# Lyrics based recommendation system

# Lyrics based recommendation system

- Required Technics
  - TF-IDF
  - Singular Value Decomposition
  - Data Pre-processing
  - Web Scraping
  - (If you can) Web programming

# Limitation of lyrics

- Not a complete sentence
- Grammatically wrong
- Participation of multiple users. Not uniform

→ Decrease embedding quality

# TF-IDF(Term Frequency - Inverse Document Frequency)

- TF denotes the frequency that how many specific word is used in a particular document(lyric).
- DF indicates the number of documents(lyrics) containing the word.
- IDF means the total number of docs(lyrics) divided by DF and take a logarithm. Reduce the weight of words that lack information.

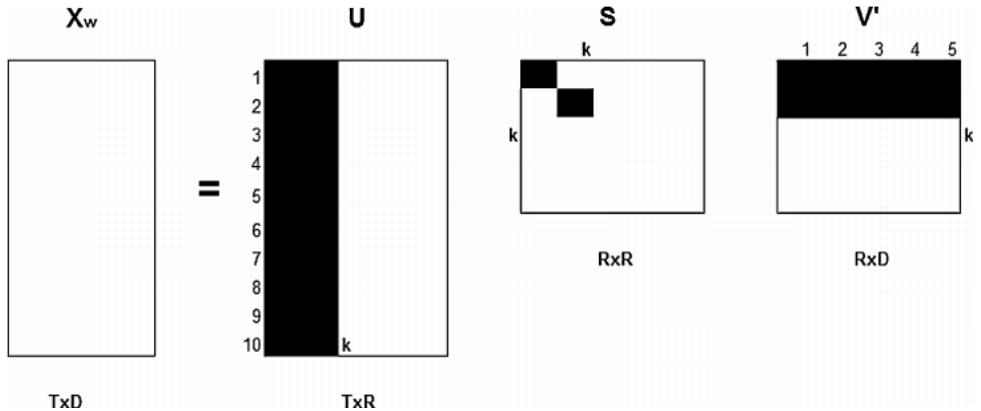
  (Effectively jettison 을, 를, 이, 가 etc.)

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of $i$ in $j$
$df_i$ = number of documents containing $i$
$N$ = total number of documents

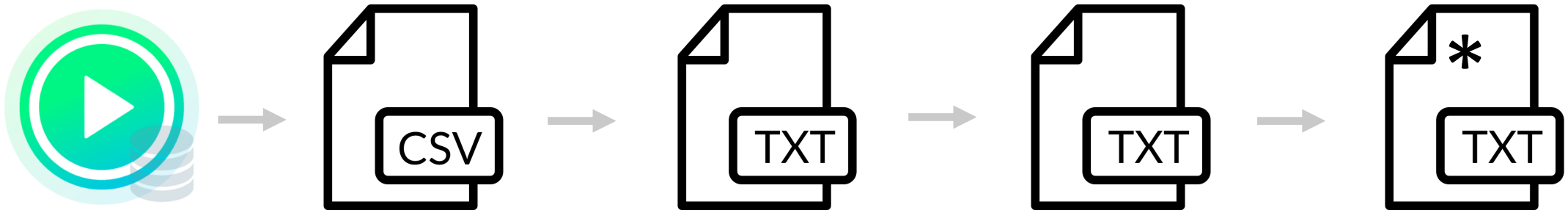
# Singular Value Decomposition

- Matrix itself contains unnecessaries(sparse matrices)
- Lower a dimension of matrix while maintaining its core information.



# Process(Lyrics)

Object : 1.2 million lyrics in Naver Music
Condition : Include 한글, alphabet only




Extract required data from Database

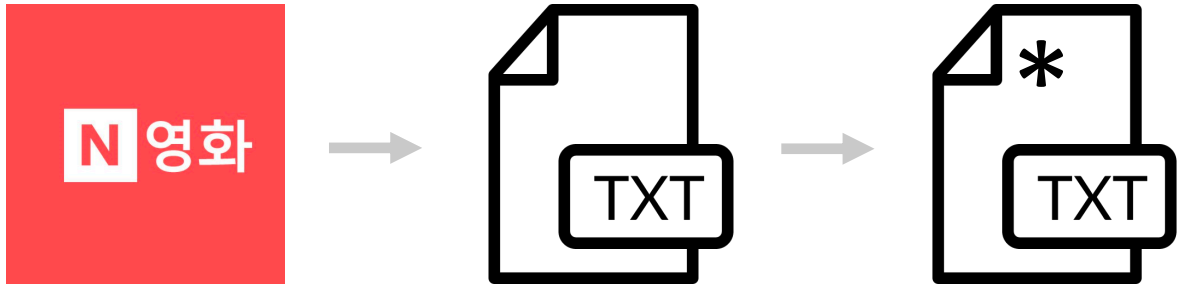
Extract LYRIC column

Eliminate symbols, stop words

Extract nouns via Mecab, nltk

# Process(Movie)

Object : 14,664 movies in Naver Movie(More than 5 reviews per 1 movie )
Condition : Include 한글 only



