ПОДХОДИ ЗА ОБРАБОТКА НА ЕСТЕСТВЕН ЕЗИК

Курсова работа

Студент: Диана Маркова, **МП** ИИ, **ф.н.** 5MI3400638

Тема: Оценка за популярност на песни спрямо текст

Предметна област : Music Information Retrieval

СЪДЪРЖАНИЕ

1. Въведение

1.1. Контекст

Избраната тема принадлежи на областта на Music Information Retrieval. Проблемът има практическо значение, особено в текущото положение на музикалната индустрия - всеки нов музикален артист се сблъсква с по-усилена конкуренция от всякога. За да може да осигури успеха си, той (или тя) се нуждае от метод, който количествено да оцени вероятността за успешно пласиране на изкуството си.

1.2. Формулиране на задачата

Задачата се разглежда под формата на две разновидности - класификация и регресия. Обикновено това се диктува от наличните данни - дали етикета, който представлява популярност е от типа *хит или не-хит* (бинарна класификация) или представлява *коефициент на популярност*, в който случай задачата е регресионна. Изборът на предикторните променливи и в двата случая варира, а разнообразието, мултимодалността и пълнотата им е от огромна важност за успеха на решението.

1.3. Съществуващи решения

Можем да разделим подходите за решаване на задачата на 2 основни вида по отношение на извличането на атрибути от текст. Допълнителна характеризация може да направим и по начина на формиране на свойства, свързани с аудио. SOTA се постига при комбинирано използване на характеристики, извлечени от аудио, текст, информация от по-високо ниво за аудиото и метаданни[1,2] (на база проучванията, които използват балансирана и голяма извадка данни).

1.4. Данни

В нашия случай наличните данни съдържат коефициент на популярност (като процент).

1.5. Цели

Повечето съществуващи проучвания върху задачата обикновено предлагат решения и се целят да подобрят предишни. Няма експлицитно проучване за ефективността на полученото представяне на семантиката на текст, когато такова се използва.

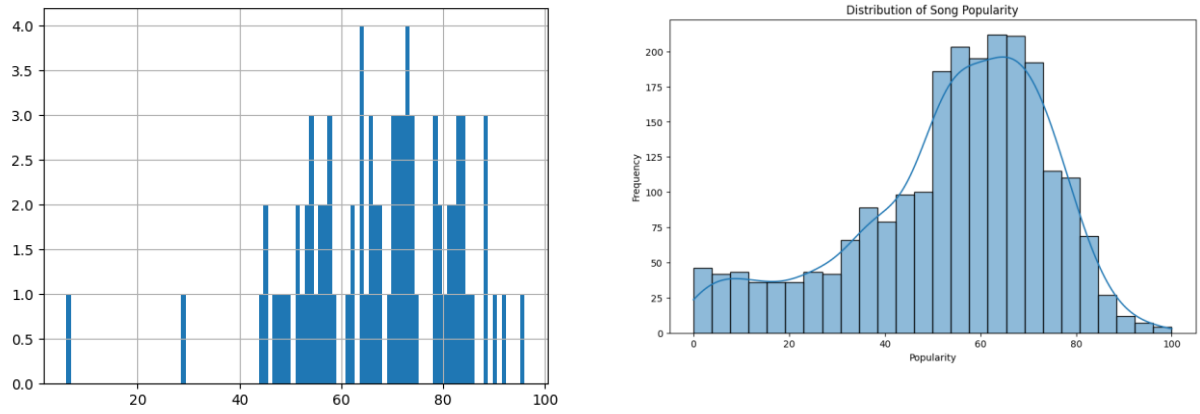
Целите на проекта, съобразно ограниченията, наложени от данните, времето и наличната изчислителна мощ, формулираме така: (1) Качествена оценка на способността на **encoder** модел за представяне на семантиката на текст на песен във

векторно пространство; (2) Количествена оценка за приноса на семантиката на текст в предсказване на популярност, на база сравнение с по-класически NLP подход за получаване на представяне на текст.

Курсовата работа има следната структура: Секция 2 разглежда процеса по събиране на данни, формирането на набори от данни и описание на методите за проведените експерименти, насочени към постигане на целите; Секция 3 се фокусира върху дискутиране на резултатите от проведените експерименти; Предоставя се заключение и предложения за бъдещи подобрения в Секция 4.

2. Материали и методи

2.1. Използвани данни



(a) (b)

Фигура 1. Разпределение на популярност в **(a)** Spotify Top Songs 2024 и **(b)** Popularity and Audio Features набори от данни

2.1.1. Набор от данни *Spotify Top Songs 2024 (текст и популярност)*

Беше съставен от песни, които попадат в плейлиста, който се генерира ежегодно от Spotify за потребителя за 2024г. от най-често слушаните песни. Бяха премахнати песни, които нямат текст, и финално се съставляваше от **83** песни. Поради промени в политиката на компанията, вече аудио атрибути на песни не могат да бъдат извлечени чрез API-то, което предоставят за разработчици [3,4]. Тъй като размерът на този набор данни е сравнително малък, както и от съображения за небалансирана популярност

на извадката и предвид разбирането на автора за тематиката и семантиката на песните в него, той служи за клъстериране и наблюдаване на зависимости. Текстовете на песните тук са извлечени в пълен вид посредством Genius API-то [5].

