Министерство науки и высшего образования Российской Федерации ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «Национальный исследовательский университет ИТМО» (Университет ИТМО)

Факультет Инфокоммуникационных технологий (ИКТ)

Образовательная программа Интеллектуальные системы в гуманитарной сфере

О Т Ч Е Т

по учебной практике

Тема задания: Анализ текстов песен группы "Король и шут"

Обучающийся: Козлов Всеволод Денисович, группа К33421

Руководитель практики от университета: Валитова Юлия Олеговна

Практика пройдена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург 2024

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc160033954)

[1.1 Сбор информации об альбомах и текстах песен группы 4](#_Toc160033955)

[1.2 Сбор информации об эмоциональной окраске слов 5](#_Toc160033956)

[2 Обработка данных 6](#_Toc160033957)

[2.1 Предобработка 6](#_Toc160033958)

[2.2 Разметка частей речи 7](#_Toc160033959)

[2.3 Векторизация 7](#_Toc160033960)

[2.4 Результат 7](#_Toc160033961)

[3 Анализ 9](#_Toc160033962)

[3.1 Соотнесение слов из датасетов со словами из текстов песен 9](#_Toc160033963)

[3.2 Пробная визуализация облака слов 9](#_Toc160033964)

[3.3 Применение библиотеки “dostoevsky” [7] для сентимент-анализа 10](#_Toc160033965)

[3.4 Написание алгоритма для поиска песен по целевому запросу 11](#_Toc160033966)

[4 Написание бэкенд части приложения 12](#_Toc160033967)

[4.1 Точки входа приложения 12](#_Toc160033968)

[5 Написание фронтенд части приложения 14](#_Toc160033969)

[5.1 Используемые инструменты 14](#_Toc160033970)

[5.2 Демонстрация результатов работы 14](#_Toc160033971)

[ЗАКЛЮЧЕНИЕ 20](#_Toc160033972)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 21](#_Toc160033973)

# ВВЕДЕНИЕ

Сложность анализа текстовой информации заключается в том, что нет универсального алгоритма преобразования текстовой информации в данные, над которыми можно производить вычисления. Алгоритм преобразования во многом зависит от специфики данных и анализа, который будет проводиться над данным в дальнейшем.

Когда же мы преобразовали текст в данные, с которыми можем работать, это может быть таблица частотности, TF-IDF или же любой другой метод векторизации текст, возникает трудность с интерпретацией полученных векторов. Векторы имеют слишком большую размерность для визуализации и представления в виде понятным человеку.

После интерпретации полученных векторов при помощи алгоритмов необходимо получить визуальное представление результатов, чтобы можно было сделать выводы на основе имеющихся данных. Этим я хочу сказать, что недостаточно просто представить набор метрик. Метрики необходимо грамотно представить. Здесь мы уже переходим в область визуализации, которая находится на стыке вычислений и дизайна

Целью проекта я поставил пройти через все этапы обработки текстовых данных, собранных из текстов песен группы «Король и Шут». Результатом проекта будет веб-сервис с интерактивными визуализациями. Интерактивность визуализации проявляется в возможности выбора промежутка годов и конкретных альбомов, по которым будут строиться графики

# 1 Сбор данных

Для задачи мне необходимы все тексты песен группы «Король и Шут», разбитые по альбомам. Также для сентимент-анализа мне необходим наборы данных, в которых слову будет соотноситься его эмоциональная окраска. Насколько позитивно слово? Насколько негативно слово?

## 1.1 Сбор информации об альбомах и текстах песен группы

Так как мне не удалось найти открытый набор данных с этой информацией. Я решил написать парсер. В качестве сайта для парсинга я выбрал сайт korol-i-shut.su [1]. Результаты работы парсера я положил в csv файлы. У меня получился файл с альбомами(таблица 1.1) и текстами песен(таблица 1.2)

Таблица 1.1 – фрагмент таблицы с информацией об альбомах

|  |  |  |
| --- | --- | --- |
| title | year | album\_id |
| Камнем по Голове | 1996 | 0 |
| Король и Шут | 1996 | 1 |
| Акустический альбом | 1999 | 2 |