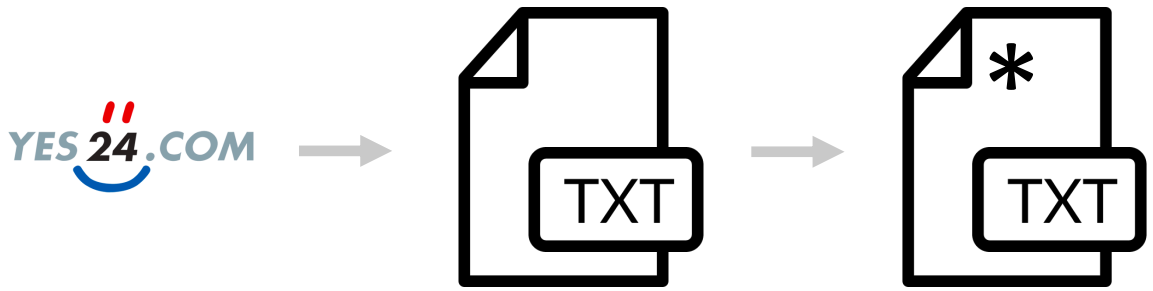

Collect titles, movie overviews, poster urls by web crawler

Extract nouns via Mecab from movie overviews

Web Cralwer Code : https://github.com/vctr7/Web_Crawling/blob/master/Naver_movie_crawling.py

# Process(Book)

Object : 5,958 books in online bookstore. Bestsellers in novel, poet, essay, religion genre.
Condition : Include 한글 only

