Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

Студент **ДИНЬ ВЬЕТ АНЬ**
фамилия, имя, отчество

Группа **ИУ7И-84Б**

Тип практики **ПРЕДДИПЛОМНАЯ**

Название предприятия **НУК ИУ МГТУ им. Н. Э. Баумана**

Студент _____ **Динь Вьет Ань**
подпись, дата *фамилия, и.о.*

Руководитель практики _____ **Кострицкий А. С.**
подпись, дата *фамилия, и.о.*

Оценка _____

2024 г.

«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой ИУ7
_____ И. В. Рудаков
« 10 » мая 2024 г.

З А Д А Н И Е
на прохождение производственной практики

Преддипломная практика

Студент

Динь Вьет Ань 4 курса группы **ИУ7И-84Б**

в период **с 13. мая.2024 г. по 25. мая.2024 г.**

Предприятие: **НУК ИУ МГТУ им. Н. Э. Баумана**

Руководитель практики от предприятия (наставник):
Кострицкий Александр Сергеевич, старший преподаватель
(Фамилия Имя Отчество полностью, должность)
Руководитель практики от кафедры:
Кострицкий Александр Сергеевич, старший преподаватель
(Фамилия Имя Отчество полностью, должность)

Задание:
1. Выбрать средства программной реализации, спроектированного в ходе выполнения выпускной квалификационной работы метода.
2. Реализовать программное обеспечение метода.
3. Провести исследование характеристик разработанного программного обеспечения.

Дата выдачи задания « 10 » мая 2024 г.

Руководитель практики от предприятия **_____/Кострицкий А. С./**

Руководитель практики от кафедры **_____/Кострицкий А. С./**

Студент **_____/ Динь Вьет Ань /**

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Во время выполнения выпускной квалификационной работы был разработан метод классификации новостных текстов по тематике с помощью метода опорных векторов.

Целью данной работы является разработка программного обеспечения, демонстрирующего практическую осуществимость спроектированного в ходе выполнения выпускной квалификационной работы метода.

Для достижения поставленной цели необходимо выполнить следующие задачи:

— выбрать средства программной реализации спроектированного в ходе выполнения выпускной квалификационной работы метода;

— реализовать программное обеспечение метода;

— провести исследование характеристик разработанного программного обеспечения.

1 Основная часть

1.1. Выбор языка программирования

Для реализации программного обеспечения был использован язык программирования Python[1]. Этот выбор обусловлен следующими причинами:

— это язык программирования высокого уровня, имеет простой синтаксис;

— наличие библиотек с открытым исходным кодом, позволяющих работать с естественноым языком, выпониять классификации методом SVM и визуализировать данные;

— имеется навыки использования данного языка программирования, что сократить время написания программы.

1.2. Выбор среды программирования

Для разработки программы использовалась среда Visual Studio Code [2] в качестве среды разработки по следующим причинам:

— доступна бесплатная версия;

— имеет множество утилит, упрощающих написание кода;

— имеется навыки программирования при помощью данной среды, что сократить время написания программы.

Библиотека scikit-learn используется для работы с машинным обучением, потому что она предоставляет множество различных инструментов для разных задач машинного обучения, включая классификацию, регрессию, кластеризацию, обучение и тестирование моделей, а также предварительную обработку данных, и так далее.

Для создания пользовательского интерфейса используется фреймворк Qt с помощью библиотеки PyQt5. Кроме того, в процессе разработки программного обеспечения библиотека pandas также используется для анализа набора данных, а библиотеки nltk и pymorphy2 — для очистки и предварительной обработки текстов.

1.3. Формат входных и выходных данных

Для модуля создания выборки данных входными данными является набор данных, скачанный с сайта kaggle.com, а выходные данные представляют собой файлы с указанным количеством строк для каждой темы.

На этапе обучения классификатора входными данными является файл, который создается из набора новостных текстов, в формате файла CSV. Каждая строка соответствует новости, состоящему из двух частей: текста и тематика новости. Каждый файл содержит определенное количество текста по каждой тематику.

На этапе классификации входными данными является текст или текстовый файл (в формате файла TXT), содержащий текст. Текст должен быть на русском языке, не менее 10 слов.

Результатом модуля классификации является прогнозируемое название тематика входного текста или содержимого файла и выводятся на пользовательский интерфейс программного обеспечения.

1.4. Руководство пользователя

Для запуска разработанного программного обеспечения требуется установить интерпретатор для Python и используемые библиотеки. Используемые в разработке библиотеки, которые необходимы для запуска ПО, приведены в файле requirements.txt, который находится в корневом каталоге проекта. С помощью пакетного менеджера pip все зависимости можно установить или обновить, запустив в терминале команду, приведенную в листинге 1.1.