Таблица 1.2 – фрагмент таблицы с информацией об альбомах

|  |  |  |
| --- | --- | --- |
| title | lyrics | album\_id |
| Смельчак и Ветер | Припев:  Я ведь не из робких,  Все мне по плечу.  … | 0 |
| Проказник Скоморох | На свадьбе скоморох,  Был прытким как горох.  Он бегал по столам,  Кидался пудингом в гостей… | 0 |
| Верная Жена | Дождливой ночью парень, выбравшись из леса  Вдруг одинокую избушку увидал… | 0 |

## 1.2 Сбор информации об эмоциональной окраске слов

С наборами данных по эмоциональной окраске слов задача обстояла лучше. Мне удалось найти 3 набора данных по этой теме:

1. Rusentilex [2]
2. Linis 2015 [3]
3. Linis 2016 [3]

Датасеты имели различную разметку. Пришлось обработать их, чтобы привести к единому формату. В итоге у меня получился набор начальных форм слов с оценками в диапазоне от -2 до 2. Где -2 означает максимальную негативную коннотацию. 2 Означает максимальную позитивную коннотацию. 0 – нейтральная коннотация

# 2 Обработка данных

Теперь мы имеем все необходимые данные для начала работы. Превратим данные в векторизованное представление для дальнейшей работы

### 2.1 Предобработка

В предобработке я выделил 3 основных этапа:

1. Разбиение на токены
2. Удаление служебных символов и пунктуации
3. Фильтрация по стоп-словам
4. Лемматизация

Этапы стандартны для обработки текстов. Единственным отличием

является то, что я выделил стоп-слова специфические для набора данных. Например, в них вошло слово “припев”. Также присутствие этапа лемматизации обусловлено тем, что в дальнейшем мне необходимо было минимизировать количество уникальных слов в тексте. Это поможет при дальнейшей обработке. На рисунке 2.1 приведена функция, которая выполняет обработку.



Рисунок 2.1 –Функция для обработки текстовой информации

## 2.2 Разметка частей речи

После обработки все слова были поделены на 3 категории: существительные, глаголы и прилагательные. Для разметки по частям речи использовалась библиотека PyMorphy2 [4]

## 2.3 Векторизация

Для векторизации был выбран алгоритм TF-IDF [5]. Этот алгоритм был выбран так как большинство алгоритмов сентимент-анализа показывают на нем метрики лучше по сравнению с “мешком слов” [6].

Также был написан простой алгоритм для подсчета слов в списке документов. Это пригодится нам в этапе анализа

## 2.4 Результат

В результаты мы получаем 2 файла: “songs\_tagged.pkl” и “corpus.npz”. В первом хранится хранятся токены, разбитые на части речи, фрагмент можно увидеть в таблице 2.1 Во втором векторы tf-idf для каждого текста. Фрагмент можно увидеть в таблице 2.2.

Таблица 2.1 – фрагмент содержания файла “songs\_tagged.pkl”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| title | album\_id | tokens | adjectives | verbs | nouns |
| смельчак и ветер | 0 | [робкий, плечо, сильный, ловкий, ветер, проучить… | [робкий, сильный, ловкий, сильный, поздний, сумасшедший, храбрый, спокойный] | [проучить, дуть, рвать, спасть, разойтись… | [плечо, ветер, ветер, крыша, час, округ… |
| проказник скоморох | 0 | [свадьба, скоморох, прыткий, горох, бегать, стол, кидаться… | [прыткий, старый, весёлый, хмельный, женский.. | "[бегать, кидаться, нести, ржать, смеяться, плясать… | [свадьба, скоморох, горох, стол, пудинг… |
| верная жена | 0 | [дождливый, ночью, парень, выбраться, лес, пустить… | [дождливый, одинокий, голодный, дряхлый, покойный… | [выбраться, увидать, надеяться, найтись, устать… | [парень, лес, избушка, утро, место… |

Таблица 2.2 – фрагмент содержания файла “corpus.pkl”

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| 0 | 0.5023 | 0.2300 | 0.4123 | 0.0239 | 0.0013 | 0.0123 | 0.00 |
| 1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

