**Софийски университет „Св. Кл. Охридски”**

Факултет по математика и информатика

*Катедра „Софтуерни технологии”*

**КУРСОВ ПРОЕКТ**

По “Извличане на информация”

**Тема: Жанров класификатор и търсачка на книги на български**

**Автор:**

Диана Иванова,  
  Ф№ 26351

Специалност: Софтуерни технологии

**Ръководител:**проф. Иван Койчев

София, 2020 г.

Декларация за липса плагиатство:

* Плагиатство е да използваш, идеи, мнение или работа на друг, като претендираш, че са твои. Това е форма на преписване.
* Тази курсова работа е моя, като всички изречения, илюстрации и програми от други хора са изрично цитирани.
* Тази курсова работа или нейна версия не са представени в друг университет или друга учебна институция.
* Разбирам, че ако се установи плагиатство в работата ми ще получа оценка “Слаб”.

Съдържание

[Увод 3](#_Toc64582128)

[Реализация 3](#_Toc64582129)

[Използвани технологии 3](#_Toc64582130)

[Корпус 3](#_Toc64582131)

[Реализация на търсачка 3](#_Toc64582132)

[Реализация на класификатор 3](#_Toc64582133)

[Тестване и резултати 4](#_Toc64582134)

[Резултати търсачка 4](#_Toc64582135)

[Резултати класификатор 4](#_Toc64582136)

[Примерни резултати от ръчно тестване 5](#_Toc64582137)

[Публично хранилище Github. 5](#_Toc64582138)

[Заключение 5](#_Toc64582139)

[Използвана литература 6](#_Toc64582140)

# Увод

Целта на този документ е да се представи и опише програма, която изпълнява текстова класификация по жанрове на книги, както и търсачка за книги по съдържанието им. Обучението на класификатора е с учител – използвана е *Support Vector Machine*. Част от книгите нямат записан жанр, но той може лесно да бъде определен от обучения класификатор. Търсачката прави заявки от тип свободен текст и връща резултати подредени по ранг. Търсачката работи с индекс върху 37785 текстови файла. Реализирана е посредством elastic search.

Chitanka.info е най-големия български сайт със книги във цифров формат, който е безплатен и общодостъпен. Сайтът е некомерсиален и създаден от доброволци и разчита най-вече на дарения. Бих желала да допринеса за каузата.   
 В момента търсачката в chitanka.info може да търси само по заглавия и автори.   
Би било добре да има възможност за по-детайлно и ефективно търсене, така че да се улесни изборът на книга за потребителя или за да може той по-лесно да събира информация от книги.

# Реализация

## Използвани технологии

Проектът е имплементиран на python 3.6. За целта за използвани библиотеките *nltk*, *scikit-learn, elasticsearch*. За стемер на български е използвана ре-имплементирана версия на BulStem на езика python. Стоп думите са избрани от текстов файл *BulgarianST.txt*

## Корпус

Корпусът от книги е събран изцяло от българската електронна уеб библиотека chitanka.info. Книгите са свалени локално в txt формат чрез приложение предоставено от библиотеката. Всички книги са на български език. За всяка книга са предоставени автор, заглавие, жанр, литературен тип, година и съдържание. Книгите са преработени в *json* формат.

## Реализация на търсачка

Търсенето на книги става посредством заявки от тип свободен текст – чрез така наречената *full text query*. Могат да се намират книги по съдържанието им. Резултатите се сортират според техния score, който е базиран на *tf-idf* – това е алгоритъма *BM25,* който е алгоритъмът по подразбиране в elastic search.  
Има възможност за конфигурация на заявката чрез задаване на флаг fuzziness=True като аргумент на заявката. По този начин резултатните съвпадения могат да имат максимална разстояние на редактиране със стойност 2 (*edit distance*) спрямо заявката. По този начин могат да се открият резултати при заявки с до две грешки на дума.

## Реализация на класификатор

Всяка книга се класифицира според съдържанието и, като възможните категории са 12: фентъзи, хорър (литература на ужаса), фантастика, криминална литература, драма, детски и юношеска литература, любовен роман, приключенска литература, трилър, философия, исторически роман, хумор. В корпуса има и други жанрове, но поради малкия брой примери за тях, те не са добавени в обучението и тестването.

Обучението се извършва върху 1500 примера избрани на случаен принцип, като за всеки жанр има приблизително почти един и същ брой примери. Целта е да се избегне “пристрасност” към някоя категория. Всеки пример е съдържанието на книга. На всеки пример съответства категория, която представлява жанр на книгата.

Обучението на класификатора е с учител – използвана е *Support Vector Machine* с линеен *kernel*. *SVM* е приложена върху матрица на тренировъчните данни, както и върху вектор от категории. Категориите във векторът са закодирани чрез *LabelEncoder*. Матрицата на тренировъчните данни е получена чрез векторизация на съдържанието документите чрез *TfidfVectorizer*. Игнорирани са думи с документна честота по-малка от 2 процента. (min\_df=0.02), тъй като иначе се получава *overfitting*.