Листинг 1.1 – Установка всех необходимых библиотек

```
pip install -r requirements.txt
```

Для работы библиотеки nltk необходимо скачать библиотеки. Для установки библиотеки nltk и скачивания библиотек нужно в коде программы выполнить команды, приведенные в листинге 1.2

Листинг 1.2 – Установка словарей nltk

```
import nltk
nltk.download()
```

Для создания выборки необходимо разместить папку с файлом исходного набора данных внутри корневого каталога проекта и запустить скрипт dataset.py.

Скрипт может запущен из графического интерфейса среды разработки, либо командой из терминала, приведенную в листинге 1.3.

Листинг 1.3 – Команда для создания обучающих выборок

```
python dataset.py
```

Чтобы обучить классификатор на заранее созданных размеченных выборках и предсказать тематик входного текста или текстового файла, пользователь может воспользоваться графическим интерфейсом приложения. Для открытия приложения необходимо запустить скрипт main.py посредством интерфейса среды разработки или командой в терминале, приведенную в листинге 1.4.

Листинг 1.4 – Команда для запуска приложения

```
python main.py
```

В результате разработанное программное обеспечение может установить тематик текста или выдвинуть предположение, что может быть тематиком.

1.5. Интерфейс пользователя

Пользовательский интерфейс, который представлен на рисунке 1.1, был разработан с помощью программы Qt Designer. Интеграция с кодом на Python осуществлена посредством с помощью библиотеки PyQt5, которая предоставляет различные классы для работы с объектами пользовательского интерфейса.

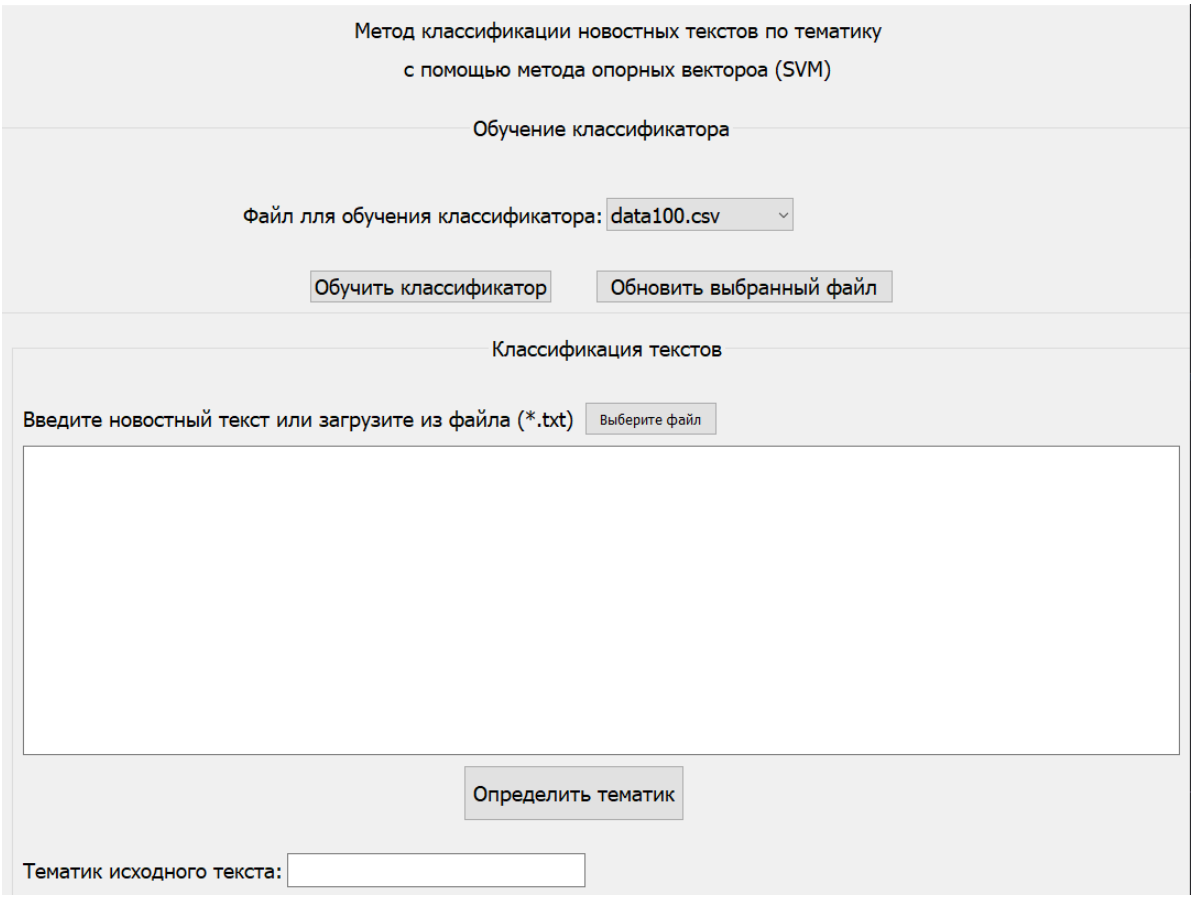

Рис. 1.1 – Пользовательского интерфейса ПО

Пользовательский интерфейс программного обеспечения состоит из 2 частей. В верхней части находится функционал обучения классификатора. В этом окне пользователь может выбрать файл для обучения классификатора из поля выбора. Каждый файл содержит определенное количество текста по каждой тематику. Пользователь может обновить содержимое этих файлов с помощью кнопки «Обновить выбранного файла». При нажатии на кнопку «Обучить классификатор» выбранная выборка разбивается на две части: 80% корпуса — обучающая выборка, 20% — тестовая. Программа сообщит, когда классификатор завершит обучение, как показано на рисунке 1.2.

