**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине

«Прикладные интеллектуальные системы и экспертные системы»

Студент Бубырь Д. А.

Группа М-ИАП-23-1

Руководитель Кургасов В. В.

доцент, канд. пед. наук

Липецк 2023 г.

Цель работы

Получить практические навыки обработки текстовых данных в среде Jupiter Notebook. Научиться проводить предварительную обработку текстовых данных и выявлять параметры обработки, позволяющие добиться наилучшей точности классификации.

Задание кафедры

1. В среде Jupiter Notebook создать новый ноутбук (Notebook).
2. Импортировать необходимые для работы библиотеки и модули.
3. Загрузить обучающую и экзаменационную выборку в соответствие с вариантом.
4. Вывести на экран по одному-два документа каждого класса.
5. Применить стемминг, записав обработанные выборки (тестовую и обучающую) в новые переменные.
6. Провести векторизацию выборки:
   1. Векторизовать обучающую и тестовую выборки простым подсчетом слов (CountVectorizer) и значеним max\_features = 10000.
   2. Вывести и проанализировать первые 20 наиболее частотных слов всей выборки и каждого класса по-отдельности.
   3. Применить процедуру отсечения стоп-слов и повторить пункт b.
   4. Провести пункты a – c для обучающей и тестовой выборки, для которой проведена процедура стемминга.
   5. Векторизовать выборки с помощью TfidfTransformer (с использованием TF и TF-IDF взвешиваний) и повторить пункты b-d.
7. По результатам пункта 6 заполнить таблицы наиболее частотными терминами обучающей выборки и каждого класса по отдельности. Всего должно получиться по 4 таблицы для выборки, к которой применялась операция стемминга и 4 таблицы для выборки, к которой операция стемминга не применялась.
8. Используя конвейер (Pipeline) реализовать модель Наивного Байесовского классификатора и выявить на основе показателей качества (значения полноты, точности, f1-меры и аккуратности), какая предварительная обработка данных обеспечит наилучшие результаты классификации. Должны быть исследованы следующие характеристики.
9. По каждому пункту работы занести в отчет программный код и результат вывода.
10. По результатам классификации занести в отчет выводы о наиболее подходящей предварительной обработке данных (наличие стемминга, взвешивание терминов, стоп-слова, количество информативных терминов).

Ход работы

Вариант 2

Классы 6, 10, 11

На рисунке 1 представлены необходимые библиотеки и модули для работы программы.

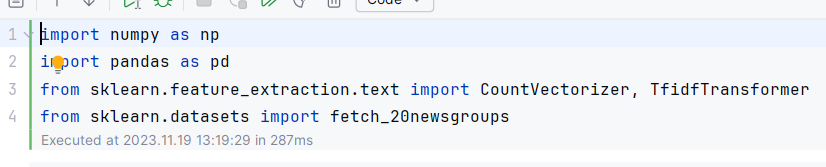


Рисунок 1 — Импортируемые библиотеки и модули

Загрузим обучающую и экзаменационную выборку в соответствии с вариантом. Код для загрузки данных представлен на рисунке 2.



Рисунок 2 — Функция для получения данных

После успешной загрузки данных посмотрим на записи. Для этого выведем по одному документу каждого класса из обучающей выборки. Результаты представлены на рисунке 3.

