

Структура Jupyter Notebook:

1. Импорт библиотек и настройка окружения.

2. Загрузка данных (тестового датасета и др.).

3. Блок "Функциональное тестирование":

- Код для взаимодействия с моделью (вызов API или функции).

- Код для расчета и визуализации метрик.

4. Блок "Тестирование производительности":

- Код для замера времени.

5. Блок "Тестирование надежности":

- Функции для генерации adversarial-примеров.

- Код для проведения тестов.

6. Блок "Этическое тестирование":

- Код для создания парных примеров и проверки модели.

7. (Опционально) Блок для генерации данных для отчета: Код, который создает CSV-файлы или картинки графиков, которые потом будут вставлены в PDF-отчет.

Общие формальные требования к отчету в формате PDF.

| Параметр | Требование | Комментарий |
|----------------------|---|---|
| Основной шрифт | Times New Roman, 14pt (для основного текста). | |
| Шрифт моноширинный | Consolas, Courier New, 12pt. | Используется для всего кода, URL, имен файлов и переменных в основном тексте. |
| Межстрочный интервал | 1.5 строки. | |
| Поля страницы | Не менее 2 см со всех сторон (левое — 3 см для подшивки). | |
| Выравнивание текста | По ширине. | Заголовки можно выравнивать по центру. |
| Абзацный отступ | 1.25 см. | |
| Нумерация страниц | Сквозная, внизу страницы (в центре или справа). | Титульный лист включается в общую нумерацию, но номер на |

| Параметр | Требование | Комментарий |
|----------------|--|------------------|
| | | нем не ставится. |
| Название файла | Отчет_АИ_Тестирование_ФИО_Группа.pdf Пример: Отчет_АИ_Тестирование_Иванов_ИИ_ПД-401.pdf | |

Детализированные требования по разделам отчета

1. Титульный лист

Название учебного заведения, отделения — 14pt, полужирный, по центру.

Название работы — 16pt, ПРОПИСНЫЕ, полужирный, по центру.

Дисциплина, специальность — 14pt, по центру.

Данные студента (ФИО, группа) и преподавателя — 14pt, выравнивание по правому краю.

Город и год — 14pt, по центру внизу.

2. Структура и заголовки

Все разделы (Введение, План тестирования, Заключение и т.д.) нумеруются арабскими цифрами (1, 2, 3...).

Заголовки разделов (уровня 1) — 16pt, полужирный, выравнивание по центру. Отступ перед и после заголовка — 12pt.

Подразделы (уровня 2, например, "3.1. Функциональное тестирование") — 14pt, полужирный, выравнивание по левому краю.

3. Оформление иллюстраций (графиков, диаграмм)

Все иллюстрации должны быть пронумерованы и подписаны.

Формат подписи: Рис. 1.1 — Матрица ошибок для тестового набора данных

Подпись располагается под рисунком, выравнивается по центру, шрифт 12-14pt.

На все рисунки должны быть ссылки в тексте (например: "...как видно на Рис. 1.1, модель чаще всего путает классы NEGATIVE и NEUTRAL").

4. Оформление таблиц

Все таблицы должны быть пронумерованы и озаглавлены.

Формат подписи: Таблица 2.1 — Сводная таблица метрик производительности

Подпись располагается над таблицей, выравнивается по левому краю, шрифт 12-14pt.

Шапка таблицы может быть выделена полужирным начертанием.

На все таблицы должны быть ссылки в тексте.

5. Оформление кода и технических терминов

Фрагменты кода в тексте (например, название функции `predict_sentiment()`) заключаются в обратные апострофы (') и набираются моноширинным шрифтом (Consolas, 12pt).

Большие блоки кода (например, ключевые функции из Notebook), если их необходимо привести в отчете, оформляются как отдельный блок с отступами, моноширинным шрифтом 12pt и серым фоном.

6. Стиль письма

Отчет должен быть написан в безличной или научной форме.

Недопустимо: "Я протестировал модель и увидел, что..."

Рекомендуется: "В ходе тестирования было установлено, что...", "Результаты показывают, что...", "Анализ матрицы ошибок выявил..."

Пример фрагмента текста, соответствующего требованиям.

1.3. Результаты функционального тестирования

В рамках функционального тестирования был проведен прогон тестового набора данных, состоящего из 500 размеченных отзывов, через модель классификации. Для оценки качества работы модели использовались метрики, приведенные в Таблице 2.1.

Таблица 2.1 — Метрики качества модели на тестовом наборе

| Класс | Precision | Recall | F1-Score |
|----------|-----------|--------|----------|
| POSITIVE | 0.92 | 0.85 | 0.88 |
| NEUTRAL | 0.78 | 0.81 | 0.79 |
| NEGATIVE | 0.87 | 0.90 | 0.88 |

Как видно из таблицы, наименьшие значения метрик наблюдаются для класса `NEUTRAL`. Это может быть связано с меньшим количеством примеров данного класса в обучающей выборке или с субъективной сложностью разметки нейтральных отзывов.

Для детального анализа ошибок была построена матрица ошибок (Рис. 2.1).

Рис. 2.1 — Матрица ошибок модели классификации тональности

[ВСТАВИТЬ ИЗОБРАЖЕНИЕ МАТРИЦЫ ОШИБОК]

Анализ матрицы ошибок подтверждает, что модель чаще всего путает класс `NEUTRAL` с классами `POSITIVE` и `NEGATIVE` (по 15 и 12 случаев соответственно).