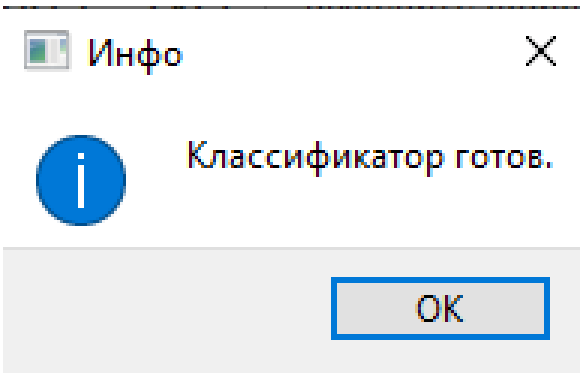

Рис. 1.2 – Классификатор готов

В нижней части находится функционал предсказания тематика текста. Если пользователь попытается определить тематик, не обучив сначала классификатор, будет возвращена ошибка, как показано на рисунке 1.3.

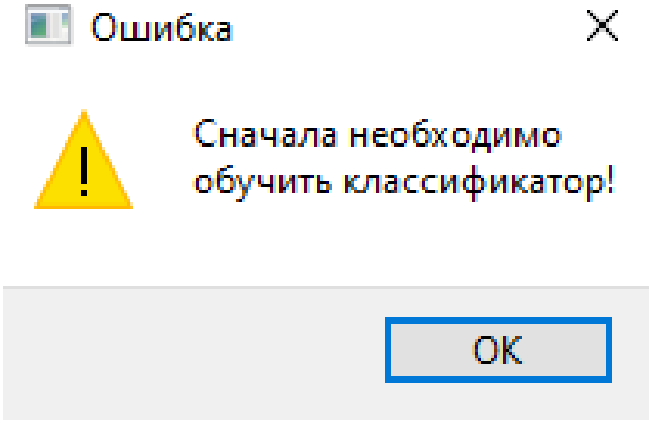

Рис. 1.3 – Ошибка, возникающая в случае, если классификатор не обучен

Пользователь может ввести текст в поле ввода или выбрать файл. Программа поддерживает только txt файлы (текстовые файлы). При нажатии на кнопку «Выбрать файл» открывается файловая система компьютера, в которой необходимо выбрать интересующий текстовый файл. Когда пользователь выбирает файл, его содержимое отображается в текстовом поле. Если пользователь нажимает кнопку «Определить тематик», когда вводимый текст пуст или имеет неверный формат, будет возвращена ошибка, как показано на рисунке 1.4.

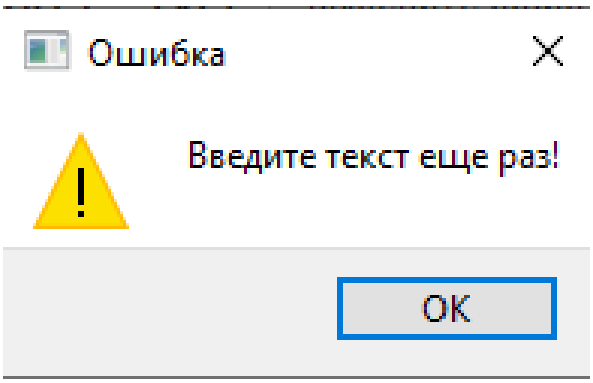

Рис. 1.4 – Ошибка, возникающая в случае, если текстовое поле пусто

Если входный текст не пуст, то запускается алгоритм классификации. После того, как классификатор завершит работать, на экран будет выведено название наиболее вероятного тематика, как показано на рисунке 1.5.

