

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Кафедра 319 «Системы интеллектуального мониторинга»

Курсовая работа

по дисциплине «Технология разработки программного
обеспечения»

**«Проектирование и разработка веб-приложения
классификации новостей с применением методов
машинного обучения»**

Студент _____ Давыдов Ю.П.

Группа _____ М30-222М-19

Руководитель _____ Полицына Е.В.

Оценка _____ Дата защиты «____» _____ 2020 г.

Москва 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Кафедра 319 «Системы интеллектуального мониторинга»

З А Д А Н И Е

на курсовую работу по дисциплине
Технология разработки программного обеспечения

Студент МЗО - 222М - 19 Давыдов Юрий Павлович

(№ группы, Ф. И. О.)

Тема Проектирование и разработка веб-приложения классификации
новостей с применением методов машинного обучения

Перечень вопросов, подлежащих разработке в курсовой работе:

1. Проектирование архитектуры системы и выбор средств разработки
2. Проектирование и реализация ядра системы - классификатора
3. Проектирование и реализация фронтенда
4. Проектирование и реализация бекенда

Рекомендуемая литература

1. Django [Электронный ресурс]. - Режим доступа:
<https://docs.djangoproject.com/>. - Заглавие с экрана. - (Дата обращения
26.12.20).
2. Python [Электронный ресурс]. - Режим доступа: <https://www.python.org/3/>
- Заглавие с экрана. - (Дата обращения 27.12.20).
3. Мюллер. Введение в машинное обучение

Задание выдано «12» сентября 2020 г.

Руководитель Полицына Екатерина Валерьевна, к.т.н., доцент
кафедры 319 МАИ

(Ф. И. О., должность, подпись)

Студент

Давыдов Ю.П.

(подпись)

Содержание.

1. Описание возможностей приложения.	5
1.1 Общие сведения.	5
1.2 Назначение и цель.	5
2. Требования к приложению.	5
2.1 Функциональные требования.	5
2.2 Требования к реализации.	6
3. Архитектура системы.	6
3.1 Схема архитектуры.	6
3.2 Протоколы взаимодействия.	7
3.3 Используемые технологии.	16
3.4 Паттерны проектирования.	16
4. Описание классификатора.	16
4.1 Классы.	16
4.2 Вектор признаков.	16
4.3 Модель машинного обучения классификатора.	17
4.4 Обучающая и тестовая коллекция документов.	17
4.5 Оценка точности классификации.	17
5. Описание инфраструктуры разработки.	18
5.1 Система контроля версий.	18
5.2 Сборка и развертывание приложения.	18
5.3 Обновление модели классификатора.	19
5.4 Лемматизация коллекции текстов.	19
5.5 Загрузка списка стоп слов.	19

6. Результаты тестирования.	19
7. Интерфейс и возможности системы.	20
8. Анализ полученных результатов.....	24
9. Список использованных источников.....	25

1. Описание возможностей приложения.

1.1 Общие сведения.

Веб-приложение классификации новостных статей представляет веб-сервис, который позволяет классифицировать новости и управлять коллекцией собранных новостей с информационного портала «<https://overclockers.ru/>».

1.2 Назначение и цель.

С помощью данного веб-приложения пользователь получает возможность определять, к какой категории относится новостная статья, которую он ввел. Так же данный сервис имеет возможность просмотра собранных новостных статей, которые выполняли роль входных данных для обучения классификатора, включая возможность их фильтрации. Ко всему прочему веб-приложение владеет как возможностью удаления и изменения собранных новостных статей, так и возможностью добавления новостных статей в коллекцию.

2. Требования к приложению.

2.1 Функциональные требования.

Функциональные требования к системе следующие:

- добавление новостной статьи;
- изменение новостной статьи;
- удаление новостной статьи;
- вывод новостных статей по страницам;
- фильтрация новостных статей по категориям;
- поиск по словам в заголовках статей;
- классификация текста новостной статьи с помощью методов машинного обучения;
- обновление модели классификатора.

2.2 Требования к реализации.

Система должна быть реализована на базе клиент-серверной архитектуры в виде веб-сервиса, обладающего возможностью классификации новостных статей и имеющего пользовательским интерфейс.

Элементы веб-приложения должны быть реализованы с помощью следующих технических инструментов:

- back-end должен быть реализован на языке программирования Python с помощью фреймворка Django;
- front-end должен быть реализован с помощью HTML и языка программирования Javascript;
- проект, а также вся необходимая документация вместе с нужными данными должны храниться на сайте сервиса GitHub в системе контроля версий git.

При работе пользователя с данной системой, она должна быть отказоустойчивой.

3. Архитектура системы.