2.1.2. Набор от данни *Popularity and Audio Features (текст, аудио-характеристики и популярност)*

Този набор от данни представлява съединение на 2 отделни набора, налични в *Kaggle* и *Mendeley;* единият съдържа текст, но не и популярност на множество от песни [6] , а другият не съдържа текст, но съдържа популярност [7] . След обработка и пълно вътрешно съединение останаха **2299** песни.

Съществено е, че в този dataset текстовете на песните са откъси (сравнително кратки), върху които предварително са направени stemming/lemmatization**,** премахване на стоп-думите и пунктуацията. Например, откъс от песента Natural (на Imagine Dragons) има вида:

> *hold line give give tell house come consequence cost tell star align heaven step save cause house stand strong leave heart cast away product today prey stand edge face cause natural beat heart stone gotta cold world yeah natural live life cutthroat gotta cold yeah natural*

2.2. Извличане на характеристики. Обработка на естествен език

Popularity and Audio Features наборът от данни съдържа освен текстове и популярност, и други свойства - 'dating', 'violence', 'world/life', 'night/time', 'shake the audience', 'family/gospel', 'romantic', 'communication', 'obscene', 'music', 'movement/places', 'light/visual perceptions', 'family/spiritual', 'like/girls', 'sadness', 'feelings', 'danceability_x', 'loudness_x', 'acousticness_x', 'instrumentalness_x', valence', 'energy_x', 'topic', 'age', 'song_name', 'song_duration_ms', 'acousticness_y', 'danceability_y', 'energy_y', 'instrumentalness_y', 'key', 'liveness', 'loudness_y', 'audio_mode', 'speechiness', 'tempo', 'time_signature', 'audio_valence', 'lyrics_length'

За обработка на текстовете разлеждаме следните подходи:

2.2.1. Term frequency–inverse document frequency (tf–idf)

Tf-df е статистическа метрика за оценка на важността на дума в документ спрямо колекция от документи. Тя се задава като произведение на следните компоненти:

$$TF(t,\ d)\ =\ \frac{\text{Брой срещания на } t \text{ в документа } d}{\text{Общ брой думи в } d}$$

$$IDF(t) = \log \left(\frac{\text{Общ брой документи}}{\text{Брой документи, в които се среща } t} \right)$$

$$tf - idf(t, d) = TF(t, d) \times IDF(t)$$

Тя ще ни послужи за база, върху която да направим сравнение.

2.2.2. Токенизация и получаване на векторно представяне на текст (semantic embeddings)

Transformer моделите ни предоставят възможност за семантично кодиране на последователности по начин, вземащ предвид реда на думите и контекстуалните зависимости чрез multi-headed attention механизма [8]. Има различни разновидности на архитектурите [9], но от интерес за нас е encoder частта, тъй като чрез нея ще получим векторното представяне на последователността, която представлява текстът на песен. По-конкретно, при модели подобни на BERT, ние се интересуваме от векторът-ред, съдържащ CLS (classification token) attention стойностите. Това е така, защото той резюмира реда и връзките между думите. [10]

2.2.2.1. Bidirectional Encoder Representations from Transformers (BERT)

Използва както ляв, така и десен контекст за пресмятане на зависимости между думи. BERT е предварително трениран да решава задачите за Masked Language Modelling (MLM) и Next-Sentence Prediction (NSP). Чрез добавяне на feed-forward слой може да бъде fine-tuned за решаване на задачи като класификация на текст и Question Answering. Максималната дължина на последователност, която може да обработва, е 512 токена. [10] Това е причината да го използваме за получаване на ембединги за *Popularity and Audio Features* набора данни.

2.2.2.2. Longformer

$O(n^2)$ сложността по памет и време (n - бр. токени) на традиционните transformer модели (вкл. BERT) прави обработката на по-дълги документи непрактична. Longformer адресира този проблем, използвайки т.нар. sparse attention patterns, за да намали изискванията за памет, запазвайки контекстуалното разбиране. Така максималната дължина на последователност, която може да обработва, е 4096 токена. [11] Затова спокойно ще го използваме за генериране на ембединги на текстове за *Spotify Top Songs 2024* dataset-а.

2.3. Поток на работа. Използвани технологии

2.3.1. Експерименти

2.3.2.1. Клъстериране

Клъстерирането е по отношение на полученото векторно представяне от Longformer модела от библиотеката **transfomers** за Spotify Top Songs 2024. Алгоритъмът, който е приложен тук, е KMeans, като преди това е приложена нормализация.