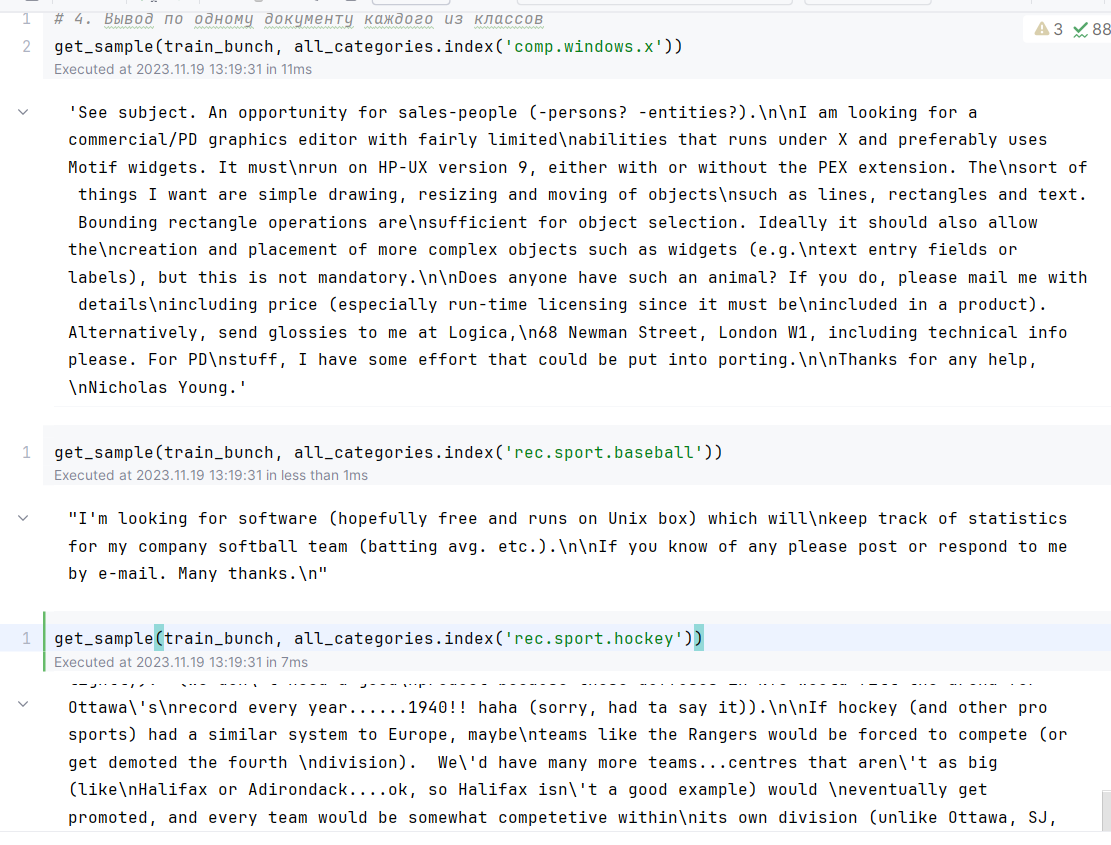


Рисунок 3 — Демонстрация текста категорий, заданных по варианту

Применим стемминг к исходным данным и посмотрим на обработанные данные, которые представлены на рисунке 4.