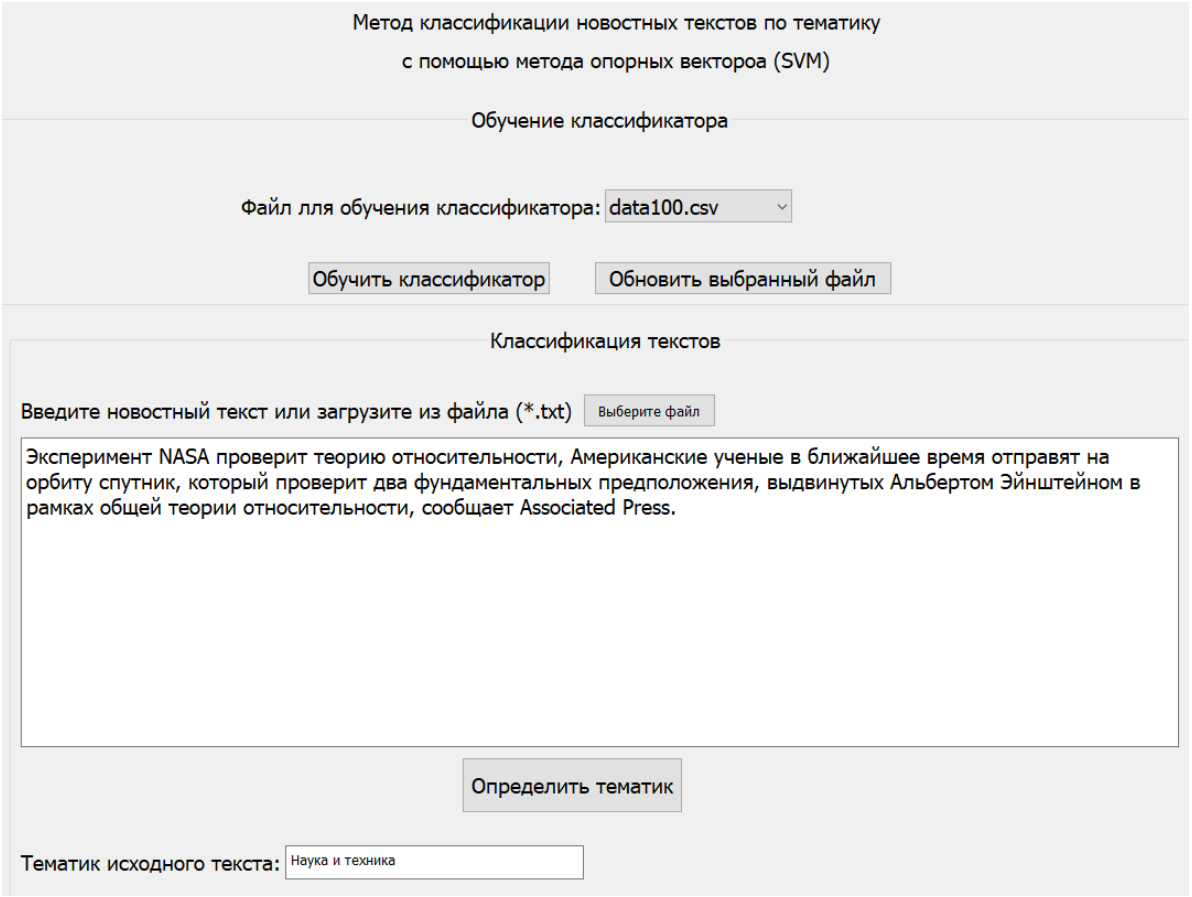

Рис. 1.5 – Результат работы классификации

1.6. Технические характеристики

Технические характеристики устройства, на котором выполнялись исследования разработанного метода:

— операционная система Window 10 Home Single Language;

— память 8 Гб;

— процессор 11th Gen Intel(R) Core(TM) i5-1135G7 2.42 ГГц, 4 ядра.

Во время выполнения исследований устройство было подключено к сети электропитания, нагружено приложениями окружения и самой системой замера.

Для проведения исследований разработанного метода используются выборки, созданные на основе текстов по 5 темам: Наука и техника, Спорт, Культура, Экономика и Мир. Количество текста по каждой тематике в каждой выборке используется одинаковое. Исходная выборка разбивается на обучающую (80% объема) и тестовую (20%) выборки.

Как отмечалось ранше, метрики аккуратности и F1-меры используются в качестве метрики для оценки качества классификатора.

1.7. Влияние размера обучающей выборки

Обучающая выборка — это набор данных, который используется для обучения модели машинного обучения. Она представляет собой подмножество общего набора данных, которое содержит примеры, сопоставленные с соответствующими целевыми значениями или метками. Размер обучающей выборки — фактор, влияющий на качество классификатора.

Для проведения данного исследования было проведено сравнение зависимости времени обучения и качества классификатора от количества текстов на обучающей выборке.

Результаты проведенного исследования приведены в таблицах 1.1 и 1.2.

Таблица 1.1 – Замеры времени обучения от размера выборки

Количество текстов в выборке	*Времени обучения классификатора (с.)*
50	0.03
100	0.08
200	0.23
500	1.59
1000	5.67
5000	135.10

Таблица 1.2 – Зависимость качества классификатора от размера выборки

Количество текстов в выборке	*Точность на тестовой выборке*	*F1-мера*
50	0.2	0.27
100	0.9	0.89
200	0.9	0.90
500	0.92	0.92
1000	0.92	0.92
5000	0.943	0.94

Ниже, на рисунках 1.6 и 1.7, представлены данные из таблиц 1.1 и 1.2 соответственно в виде графика.