2.3.1.2. Моделиране на данни. Алгоритми за МС за регресия на популярност и метрики.

Тези експерименти моделират множеството *Popularity and Audio Features*. Използваните алгоритми за самообучение са многослоен персептрон и линейна регресия. Фиксирани са стандартни метрики за оценка - Mean Absolute Error (MAE), Root Mean Squared Error (RSME), R^2.

Табл. 1. Ескперименти за регресия

№	Експеримент за регресия
(i)	Без извличане на текстови характеристики
(ii)	Извличане на текстови характеристики чрез tf-idf
(iii)	Извличане на текстови характеристики чрез BERT

2.3.2. Toolkit

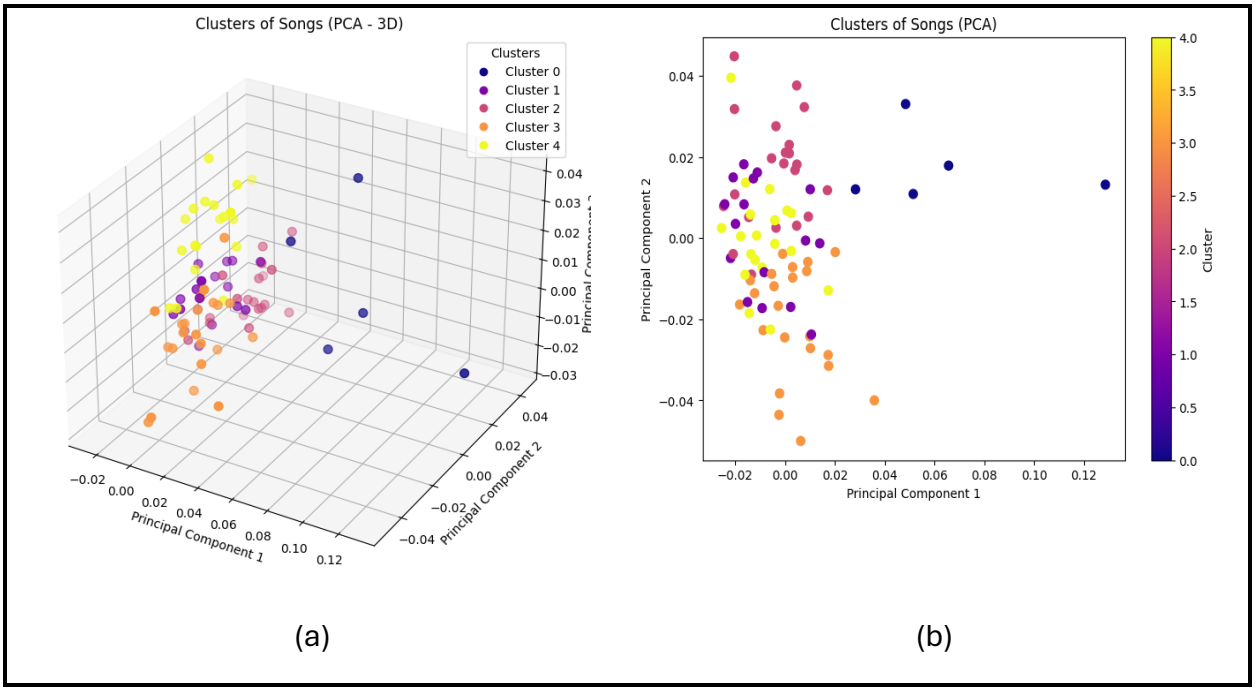
За събирането на данните за Spotify Top Songs 2024 бяха използвани библиотеки, които предоставят интерфейс на **Python** към респективните APIs на Spotify (за наименования на песни и метаданни) и Genius (за текстове) - Spotipy[12] и LyricsGenius[13].

Python заедно с **ipynb** (Interactive Python Jupyter Notebook) беше използван за построяване на наборите данни (**pandas** module), анализ и визуализация (**numpy, seaborn, matplotlib** modules), извличане на атрибути от текстовете (**transformers**, **sklearn**), както и нормализация, разделяне на данни за обучение и тестване, моделиране и оценка (**pytorch**, **sklearn** modules);

Git беше използван за version-control. Кодът на курсовата работа е достъпен в **GitHub** публично хранилище [14].