# 3 Анализ

## 3.1 Соотнесение слов из датасетов со словами из текстов песен

Возьмем уникальные слова из всех песен, приведенные к начальной форме, и уникальные слова из наборов данных с эмоциональной окраской слов. Объединим их в единый набор данных, где каждому слову, встречавшемуся хотя бы раз в одной из песен, будет соответствовать число от -2 до 2, отражающее его эмоциональную окраску. Для всех слов, которые присутствуют в песнях, но отсутствуют в наборах данных с эмоциональной окраской слов, поставим 0. Фрагмент результата можно увидеть в таблице 3.1

Таблица 3.1 – фрагмент таблицы с эмоциональной оценкой слов

|  |  |
| --- | --- |
| word | rating |
| неподвижный | 0.0 |
| былина | 0.0 |
| безголовый | -0.6666666666 |
| закрыть | 0.0 |
| близко | -0.3333333333 |
| ладный | 0.3333333333 |

## 3.2 Пробная визуализация облака слов

Теперь мы имеем все, чтобы преобразовать данные в облако слов, где размер слова обозначает частоту употребления, а цвет – эмоциональную окраску слова. Результат представлен на рисунке 3.1.



Рисунок 3.1 – Пробное облако слов

## 3.3 Применение библиотеки “dostoevsky” [7] для сентимент-анализа

Если до этого мы работали с отдельным словами без контекста, то теперь мы посмотрим на текст песни целиком. И на основе текста определим к какой категории песня больше относится: негативная, позитивная, нейтральная. Результат положим в csv файл. Фрагмент его содержание можно посмотреть в таблице 3.2.

Таблица 3.2 – Фрагмент таблицы с эмоциональной оценкой текстов песен

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| title | album\_id | positive | negative | neutral |
| смельчак и ветер | 0 | 0.1688 | 0.1824 | 0.1561 |
| проказник скоморох | 0 | 0.15204 | 0.2337 | 0.1225 |
| верная жена | 0 | 0.1097 | 0.3007 | 0.1480 |

## 3.4 Написание алгоритма для поиска песен по целевому запросу

Также в веб-сервисе я хочу реализовать поиск по текстам песен. Для этого была решена использоваться косинусная мера между векторами, полученными при помощи TF-IDF [8]. Это позволяет нам оценить соноправленность вектора текста песни и вектора запроса. В результате получаем число от 0 до 1. В таблице 3.3 приведены примеры результата для запроса “Песня о 2-х друзьях, на которых напали разбойники”

Таблица 3.3 – Лучшие совпадения по косинусной мере

|  |  |
| --- | --- |
| Название песни | Совпадение |
| Два Друга и Разбойники | 0.205547 |
| Песня Мушкетёров | 0.110314 |
| Собрание | 0.106530 |
| В Париж - Домой | 0.099017 |
| Бунтарь | 0.072206 |

# 4 Написание бэкенд части приложения

Для веб-приложения мы напишем API на “Django-rest-framework”[9]. API будет высылать ответы в формате JSON на HTTP запросы.

## 4.1 Точки входа приложения

На основе анализа проведенного в главе 3. Создадим точки входа для веб-приложения:

Точка входа для получения частоты слов для облака слов

URL: word\_cloud/word\_frequency

Принимаемые типы запросов: GET

Параметры GET-запроса:

* tag\_type – Часть речи, возвращаемых слов
* year\_min – Минимальный год текстов песен
* year\_max – Максимальный год текстов песен
* albums\_titles – Список альбомов

Точка входа для получения цвета, соответствующему каждому слову.

URL: word\_cloud/word\_color

Принимаемые типы запросов: GET

Точка входа для получения списка альбомов

URL: album/list

Принимаемые типы запросов: GET

Точка входа для получения данных о сентимент-анализе текстов песен

URL: vizualization/sentiment

Принимаемые типы запросов: GET

Параметры GET-запроса:

* year\_min – Минимальный год текстов песен
* year\_max – Максимальный год текстов песен
* albums\_titles – Список альбомов

Точка входа для поиска песен по целевому запросу

URL: song/search

Принимаемые типы запросов: POST

Параметры POST-запроса:

* query – Целевой запрос

# 5 Написание фронтенд части приложения

Мы имеем все данные, необходимые для создания веб-приложения с интерактивными визуализациями.