3.1 Схема архитектуры.

На рисунке 1 показана архитектура разработанного приложения.

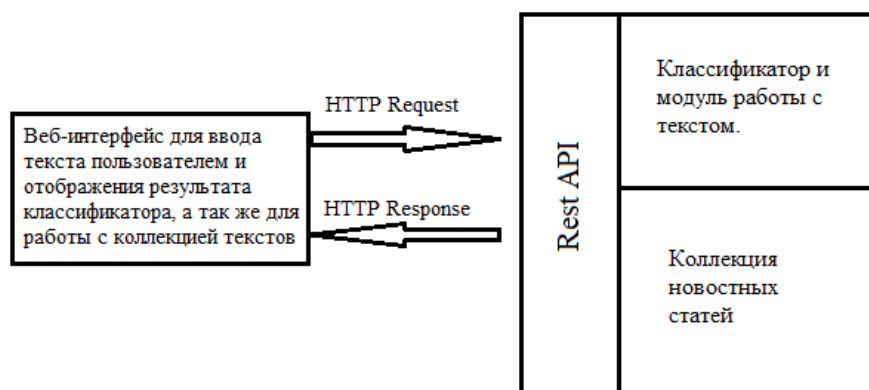


Рисунок 1 – Архитектура приложения.

Front-end часть приложения, представляющая собой веб-интерфейс как для ввода текста новости, так и для работы с коллекцией текстов, которая взаимодействует с Back-end частью приложения посредством принятия и отправки HTTP запросов.

Back-end часть приложения состоит из классификатора, модуля для работы с новостной коллекцией, а также сама коллекция новостных статей, сохраненная в формате xml. RestAPI позволяет получать от фронта HTTP запросы с необходимыми входными данными, обрабатывать входные данные, и отправлять фронту HTTP ответы с полученными выходными данными. Входные и выходные данные представлены в виде JSON.

3.2 Протоколы взаимодействия.

Разработанная система является RESTful веб-сервисом. Используемые в системе API:

- Страница классификатора.

GET /

Возвращает HTML-страницу для классификации текстов.

- Страница фильтрации новостей.

GET /news-filter

Возвращает html страницу для фильтрации новостей по категориям, а также поиска новостей, причем поиск осуществляется по заголовкам новостей.

- Страница добавления новости.

GET /new-add

Возвращает html страницу для добавления новости в коллекцию.

- Страница отображения полной информации о новости.

GET /news/{newId}

Возвращает html страницу для отображения полной информации о новости, с возможностью её редактирования и удаления.

- Вывод коллекции новостей по страницам.

GET /rest-api/news/pages/{page}

Точка входа возвращает html страницу с json, который содержит список новостей коллекции определенной страницы, а также количество новостей в коллекции. Json-схема выходного json представлена в листинге 1.

Листинг 1 - Json-схема выходного json для вывода коллекции новостей по страницам.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "listNews",
  "description": "information about news on page",

  "properties": {
    "countNews": {
      "type": "integer"
    },
    "newsPreviewInformation": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "id": {
            "type": "integer"
          },
          "title": {
            "type": "string"
          },
          "category": {
            "type": "string"
          }
        }
      }
    }
  }
}
```



```

"properties":{
    "countNews":{
        "type": "integer"
    },
    "newsPreviewInformation":{
        "type": "array",
        "items":{
            "type": "object",
            "properties": {
                "id":{
                    "type": "integer"
                },
                "title":{
                    "type": "string"
                },
                "category":{
                    "type": "string"
                },
                "author":{
                    "type": "string"
                }
            }
        }
    }
}

```

```
}
```

- Добавление новости.

POST /rest-api/news/add-new

Точка входа принимает в теле запроса json, который содержит информацию о новой новости. Возвращает html страницу со статусом 200 при успешном добавлении новости. Json-схема входного json представлена в листинге 4.

Листинг 4 - Json-схема входного json для добавления новости.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "title": "newForAdd",
  "description": "information about new for add",

  "properties":{
    "link":{
      "type": "string"
    },
    "category":{
      "type": "string"
    },
    "date":{
      "type": "string"
    },
    "title":{
      "type": "string"
    },
    "author":{
      "type": "string"
    },
    "text":{
```

```

        "type": "string"
    }
}
}

```

•Удаление новости.

DELETE /rest-api/news/{newId}

Точка входа удаляет выбранную новость. Возвращает html страницу со статусом 200 при успешном удалении новости.

•Редактирование новости.

PUT /rest-api/news/{newId}

Точка входа принимает в теле запроса json, который содержит информацию об изменениях новости. Возвращает html страницу со статусом 200 при успешном изменении новости. Json-схема входного json представлена в листинге 5.