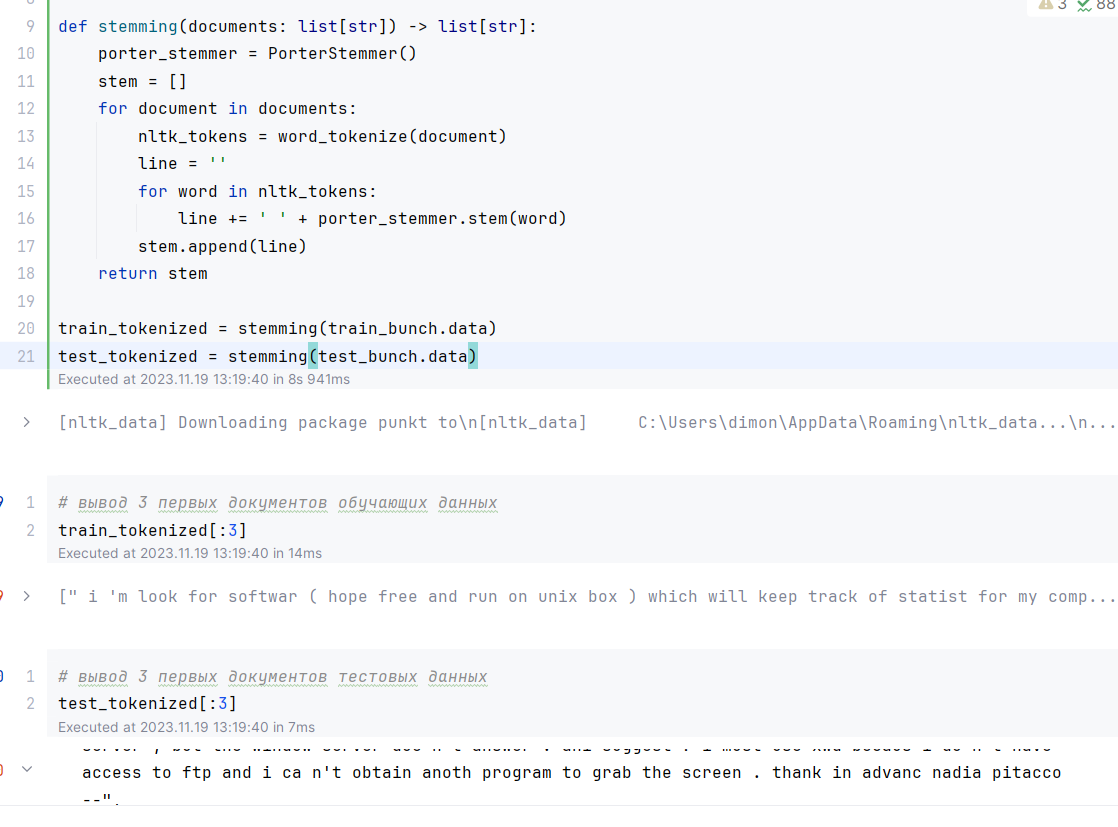


Рисунок 4 — Данные, обработанные стеммингом

Проведем векторизацию выборки. Для этого векторизуем обучающую и тестовую выборку простым подсчетом слов с использованием класса CountVectorizer и значением max\_features = 10000, код для выполнения данного способа представлен на рисунке 5. Выведем первые 20 наиболее частотных слов по всей выборки и отобразим на рисунке 6.

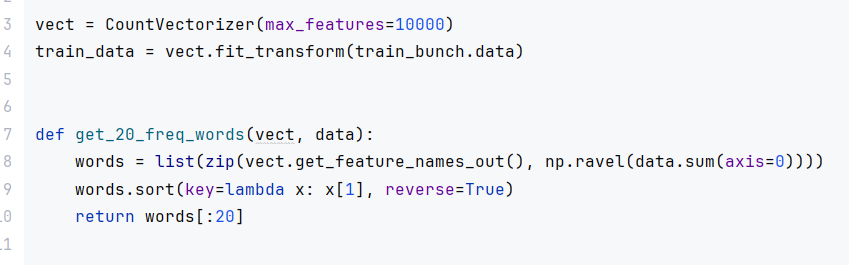


Рисунок 5 – Код для векторизации обучающей выборки простым подсчетом слов

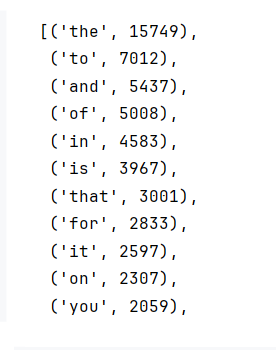


Рисунок 6 – Результат векторизации тестовой выборки простым подсчетом слов

Применим процедуру отсечения стоп-слов и повторим вывод полученных результатов. Код для обработки данных путем отсечения стоп-слов представлен на рисунке 7. Результат векторизации обучающей и тестовой выборки простым подсчетом слов с отсечением стоп-слов представлен на рисунке 8.



Рисунок 7 - Код для векторизации обучающей выборки простым подсчетом слов с отсечением стоп-слов

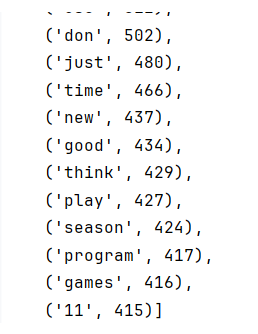


Рисунок 8 – Результат векторизации тестовой выборки простым подсчетом слов с отсечением стоп-слов

Также проведем аналогичный анализ для данных после стемминга. Результат векторизации обучающей и тестовой выборки после стемминга простым подсчетом слов без отсечения стоп-слов представлен на рисунке 9. Результат векторизации обучающей и тестовой выборки после стемминга простым подсчетом слов с отсечением стоп-слов представлен на рисунке 10.



Рисунок 9 - Результат векторизации обучающей и тестовой выборки после стемминга простым подсчетом слов без отсечения стоп-слов