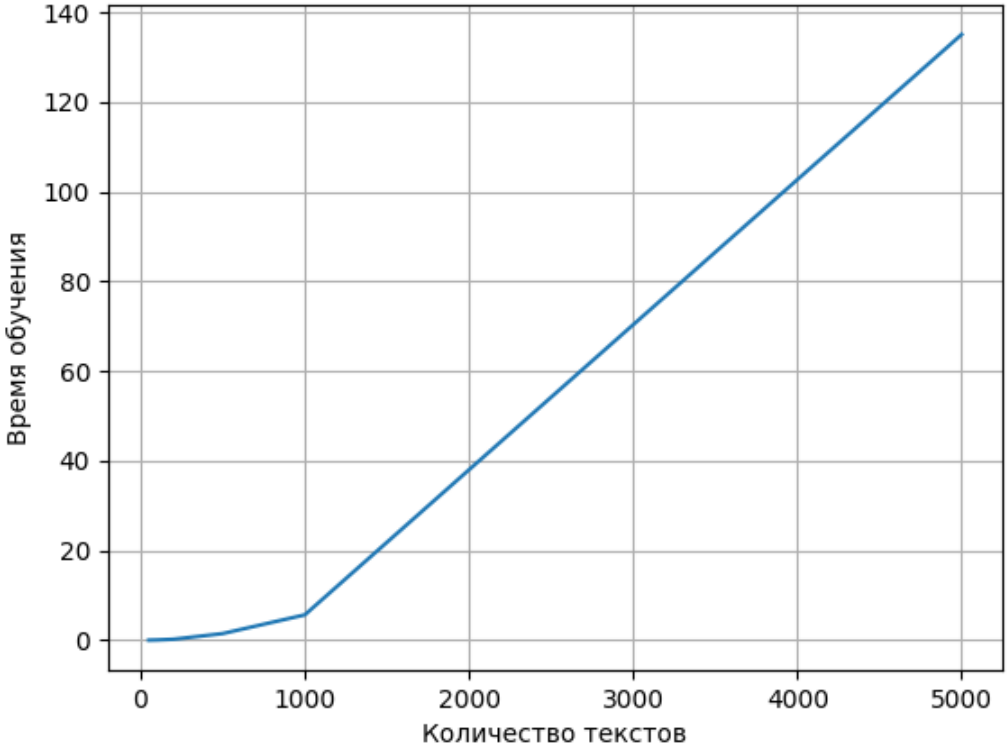

Рис. 1.6 – График зависимости времени обучения от размера выборки

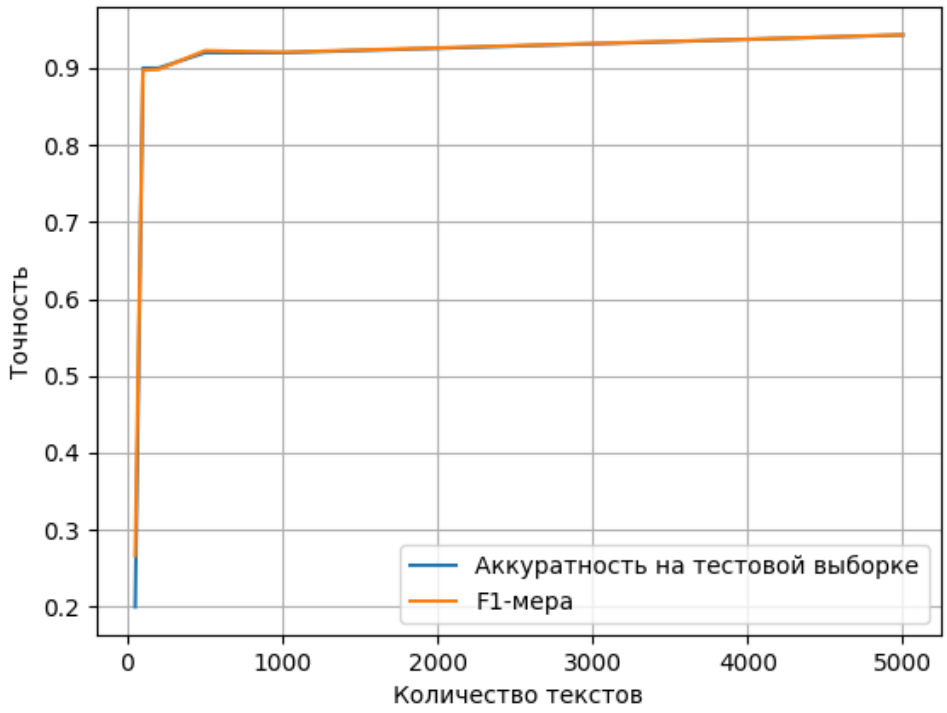

Рис. 1.7 – График зависимости качества классификатора от размера выборки

На основании полученных результатов можно сделать вывод, что количество текстов в обучающей выборке влияет на время обучения и качество клас-