## 5.1 Используемые инструменты

В качестве фронтенд фреймворка используется “Vue.js” [10] В качестве UI-фреймворка используется “Vuetify” [11]. Для визуализаций используется plotly [12] и wordcloud2.js [13].

## 5.2 Демонстрация результатов работы

Были сделаны 4 страницы: главная страница; страница с облаком слов; страница с эмоциональным анализом; страница с поиском по песням. Все страницы доступны, либо нажатием соответствующего пункта в выпадающем меню(рисунок 5.1), либо нажатием на заголовок приложения(рисунок 5.2).



Рисунок 5.1 – заголовок приложения

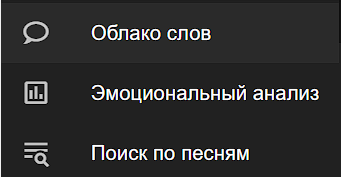


Рисунок 5.2 – Выпадающее меню для навигации

Главная страница служит для ознакомления пользователя с направленностью веб-сервиса. Объясняет, как перемешаться по сайту. Ее содержание можно посмотреть на рисунке 5.3



Рисунок 5.3 – Внешний вид главной страницы

На странице есть 3 основных элемента: меню для выбора периода и альбомов(рисунок 5.4); визуализация, соотносящая размер слова и количество употребления слова в тексте(рисунок 5.5); облака слов для 4-х категорий(рисунок 5.6)