Листинг 5 - Json-схема входного json для редактирования новости.

```

{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "title": "newForUpdate",
    "description": "information about new for update",

    "properties":{
        "link":{
            "type": "string"
        },
        "category":{
            "type": "string"
        }
    }
}

```

```

    },
    "date":{
        "type": "string"
    },
    "title":{
        "type": "string"
    },
    "author":{
        "type": "string"
    },
    "text":{
        "type": "string"
    }
}
}

```

- Классификация текста.

POST /rest-api/classification-new

Точка входа принимает в теле запроса json, который содержит классифицируемый текст. Возвращает json, который содержит результат классификации - категорию введенного текста. Json-схема входного json представлена в листинге 6. Json-схема выходного json представлена в листинг 7.

Листинг 6 - Json-схема входного json для классификации текста.

```

{
    "$schema": "http://json-schema.org/draft-04/schema#",

```

```

    "title": "textForClassify",

    "description": "text for classify category",


    "properties":{

        "text":{

            "type": "string"

        }

    }

}

```

Листинг 7 - Json-схема выходного json для вывода результата классификации текста.

```

{

    "$schema": "http://json-schema.org/draft-04/schema#",

    "title": "resultClassification",

    "description": "category of text for classification",


    "properties":{

        "category":{

            "type": "string"

        }

    }

}

```

3.3 Используемые технологии.

Для реализации серверной части использовался язык программирования Python, а также фреймворк Django.

Для создания модели классификатора используется модуль языка Python scikit-learn.

В качестве средства реализации пользовательского интерфейса используется HTML и Javascript.

В качестве системы контроля версий использовался сервис GitHub, представляющий из себя графический интерфейс для технологии git – распределенной системы управления версиями.

3.4 Паттерны проектирования.

При разработке использовался паттерн MVC, который позволяет разрабатывать бизнес-логику и визуальное представление отдельно.

4. Описание классификатора.

4.1 Классы.

Классификатор определяет следующие классы:

- Hardware;
- Software;
- IT рынок;
- Новости сайта.

4.2 Вектор признаков.

За основу модели представления текста новостных статей использовался метод bag-of-words. Так же была сделана предобработка текста, были убраны знаки пунктуации, а также стоп-слова. Список стоп слов был взят из

библиотеки nltk. Также для предобработки текста, происходит леммитизация текста - приведение слова к смысловой канонической форме слова с помощью библиотеки pymystem3.

Вектором признаков будет являться вектор, элементы которого - это количество вхождений слова среди всех слов из всех новостей (словаря). Его длина будет равна длине словаря.

4.3 Модель машинного обучения классификатора.

В качестве модели машинного обучения был выбран метод опорных векторов.

4.4 Обучающая и тестовая коллекция документов.

Классификатор обучался на 1657 новостных статьях. В качестве тестовой коллекции были взяты 416 новостных статей из общей коллекции текстов.

4.5 Оценка точности классификации.

Оценка точности разработанного классификатора оценивалась по нескольким критериям:

- точности (precision);
- полноте (recall);
- f1-score;
- точности (accuracy).

Точность (accuracy) классификации равна 99%. Оценки точности по категориям представлены в таблице 1.

Таблица 1 – Оценка точности обученного классификатора по категориям.

Категория	Точность (precision)	Полнота (recall)	F1-мера
Новости Hardware	0,99	0,99	0,99
Новости Software	0,98	0,97	0,98
Новости IT-рынка	0,98	0,99	0,98
Новости Сайта	1,00	0,98	0,99

5. Описание инфраструктуры разработки.

5.1 Система контроля версий.

Проектирование и реализация приложения велась локально на компьютере разработчика при помощи Microsoft Visual Studio 2017. При разработке системы использовалась система контроля версий git. Исходный код хранится локально на компьютере разработчика, и удаленно, на сайте сервиса GitHub: <https://github.com/MixBass/Task1>

5.2 Сборка и развертывание приложения.

Для успешного запуска веб-приложения необходимо наличие интерпретатора Python 3.0 (64 бит) или выше, а также операционной системы Windows 7 и выше. Далее, необходимо скачать архив с проектом, разархивировать папку NewsClassification.

Далее через командную строку активировать виртуальное окружение python с помощью команд:

- `cd {Путь до папки}\NewsClassification;`
- `env\Scripts\activate.bat.`

После активации окружения, достаточно запустить web-приложение по данной команде: `manage.py runserver`.

5.3 Обновление модели классификатора.

Для обновления модели классификатора необходимо запустить файл `fitModel.py` в директории `NewsClassifier`.