Преди документите да бъдат векторизирани те са предварително обработени в *json* файлове, след което са токенизирани, нормализирани, приложен е стеминг и стоп-думите са премахнати.

# Тестване и резултати

## Резултати търсачка

Тестването е проведено ръчно.  
Резултатите от търсенето имат json формат, който съдържа атрибути индекс, тип, идентификатор на документа, точки и полета с името на автора на книгата и заглавието и.

Първия резултат от примерно търсене със заявка “фибоначи мона лиза”:

[

{

"\_index": "books",

"\_type": "\_doc",

"\_id": "QMOH43YBH4VPkmMf6AhP",

"\_score": 23.453377,

"fields": {

"author": [

"Дан Браун"

],

"title": [

"Шифърът на Леонардо"

]

}

},

## Резултати класификатор

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **kernel** | **features** | **C parameter** | **Training data** | **result** |
| rbf | 9000 | C=1 | 2000 | 47.16666666666667 |
| linear | 9000 | C=4 | 2000 | 74.17355371900827 |
| linear | 5000 | C=1 | 2000 | 74.27385892116183 |
| linear | 2000 | C=1 | 2000 | 74.375 |
| linear | 9000 | C=1 | 6000 | 76.52905198776759 |
| linear | 8000 | C=3 | 2000 | 76.62771285475793 |
| linear | 9000 | C=1 | 2000 | 77.29166666666667 |

Резултати - точност на класификатор

Тестовете за приложени върху 500 текста, като тяхното множество не се пресича с това на тренировъчните данни. Обучената *SVM* прави предположение за категориите на 500те тестови текста, след което те са сравнени с реалните категории.

Проведени са опити за обучение и тест върху по-малки и по-големи тренировъчни матрици. Направени са и опити с повече текстове за обучение, но тогава резултатите не се подобряват. За сметка на това се губи баланса между броя примери от всяка категория, тъй като има категории с по-малко от 200 примера.   
Алгоритъмът е тестван и с различни видове kernel като *rbf* и *polynomial* – те обаче изискват повече ресурси и не подобряват резултатите.

## Примерни резултати от ръчно тестване

Агата Кристи - Дамата с воала : Криминална литература

Алеко Константинов - Разни хора, разни идеали I : Хумор

Марк Твен - Писма от Земята : Философия

Айзък Азимов - Четириизмерната котка : Хумор

Тери Пратчет - Автентичната котка : Хумор

Стивън Кинг - Сънят на Харви : Любовен роман

Ърнест Хемингуей - Зелените хълмове на Африка : Приключенска литература

Антоан дьо Сент-Екзюпери - Цитадела : Философия

Джеръм К. Джеръм - Какво струва да се покажеш любезен : Хумор

Стивън Кинг - Дяволската котка : Хорър (литература на ужаса)

Емили Гифин - Нещо назаем : Любовен роман

Христо Смирненски - Юноша : Детска и юношеска литература

Рей Бредбъри - Градът, където никой не спира : Фантастика

# Публично хранилище Github.

Кодът се съдържа в папка. Може да се достъпи и на следния линк: <https://github.com/IvanovaD/Search_Engine_and_SVM_Classificator>

Заключение  
Възможно е да се добави разширено търсене чрез синоними и проверка на правопис. Търсенето на синоними би могло да се осъществи с *BulTreeBank* ресурси.  
Допълнителни експерименти могат да се проведат с други класификатори и да се направи сравнителен анализ спрямо резултатите – например класификация с Наивен Бейсов Класификатор или друг алгоритъм.   
Друго възможно подобрение е разширение на жанровете с лирически, тъй като за момента класификатора работи само с проза.

# Използвана литература

1. [Bedi, Gunjit. “Simple Guide to Text Classification(NLP) Using SVM and Naive Bayes with Python.” *Medium*, Medium, 13 July 2020,](mailto:1)%09https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34)
2. [Joachims, Thorsten. *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*. Universitat Dortmund,](http://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf.)
3. [“1.4. Support Vector Machines.” *Scikit*, scikit-learn.org/stable/modules/svm.html.](https://scikit-learn.org/stable/modules/svm.html)
4. [“Free and Open Search: The Creators of Elasticsearch, ELK & Kibana.” *Elastic*,](http://www.elastic.co/)
5. [Shaikh, Javed. “Machine Learning, NLP: Text Classification Using Scikit-Learn, Python and NLTK.” *Medium*, Towards Data Science, 30 Oct. 2017,](file:///C:\Users\Diana%20Ivanova\AppData\Roaming\Microsoft\Word\towardsdatascience.com\machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a)
6. [“Произведения.” *Моята Библиотека*](file:///C:\Users\Diana%20Ivanova\AppData\Roaming\Microsoft\Word\chitanka.info\texts)