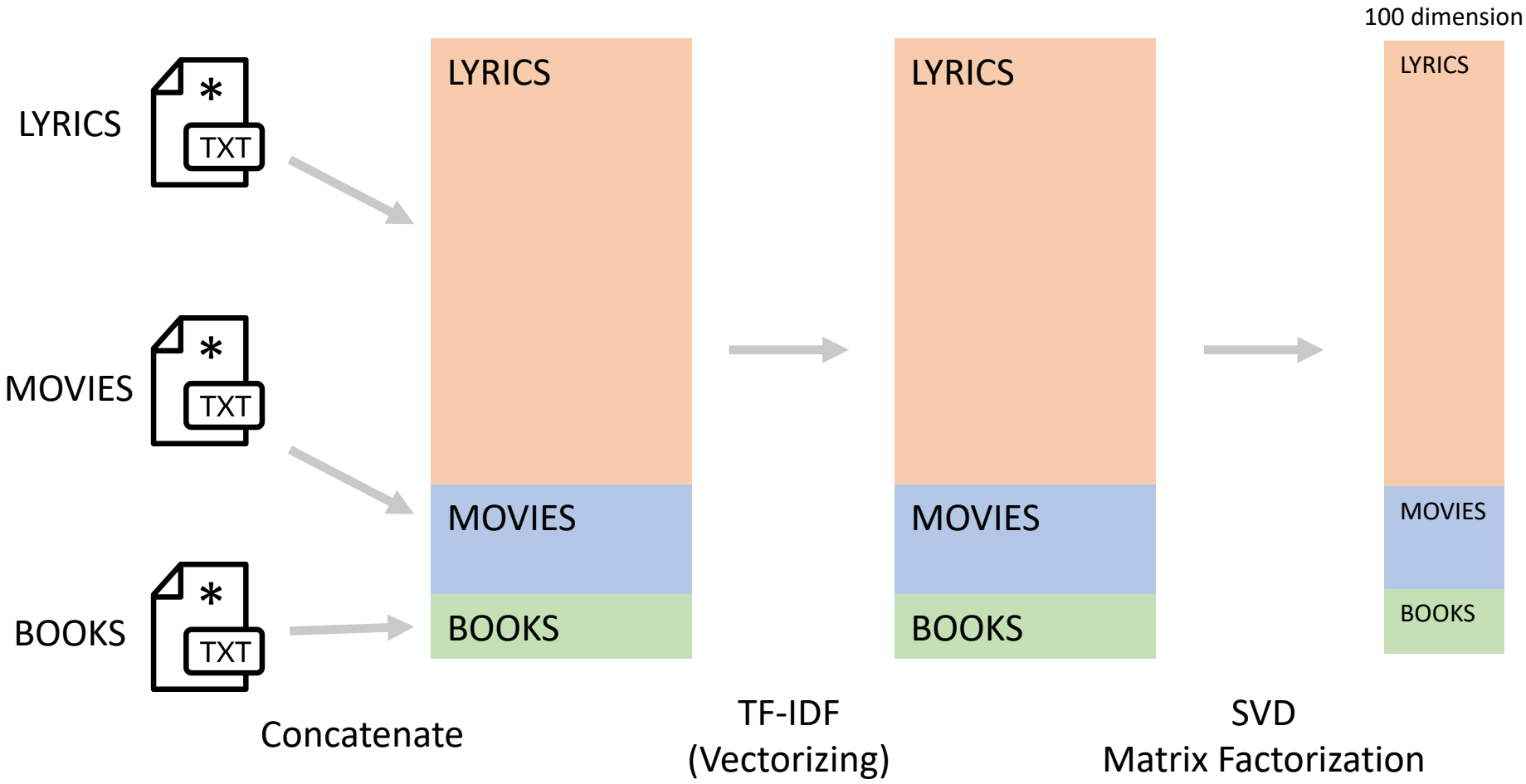



Collect titles, book overviews, cover urls by web crawler

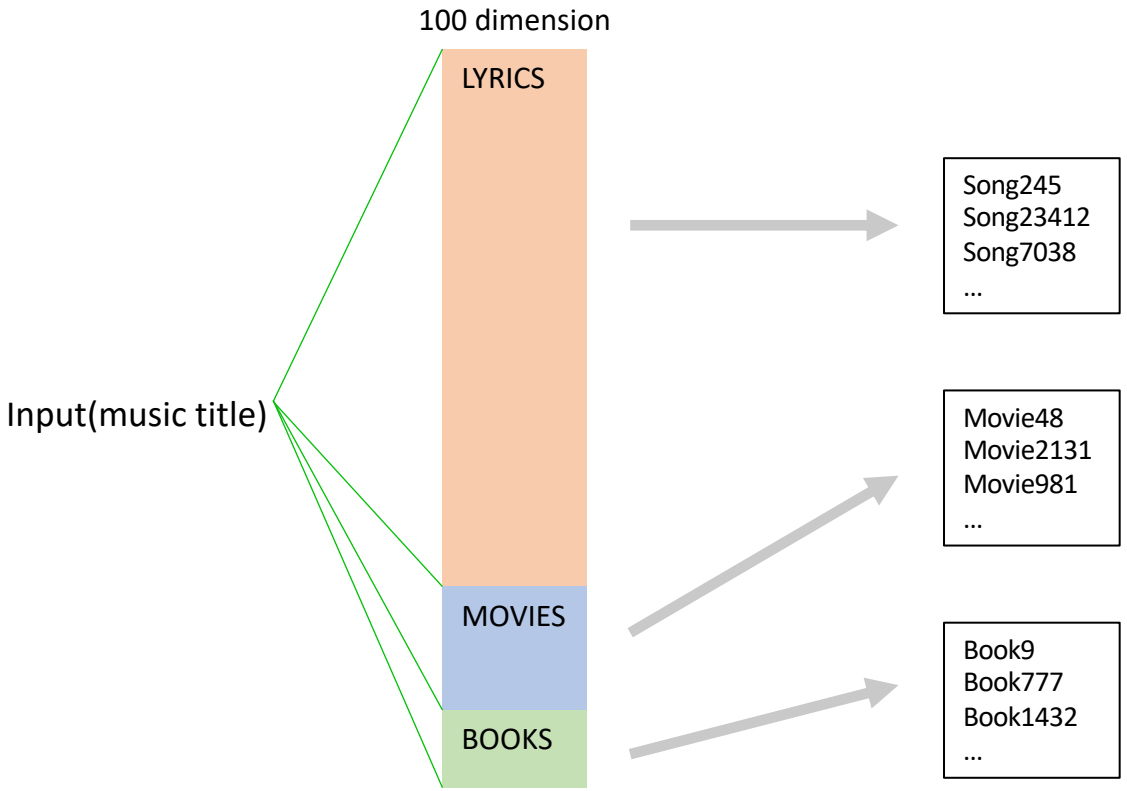
Extract nouns via Mecab from movie overviews

Web Cralwer Code : https://github.com/vctr7/Web_Crawling/blob/master/yes24_book_crawling.py

# Process


LYRICS
TXT
MOVIES
TXT
BOOKS
TXT
LYRICS
MOVIES
BOOKS
LYRICS
MOVIES
BOOKS
100 dimension
LYRICS
MOVIES
BOOKS
Concatenate
TF-IDF
(Vectorizing)
SVD
Matrix Factorization

# Process




Finish? Not yet.