5.4 Леммитизация коллекции текстов.

Для леммитизации коллекции текстов необходимо запустить файл `lemmitizeCollection.py` в директории `NewsClassifier`.

5.5 Загрузка списка стоп слов.

Для загрузки списка стоп слов необходимо запустить файл `downloadStopWords.py` в директории `NewsClassifier`.

6. Результаты тестирования.

Для тестирования системы применялось ручное тестирование. Тестирование проводилось методом черного ящика. В процессе тестирования использовался «Яндекс.Браузер» версии 21.2. Краткое описание проводимых тестов и результатов в таблице 2.

Таблица 2 – Краткое описание тестов и их результат.

Краткое описание теста	Результат тестирования
Ввод текста новостной статьи для классификации и последующая его классификация	Был выведен корректный результат классификации текста
Ввод пустого текста новостной статьи для классификации и последующая его классификация	Было выведено сообщение о том, что был введен пустой текст

Продолжение таблицы 2 – Краткое описание тестов и их результат.

Удаление новостной статьи	Было выведено сообщение о том, что статья удалена
Вывод несуществующей новостной статьи	статьи Было выведено сообщение о том, что такой статьи не существует
Вывод страницы с новостными статьями	При загрузке страницы была выведена первая страница с коллекцией новостей
Добавление новой новостной статьи	Было выведено сообщение о том, что статья успешно добавлена.
Добавление новой новостной статьи с некорректными параметрами	Было выведено сообщение о том, что параметры некорректны
Изменение новостной статьи	Было выведено сообщение о том, что статья успешно изменена
Вывод новостной статьи	При загрузке страницы с новостной статье, информация по ней была успешно отображена

7. Интерфейс и возможности системы.

Система предоставляет пользователю классифицировать текст и просматривать коллекцию новостных статей, а также производить операции с отдельными новостями.

Классификация текста происходит при нажатии кнопки «Классифицировать новость». Интерфейс страницы классификации текстов представлен на рисунке 2.

Сервис по классификации новостей

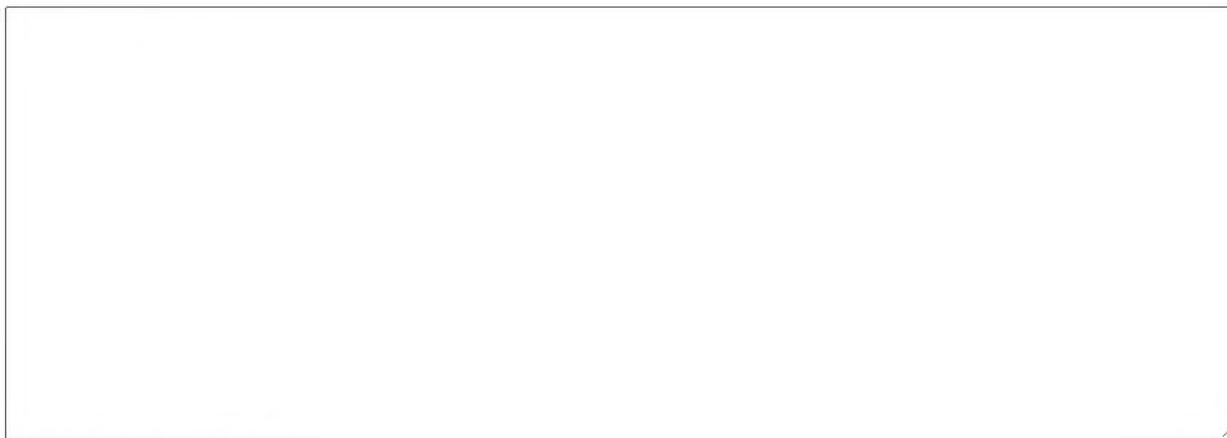
[Классификатор новостей](#)

[Коллекция новостей](#)

[Добавить новость](#)

Классификатор новостей

Введите текст новости



Классифицировать новость

Рисунок 2 – Интерфейс классификации новостной статьи.

Так же система имеет возможность добавления новостной статьи и выбора её категории, с последующим просмотром в коллекции новостей. Предусмотрена защита при заполнении данных от некорректного ввода ссылки на новость и даты публикации, защита от пустых полей в заголовке, авторе и тексте новости, и ссылке на новость. Интерфейс добавления новости изображён на рисунке 3.

[Классификатор новостей](#)
[Коллекция новостей](#)
[Добавить новость](#)

Добавить новость

Ссылка на новость:

Это не ссылка на новость

Категория:

Новости Hardware ▾

Дата публикации:

У даты некорректный формат

Заголовок:

Заголовок не заполнен

Автор:

Никнейм автора новости не заполнен

Текст новости:

Текст новости не заполнен

Добавить новость

Некоторые поля некорректно заполнены

Рисунок 3 - Интерфейс добавления новостной статьи.