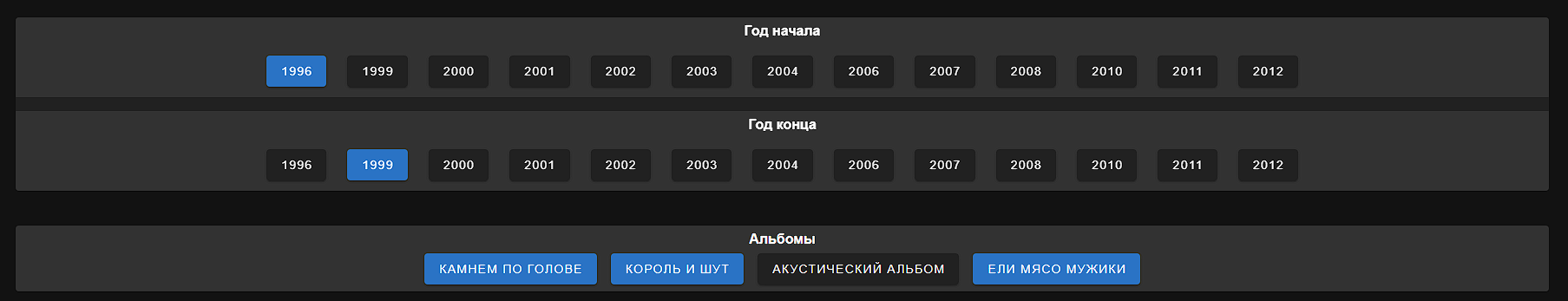


Рисунок 5.4 – Внешний вид меню для выбора периода и альбомов

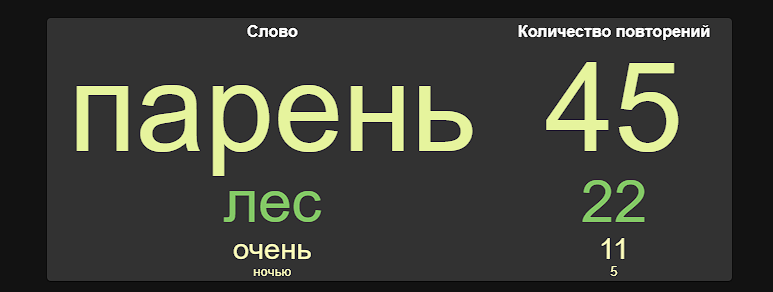


Рисунок 5.5 – отношения размера слова к частоте употребления

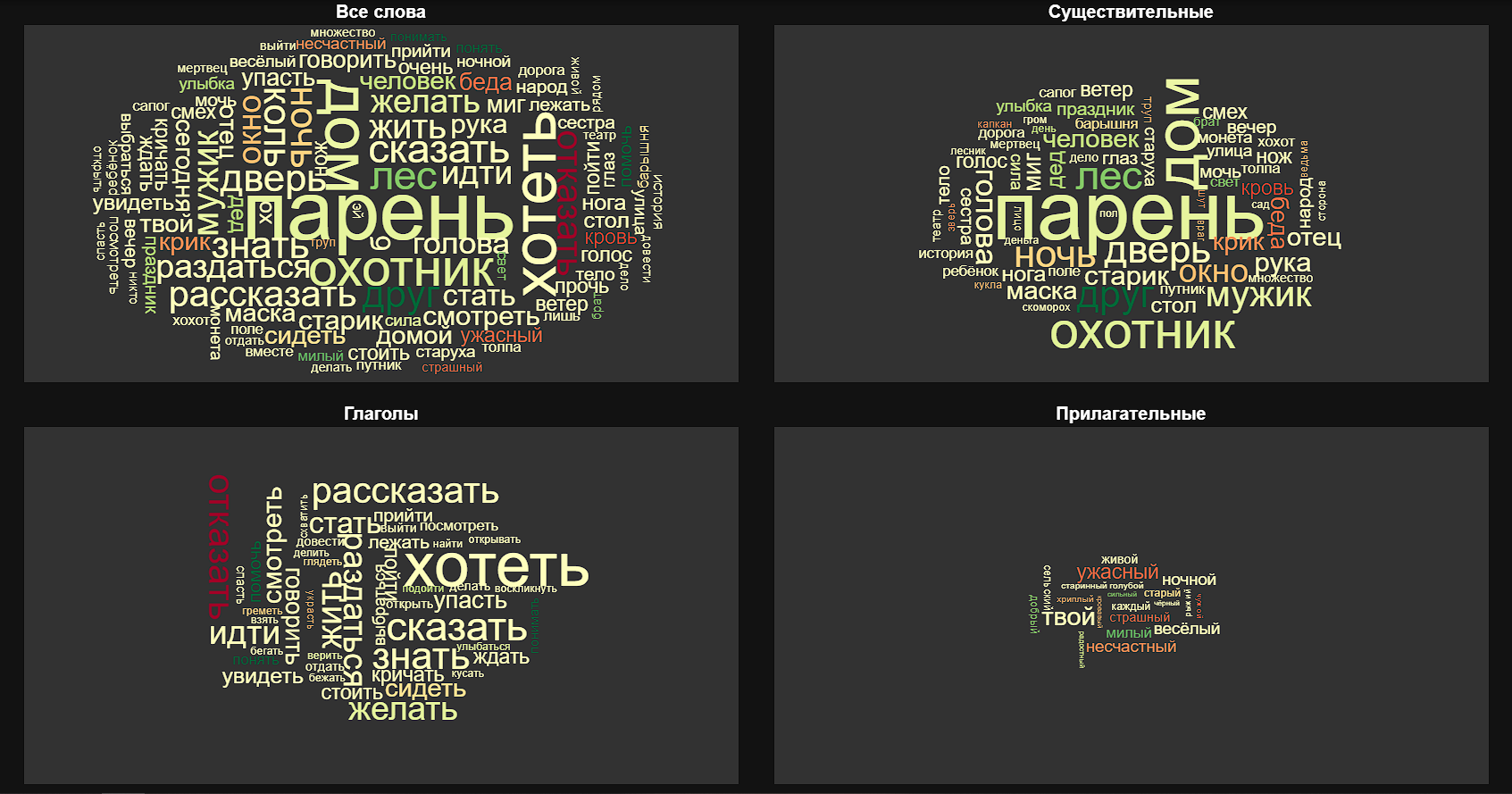


Рисунок 5.6 – Облака слов

На странице 3 основных элемента: меню для выбора периода и альбомов (рисунок 5.7); график распределения песен по преобладающей эмоциональной окраске(рисунок 5.8); график распределения вероятностей отнесения песни к эмоциональной окраске(рисунок 5.9)