Calculate cosine similarity on each section

Sort results by cosine similarity in descending order

Visualize outcomes

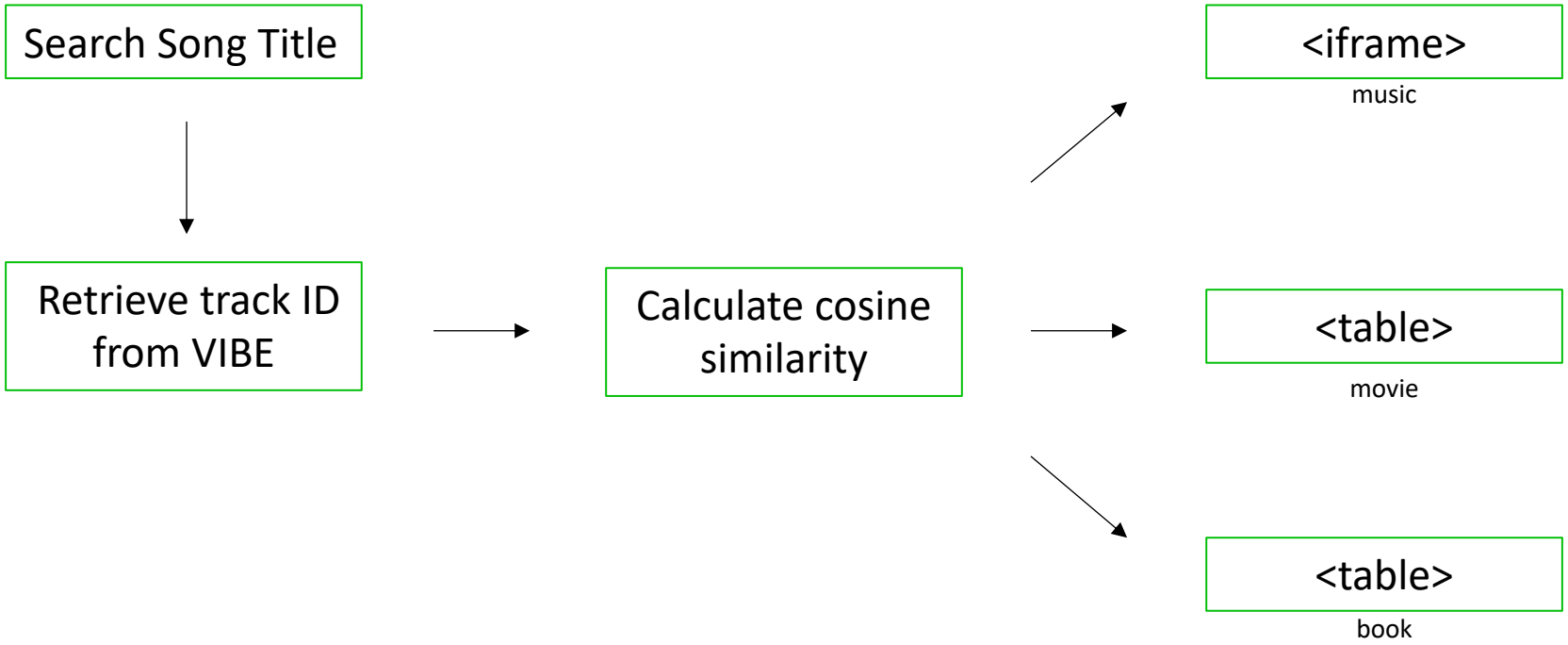
# Process

- Required elements
  - numpy
  - sklearn
  - selenium
  - Firefox driver

- Each section(songs, movies, books) have 5 contents.(total 15)
- Uses <iframe> to show VIBE website
- Visualize images using urls data