В «Коллекция новостей» находятся все новостные статьи, которые выводятся списком постранично, переход на другую страницу с новостями происходит по нажатию на номер страницы снизу всех статей. Для поиска статей используется фильтрация по категории, а также поиска по ключевым словам. Интерфейс поиска и фильтрации новостных статей изображён на рисунке 4.

Сервис по классификации новостей

[Классификатор новостей](#)

[Коллекция новостей](#)

[Добавить новость](#)

Коллекция новостей

Кол-во новостей: 4

[Термосумки STARWIND для пикников и походов](#)

Категория: Новости IT-рынка

Автор: admin

[ASUS Republic of Gamers применяет жидкий металл в термоинтерфейсе игровых ноутбуков 2020 года с процессорами Intel Core](#)

Категория: Новости IT-рынка

Автор: admin

[Термопаста ProArtist W15 комплектуется трафаретом для равномерного нанесения](#)

Категория: Новости Hardware

Автор: Алексей Сычёв

[Новая термопаста Arctic MX-5 поступит в продажу 15 марта](#)

Категория: Новости Hardware

Автор: Алексей Сычёв

1

Рисунок 4 - Интерфейс поиска и фильтрации новостных статей.

Так же имеется возможность редактирования или удаления новостной статьи, посредством нажатия на кнопки «Редактировать» и «Удалить». Интерфейс редактирования и удаления новостной статьи изображён на рисунке 5.

Сервис по классификации новостей

[Классификатор новостей](#)

[Коллекция новостей](#)

[Добавить новость](#)

Термосумки STARWIND для пикников и походов

Ссылка на новость: <https://overclockers.ru/itnews/show/91472/termosumki-starwind-dlya-piknikov-i-pohodov>

Категория: Новости IT-рынка

Дата публикации: 17 мая 2018, четверг 14:26

Автор: admin

Модельный ряд STARWIND пополнился изотермическими термосумками серии СВ. В новой коллекции представлено сразу пять моделей (СВ-112/СВ-117/СВ-120/СВ-125/СВ-138), которые изнутри отделаны специальным теплоизолирующим материалом, способным надолго сохранить холод, даже в жару. К ключевым характеристикам можно отнести сохранение температурного режима - до 12 часов и максимальное охлаждение - до 16°C ниже температуры окружающей среды. Сумки достаточно вместительны (от 12л до 38л), подходят для хранения и пищи, и напитков. Они имеют одно основное отделение, закрывающееся на молнию. Для удобной переноски термосумки оснащены двумя ручками и отстегивающимся плечевым ремнем. Новинки от STARWIND идеально подходят для отдыха на природе, пикников, туристических походов и путешествий. Новинки уже поступили в розничную продажу. Компания MERLION является эксклюзивным дистрибьютором STARWIND на территории России.

[Редактировать](#) [Удалить](#)

Рисунок 5 - Интерфейс редактирования и удаления новостной статьи.

8. Анализ полученных результатов.

Таким образом, был спроектировано и реализовано web-приложение для классификации новостных статей и работы с их коллекцией.

Разработанное web-приложение не требует больших затрат на поддержку, сопровождение и обновление. Дизайн простой и понятный.

К приложению легко добавить новые web-страницы и функционал. В плане отказоустойчивости, web-приложение защищено от ввода некорректных данных введенных пользователем.

Классификатор имеет хорошую точность распознавания категорий, а именно 99% процента.

9. Список использованных источников.

1. Overclockers.ru [Электронный ресурс]. – Режим доступа: <https://overclockers.ru/>. – Заглавие с экрана. – (Дата обращения: 29.12.20).

2. Python [Электронный ресурс]. – Режим доступа: <https://www.python.org/>. – Заглавие с экрана. – (Дата обращения: 29.12.20).

3. Django [Электронный ресурс]. – Режим доступа: <https://docs.djangoproject.com/>. – Заглавие с экрана. – (Дата обращения 29.12.20).

4. Справочник Javascript [Электронный ресурс]. – Режим доступа: <https://javascript.ru/>. – Заглавие с экрана. – (Дата обращения 29.12.20).

5. Учебник HTML [Электронный ресурс]. – Режим доступа: <https://schoolsw3.com/> – Заглавие с экрана. – (Дата обращения 29.12.20).

6. Scikit-learn [Электронный ресурс]. – Режим доступа: <https://scikitlearn.org/>. – Заглавие с экрана. – (Дата обращения 29.12.20).