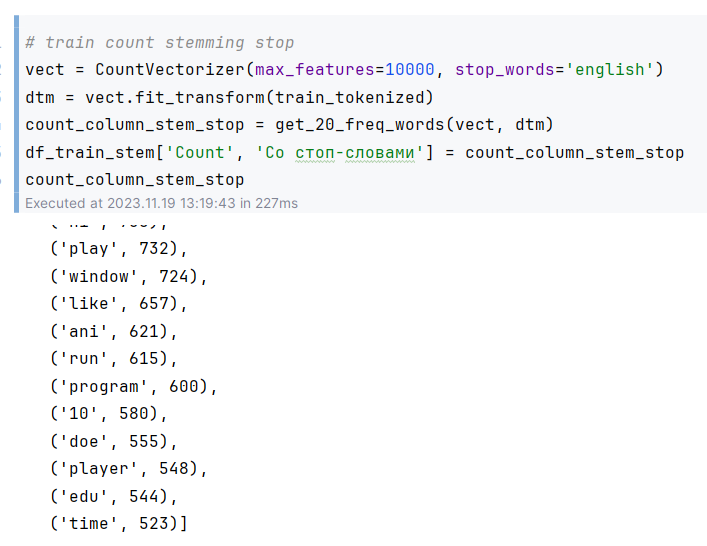


Рисунок 10 - Результат векторизации обучающей и тестовой выборки после стемминга простым подсчетом слов с отсечением стоп-слов

Воспользуемся векторизацией выборки с помощью TfidfTransformer (с использованием TF и TF-IDF взвешиваний). Векторизация выборки с использованием TfidfTransformer для набора данных без использования стоп-слов представлен на рисунке 11, с использованием стоп-слов представлен на рисунке 12.

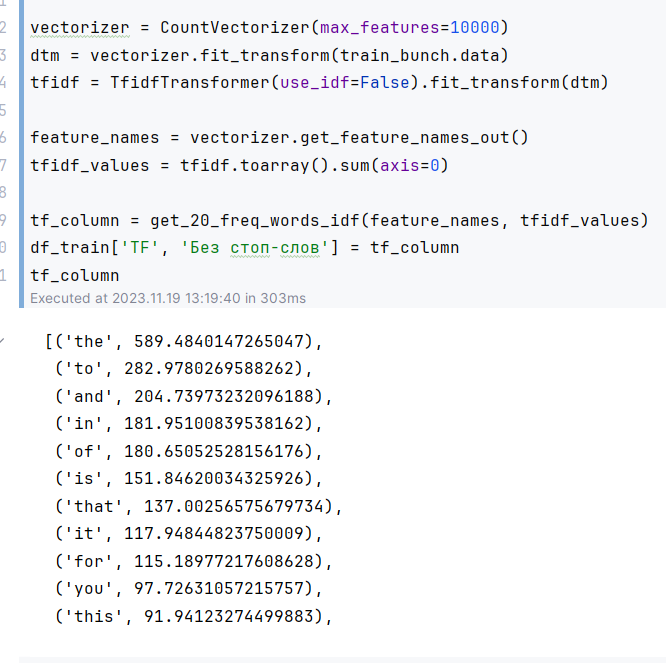


Рисунок 11 – Результат векторизации набора данных без использования стоп- слов

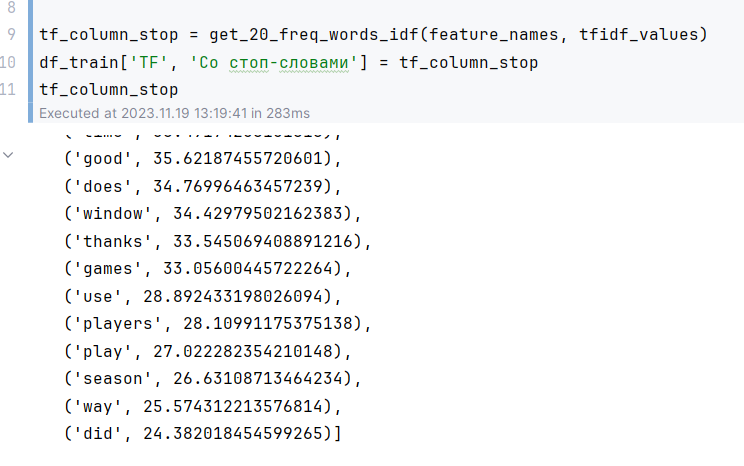


Рисунок 12 – Результат векторизации набора данных с использованием стоп- слов

Проведем аналогичную векторизацию для набора данных после стемминга. Результат векторизации набора данных после стемминга без использования стоп-слов представлен на рисунке 13, с использованием стопслов представлен на рисунке 14.