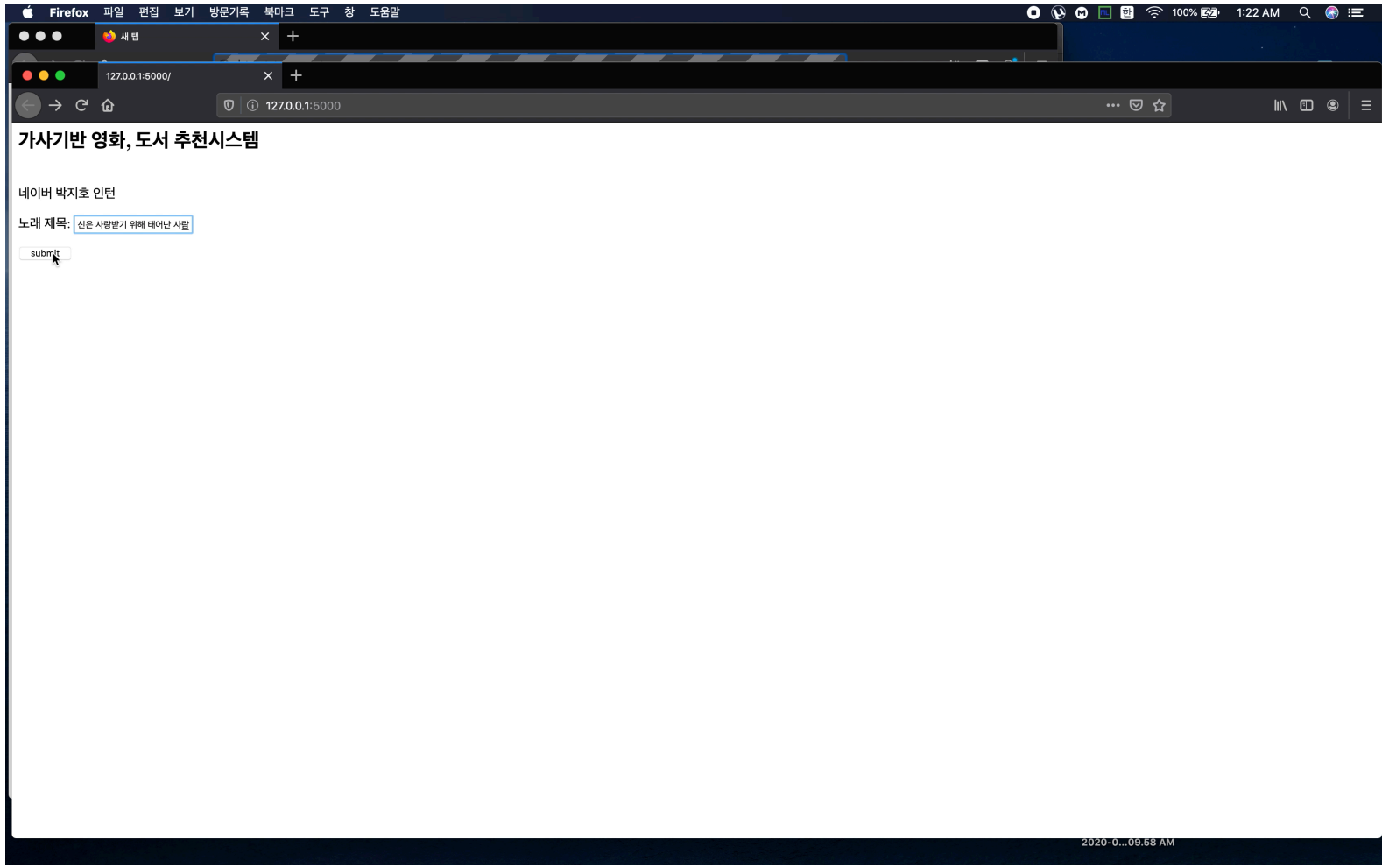
https://github.com/vctr7/Recommedation_Sys/blob/master/templates/result.html

# Process




https://github.com/vctr7/Recommedation_Sys/blob/master/app.py

# Video


Firefox 파일 편집 보기 방문기록 북마크 도구 창 도움말
100% 1:22 AM
새 탭
127.0.0.1:5000/
127.0.0.1:5000
가사기반 영화, 도서 추천시스템
네이버 박지호 인턴
노래 제목:
신은 사랑받기 위해 태어난 사람
submit
2020-0...09.58 AM

# Conclusion

- Overview does not fully contain its story and include many commercial copies. This may account for huge portion of miss rate.
- Nonetheless, it shows correlation(tendency) between input and outputs. (Christian music -> Christian books)

- As a result, if fine text data are given, lyrics could contribute to enhancement of recommendation system in music service industry.

https://github.com/vctr7/Recommedation_Sys/blob/master/templates/result.html