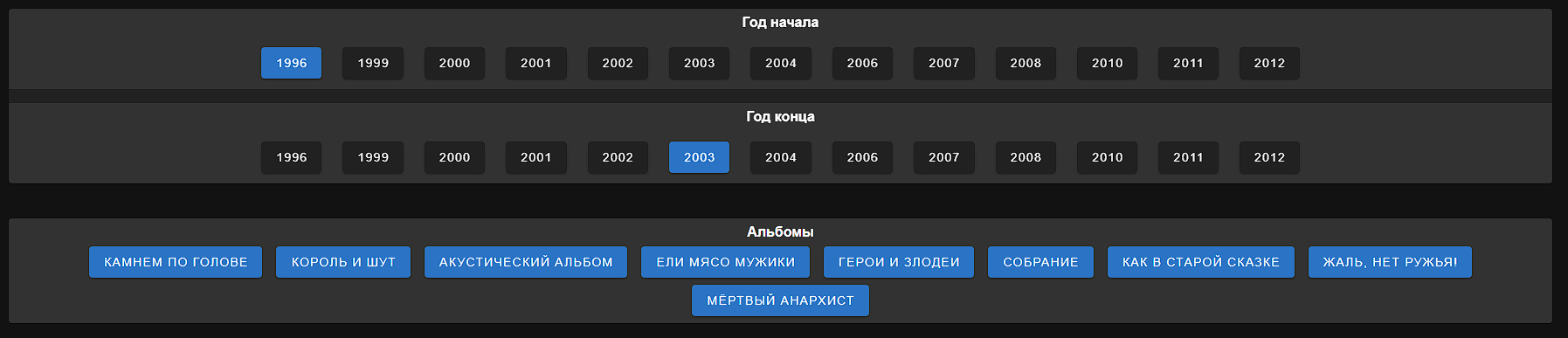


Рисунок 5.7 – Внешний вид меню для выбора периода и альбомов

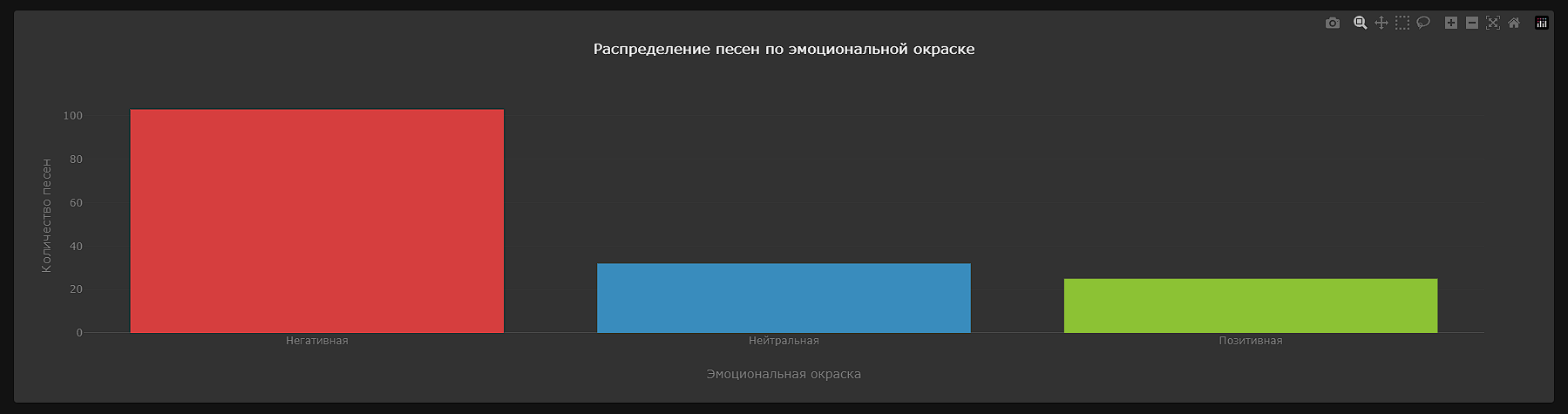


Рисунок 5.8 – Распределения песен по эмоциональной окраске

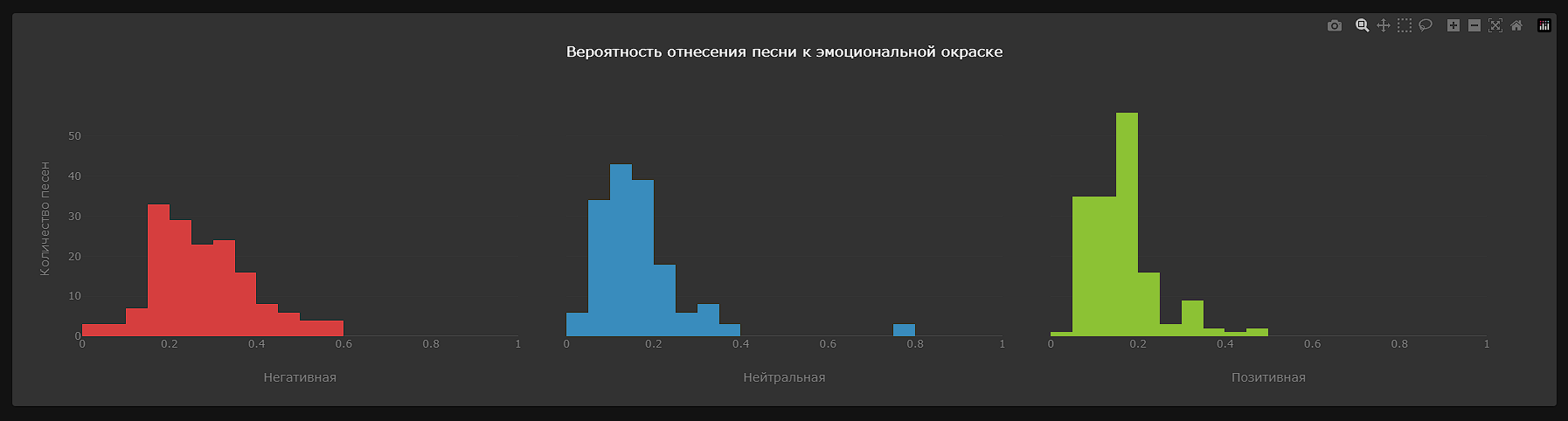


Рисунок 5.9 – График распределения вероятностей

На странице 2 основных элемента: поле для поиска(рисунок 5.10); результаты поиска(рисунок 5.11). При нажатии на наименование результаты, выезжает поле с текстом песни(рисунок 5.12)