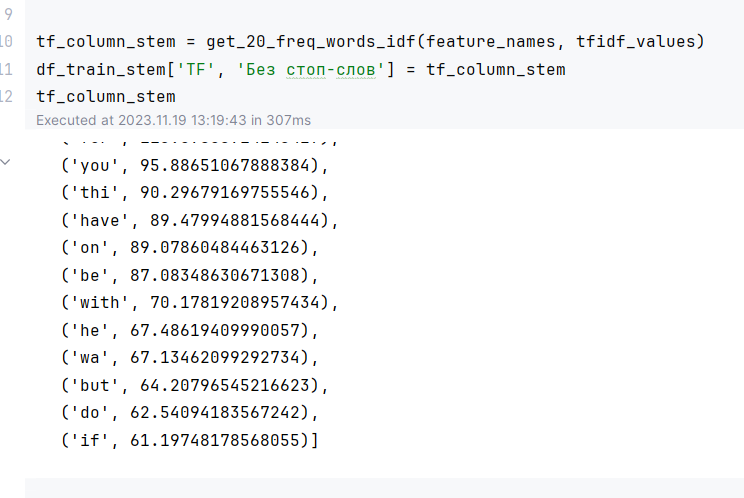


Рисунок 13 – Результат векторизации набора данных после стемминга без использования стоп-слов



Рисунок 14 – Результат векторизации набора данных после стемминга с использованием стоп-слов

Составим сводную таблицу для отображения результатов векторизации и сохраним её в файл Excel. Составленная таблица для обучающего набора данных без применения стемминга представлена на рисунке 15. Для тестового набора данных без применения стемминга представлена на рисунке 16. Для обучающего набора данных с применением стемминга представлен на рисунке 17. Для тестового набора данных с применением стемминга представлен на рисунке 18.

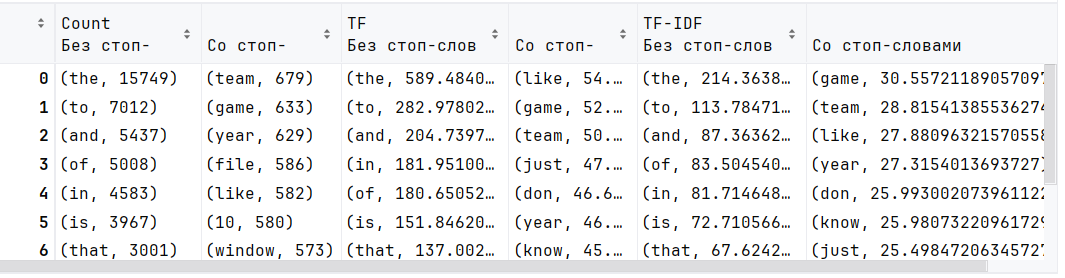


Рисунок 15 – Таблица результата векторизации для обучающего набора данных без применения стемминга

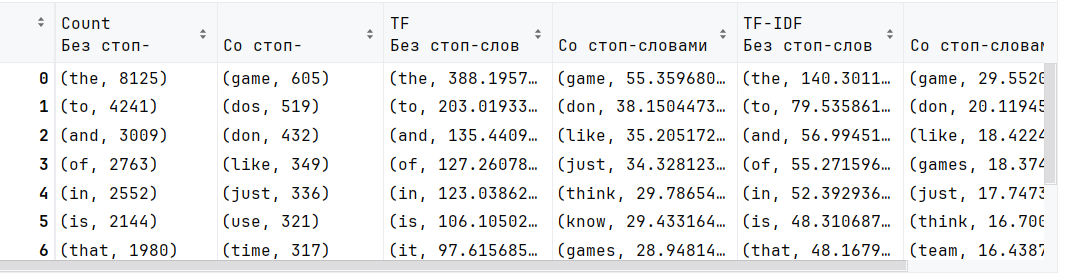


Рисунок 16 – Таблица результата векторизации для тестового набора данных без применения стемминга

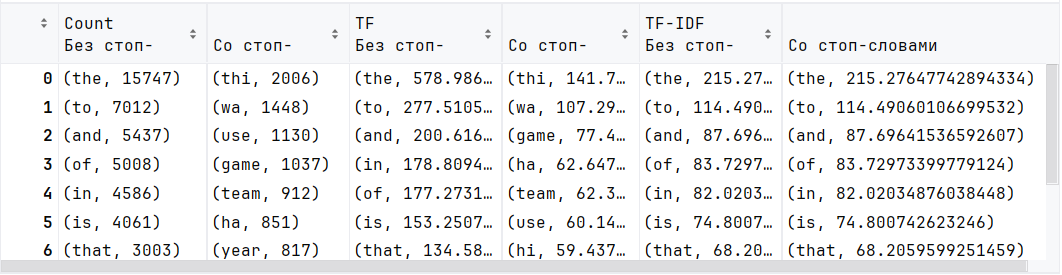


Рисунок 17 – Таблица результата векторизации для обучающего набора данных с применением стемминга