сификатора. При использовании выборок с меньшим количеством текста классификатор будет обучаться за меньшее время, но с низкой точностью. По мере увеличения количества текстов в обучающей выборке точность классификатора также увеличивается. Стоит отметить, что увеличение количества текстов увеличивает время обучения, но существенно не увеличивает точность классификатора после того, как количество текстов достигнет 1000 (при увеличении количества текстов с 1000 до 5000 текстов время увеличивается примерно в 27 раз, но точность увеличивается на 2%).

Таким образом, можно обучить классификатор на выборке из 1000 текстов для оптимизации времени обучения и качества классификатора.

1.8. Влияние наличия стоп-слов в тексте

На этапе предварительной обработки текста происходит процесс удаления стоп-слов. Как отмечалось ранее, стоп-слова — это текстовые шумы или неинформативные слова, которые встречаются в большом количестве, но не имеют семантического значения и при игнорировании этих слов исходное предложение не теряет своего смысла.

Для проведения данного исследования было проведено сравнение зависимости качества классификатора от наличия процесса удаления стоп-слов на этапе предварительной обработки текстов.

Результаты проведенного исследования приведены в таблице 1.3.

Таблица 1.3 – Зависимость качества классификатора от наличия стоп-слов

Количество текстов	*Точность на тестовой выборке (Без)*	*Точность на тестовой выборке (С)*	*F1-мера (Без)*	*F1-мера (С)*
50	0.2	0.2	0.27	0.27
100	0.9	0.95	0.89	0.94
200	0.9	0.90	0.90	0.90
500	0.92	0.93	0.92	0.93
1000	0.92	0.91	0.92	0.91
5000	0.943	0.933	0.94	0.93

Ниже, на рисунках 1.8 и 1.9, представлены данные из таблицы 1.3 в виде графика.