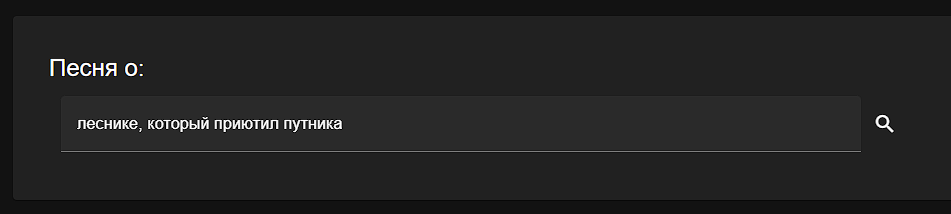


Рисунок 5.10 – Поисковое поле

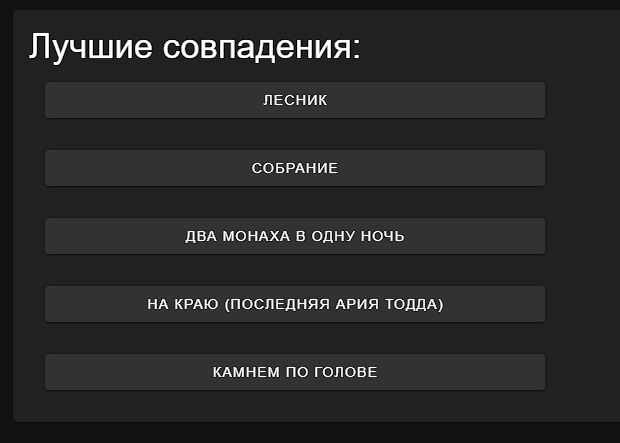


Рисунок 5.11 – Результаты поиска

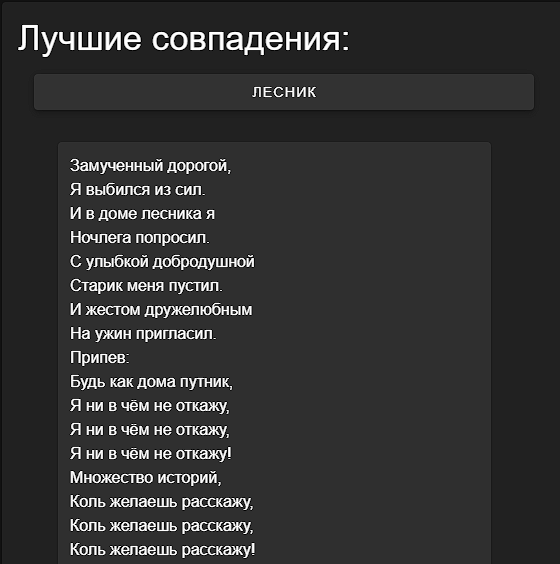


Рисунок 5.12 – Содержание выдачи

# ЗАКЛЮЧЕНИЕ

В ходе проделанной работы мы прошли все этапы анализа текстовой информации от сбора до представления результатов анализа в виде интерактивного веб-сервиса. В результате были достигнуты все цели, поставленные в начале.

Работа над этим проектом расширило мои знания обработки естественного языка и репрезентации результатов анализа. Мало провести грамотную аналитику, необходимо, чтобы остальные могли её понять. Навыки веб-разработки очень помогают с этой задачей. Рад, что у меня получилось их расширить.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Тексты песен группы Король и Шут. — Текст : электронный // Группа Король и Шут : [сайт]. — URL: https://korol-i-shut.su/albums/ (дата обращения: 06.02.2024).
2. Лукашевич Н.В., Левчик А.В. Создание лексикона оценочных слов русского языка РуСентилекс // Труды конференции OSTIS-2016, С.377-382.
3. Алексеева С.В., Кольцова Е.Ю., Кольцов С.Н. Linis-crowd.org: лексический ресурс для анализа тональности социально-политических текстов на русском языке // Компьютерная лингвистика и вычислительные онтологии: сборник научных статей. Труды XVIII объединенной конференции «Интернет и современное общество» (IMS-2015), Санкт-Петербург, 23 – 25 июня 2015 г., СПб., 2015, С. 25-32
4. Analysis of Images, Social Networks and Texts / Khachay, Yu,and Mikhail, a. P. Natalia [и др.]. — 1-е изд. — Yekaterinburg : Springer International Publishing, 2015. — 542 c. — Текст : непосредственный.
5. Fatih, Karabiber TF-IDF — Term Frequency-Inverse Document Frequency / Karabiber Fatih. — Текст : электронный // LearnDataSci : [сайт]. — URL: https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/#:~:text=Using%20scikit%2Dlearn-,What%20is%20TF%2DIDF%3F,%2C%20relative%20to%20a%20corpus). (дата обращения: 09.02.2024).
6. Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets / Akuma, S., Lubem [и др.]. — Текст : непосредственный // Int. j. inf. tecnol. — 2022. — № 15. — С. 3629–3635.
7. dostoevsky PyPI. — Текст : электронный // PyPI : [сайт]. — URL: https://pypi.org/project/dostoevsky/ (дата обращения: 09.02.2024).
8. Umadevi Document comparison based on tf-idf metric / Umadevi, M. — Текст : непосредственный // International Research Journal of Engineering and Technology (IRJET). — 2020. — № 2. — С. 1546--1550
9. djangorestframework PyPI. — Текст : электронный // PyPI : [сайт]. — URL: https://pypi.org/project/djangorestframework/ (дата обращения: 10.02.2024)..
10. Vue.js - The Progressive Javascript Framework. — Текст : электронный // vuejs : [сайт]. — URL: https://vuejs.org/ (дата обращения: 14.02.2024).
11. Vueetify - A Vue Component Framework. — Текст : электронный // vuetifyjs : [сайт]. — URL: https://vuetifyjs.com/en/ (дата обращения: 14.02.2024).
12. Plotly JavaScript Open Source Graphing Library. — Текст : электронный // plotly : [сайт]. — URL: https://plotly.com/javascript/ (дата обращения: 14.02.2024).
13. wordcloud-npm. — Текст : электронный // npmjs : [сайт]. — URL: https://www.npmjs.com/package/wordcloud (дата обращения: 14.02.2024).