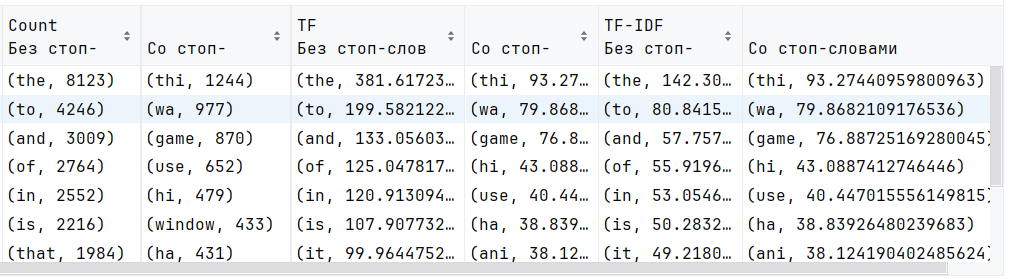


Рисунок 18 – Таблица результата векторизации для тестового набора данных с применением стемминга

Используя конвейер (Pipeline) реализуем модель наивного байесовского классификатора и выявим на основе показателей качества (значение полноты, точности, f1-меры и аккуратности), какая предварительная обработка данных обеспечит наилучшие результаты классификации. Полученный результат оптимальных параметров поиска представлен на рисунке 19.

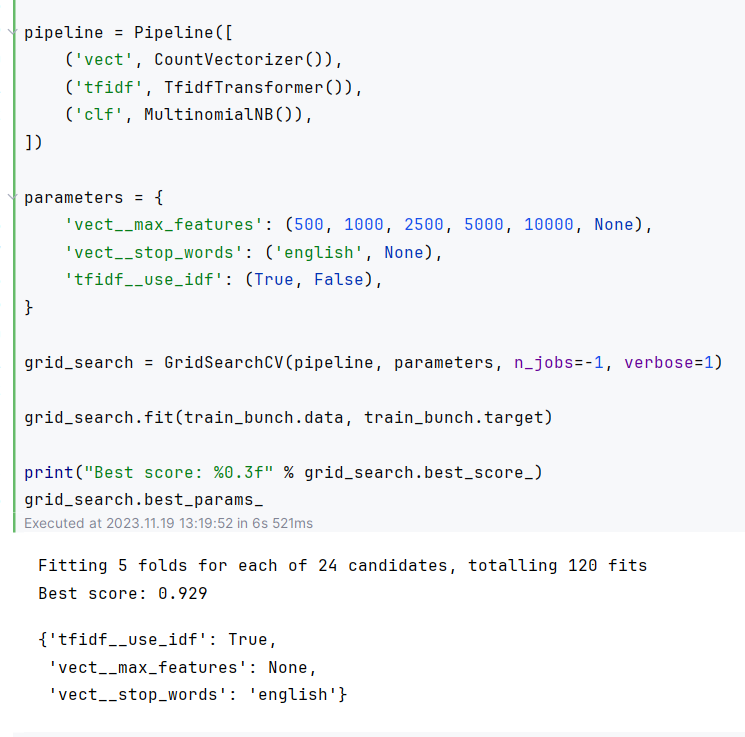


Рисунок 19 – Результат классификации после нахождения оптимальных параметров через конвейер

Вывод

В ходе выполнения данной лабораторной работы я приобрел навыки предварительной обработки текстовых данных. В практической части исследования были использованы различные методы подсчета слов, включая как использование стемминга, так и без него. Кроме того, был применен метод векторизации с использованием TfidfTransformer с разными способами взвешивания. С использованием конвейера и сетки решений были найдены оптимальные наборы параметров для классификации, метрика которых базируется на оценках качества. В результате исследования, наиболее лучшим способом предварительной обработки данных является векторизация TfidfTransformer с использованием TF-IDF взвешиваний и количество информативных терминов = None, с использованием стоп-слов.