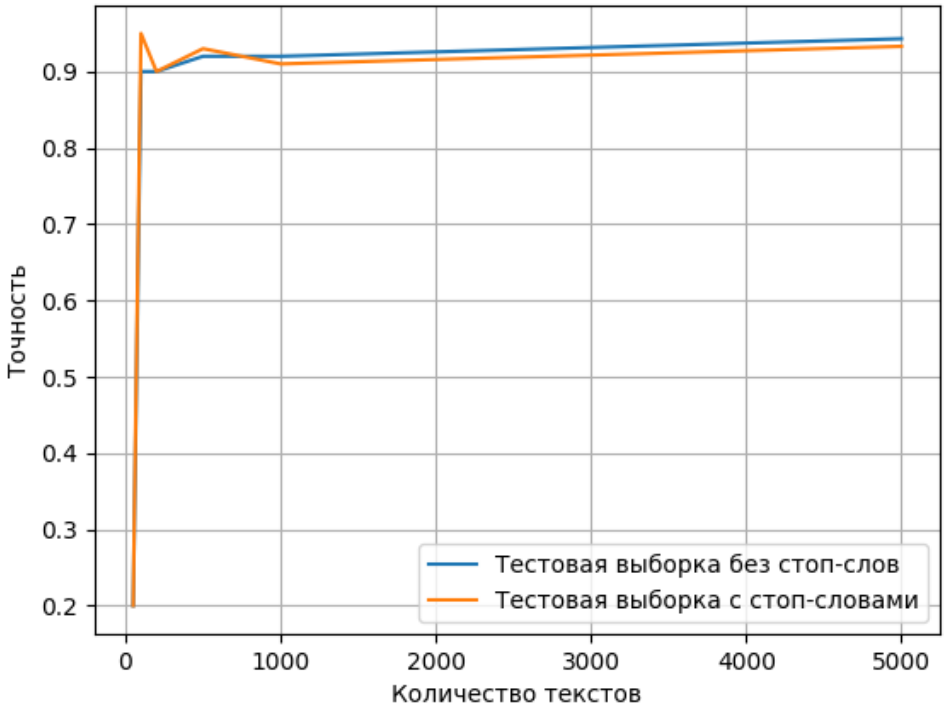

Рис. 1.8 – График зависимости точности классификатора от наличия стоп-слов

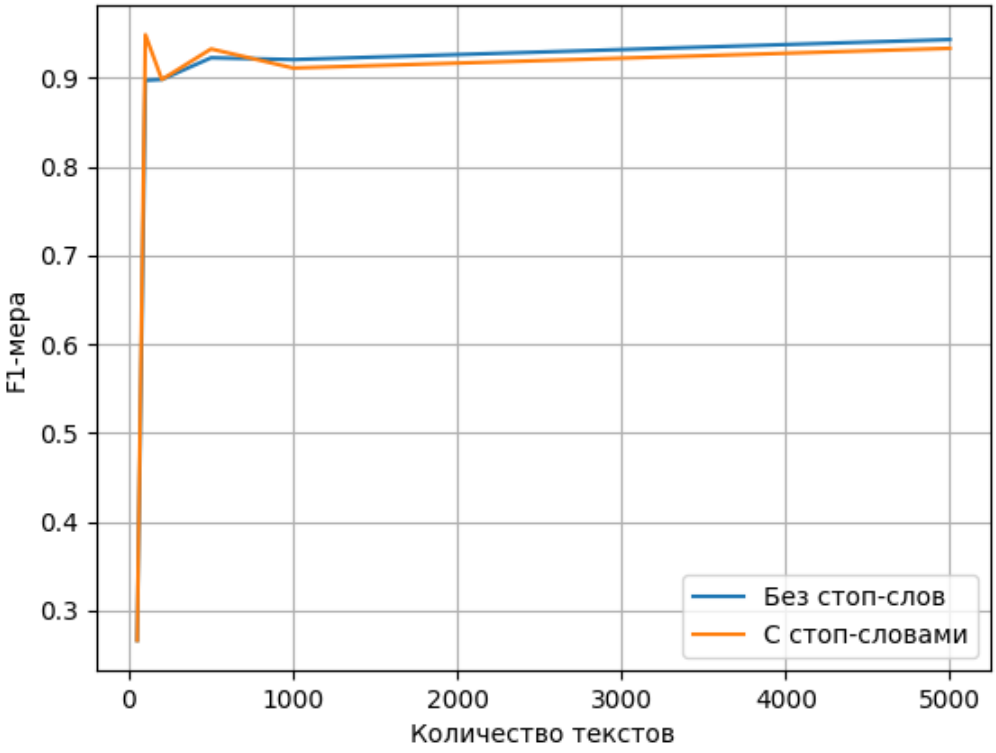

Рис. 1.9 – График зависимости F1-меры от наличия стоп-слов

На основании полученных результатов можно сделать вывод, что наличие стоп-слов в тексте обучающей выборки не сильно влияет на качество классификатора. В обучающих выборках с небольшим количеством текста (с 50 до 500 текстов) с стоп-словами обученные на них классификаторы имеют несколько более высокую точность, чем классификаторы, обученныы на обучающих выборках без стоп-слов (на 1-2%). Но в обучающих выборках с большим количеством текстов (1000 или 5000 текстов) классификаторы, обученные на обучающих выборках без стоп-слов, имеют более высокую точность (на 1%).

1.9. Влияние ядра метода опорных векторов

Ядро (kernel) в методе опорных векторов определяет функцию, которая вычисляет скалярное произведение двух векторов в пространстве признаков. Оно позволяет проецировать данные в более высокомерное пространство, где они могут быть линейно разделимыми, даже если исходное пространство не является линейно разделимым.

Для проведения этого исследования было проведено сравнение зависимости качества классификатора от ядра метода опорных векторов при использовании следующих ядер: линейное, полиномиальное, Гауссово RBF и сигмоидальное.

Ниже, на рисунках 1.10 и 1.11, представлены результаты проведенного исследования

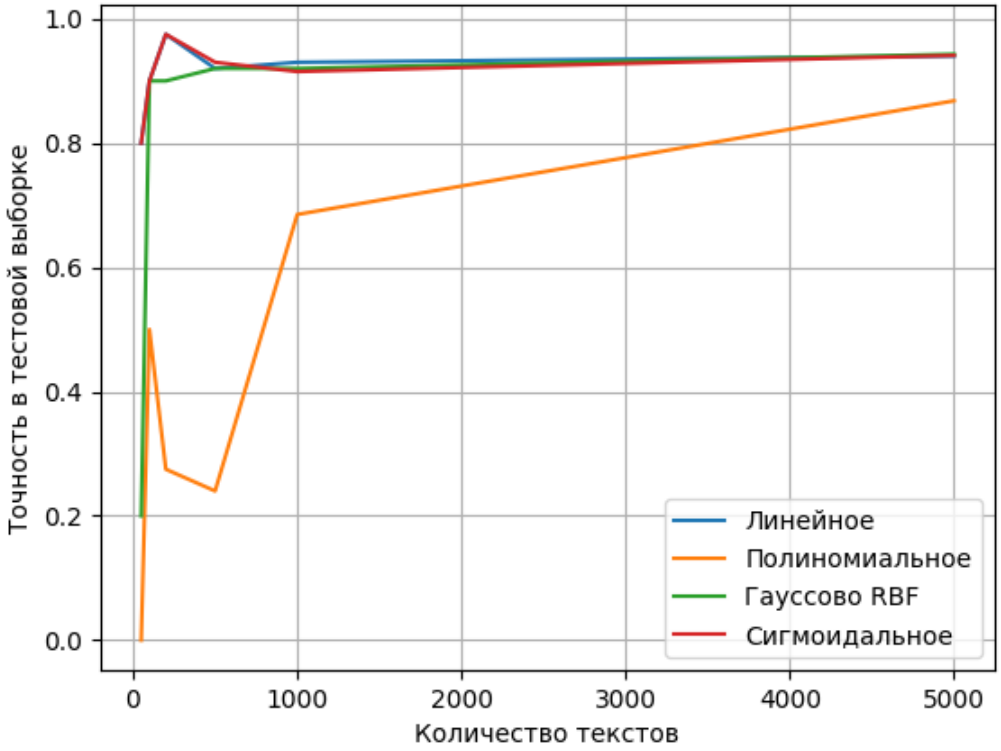

Рис. 1.10 – График зависимости точности классификатора от ядер

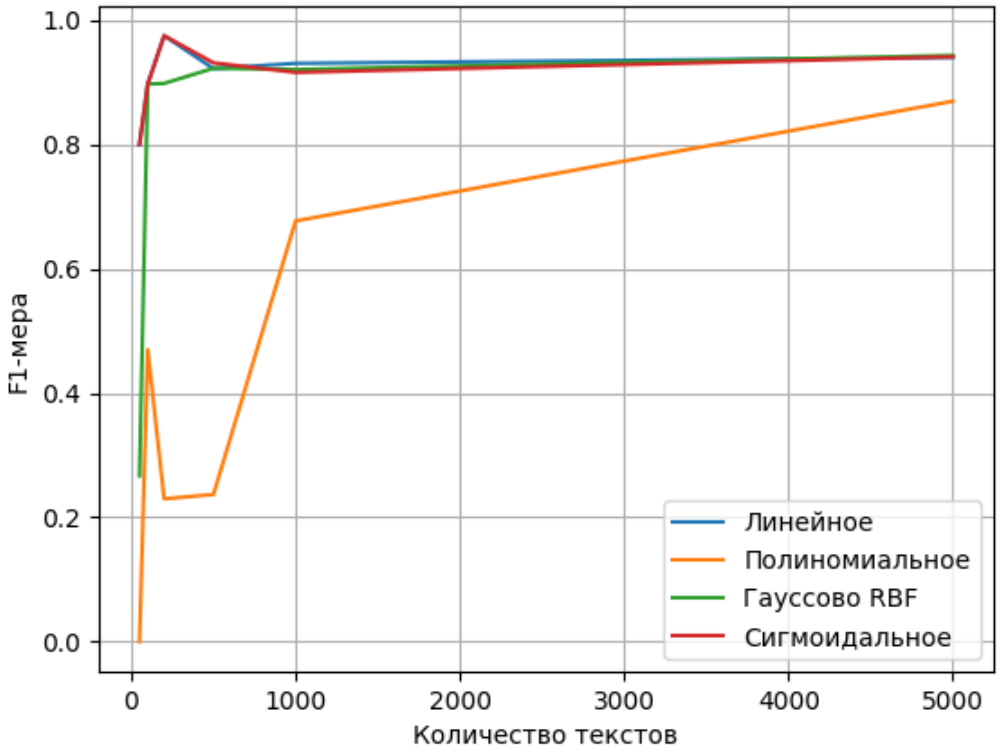

Рис. 1.11 – График зависимости F1-меры от ядер

Из полученных результатов можно сделать вывод, что зависимости точности и F1-меры классификатора от различных ядер совпадают. Классификатор с полиномиальным ядром имеет наименьшее качество, а классификаторы с другими ядрами имеет примерно одинаковую точность.

1.10. Вывод

В этом разделе были приведены выбор языка программирования и средства программирования и рассмотрены необходимые библиотеки, которые используются для разработки программного обеспечения. Также был описан формат входных и выходных данных. Также были приведены описание пользовательского интерфейса и руководство пользователя для установки и использования программного обеспечения. Были описаны технические характеристики устройства для проведения исследований и приведены исследования разработанного метода.

ЗАКЛЮЧЕНИЕ

В ходе преддипломной практики было разработано программное обеспечение, демонстрирующее практическую осуществимость спроектированного в ходе выполнения выпускной квалификационной работы метода классификации новостных текстов по тематике с помощью метода опорных векторов.

Были выполнены следующие задачи:

— выбрана средства программной реализации спроектированного в ходе выполнения выпускной квалификационной работы метода;

— реализовано программное обеспечение метода;

— проведено исследование характеристик разработанного программного обеспечения.

Таким образом, все поставленные задачи были выполнены, а цель достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Welcome to Python.org [Электронный ресурс]. — Режим доступа: https://www.python.org/ (дата обращения: 23.04.2024).
2. Visual Studio Code. — Режим доступа: https://code.visualstudio.com/ (дата обращения: 23.04.2024).