3. Резултати и дискусия

3.1. Клъстериране

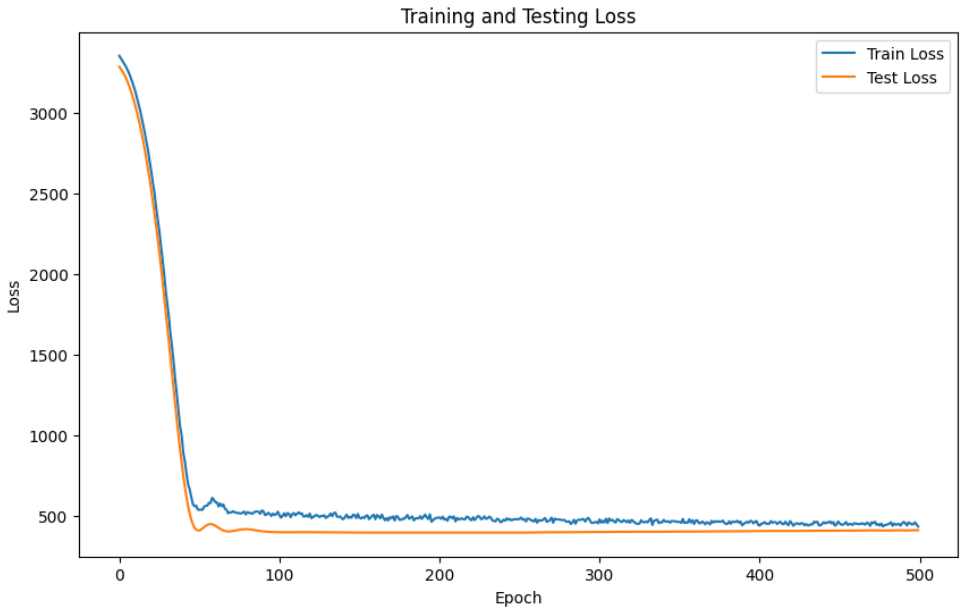


Фигура 2. Визуализация на клъстери след прилагане на PCA **(a)** 3 принципни компонента (запазват около 26% от вариацията) **(b)** 2 принципни компонента (запазват около 20% от вариацията)

Според метода на лакътя (elbow method) избрахме k=5 за KMeans. Петте клъстера наистина групират песни с подобна тематика, което показва ефективността на Longformer ембедингите. Например песните на Taylor Swift *I Can Do It With a Broken Heart*, *imgonnagetyouback* и *Fortnight*, които имат подобна тематика (това се отразява и от факта, че принадлежат на един албум - *The Tortured Poets Department*), попадат в клъстер 4, където попада и *hate to be lame* на Lizzy McAlpine, отново имаща подобна тематика. Подобно наблюдение можем да направим и за песните на Beyonce от албума *Cowboy Carter*. Отбелязваме, че **не** всички песни от един и същи артист са от един и същи клъстер. Свидетел за това е *Cruel Summer* на Taylor Swift (клъстер 2), чиято тематика и семантика е много по-различна от тези на всички песни от *The Tortured Poets Department*. Също отбелязваме, че информация за албум не се съдържа в текстовете.

3.2. Регресия

Най-ниска грешка и респективно най-висок коефициент на детерминация върху тестовото множество се получава при BERT Embeddings + Audio Features след експериментиране с различни архитектури на многослоен персептрон за моделиране на обучаващото множество от данни.



Фигура 3. Стойности на загуба при епохи на трениране върху обучителното и тестовото множество

Табл. 2. Резултати от регресия

Подход	RMSE	MAE	R^2
(i) Audio Features	23.45	549.97	-0.33
(ii) Tf-idf + Audio Features	23.51	552.84	-0.36
(iii) BERT Embeddings + Audio Featurs	20.26	410.46	0.0085

Ниският коефициент на детерминация е вероятно следствие от малката извадка.

4. Заключение и бъдещо развитие

4.1. Извод, ограничения

На първо място, Longformer се оказва надеждно решение за получаване на ембединги на текстове на песни. Неголемият размер на извадката при моделиране за предсказване на популярност, обаче, представлява значително ограничение. Все пак, резултатите ни показват, че семантиката на текстовете на песните допринася за съответната популярност и още повече, семантичните векторни представяния, получени чрез BERT, по-ефективно кодират значението им за задачата, отколкото tf-idf.

4.2. Бъдеща работа

Мултимодален подход би бил подходящ за постигане на по-добро представяне. Чрез експериментиране с по-модерни архитектури за получаване на характеристики би могло да се подобрят SOTA резултатите на [2].

5. Декларация за оригиналност

Декларирам, че резултатите, които са получени, описани и/или публикувани от други, са надлежно и подробно цитирани в библиографията, при спазване на изискванията за защита на авторското право.

Уведомена съм, че в случай на констатиране на плагиатство в представената работа тя може да бъде отхвърлена и да последват наказателни мерки.

Подпис: 

6. Референции и литературни източници

[1] Castelli E. Hit Song Prediction system based on audio and lyrics embeddings Tesi di Laurea Magistrale in Music and Acoustic Engineering. n.d.

[2] Martin-Gutierrez D, Hernandez Penaloza G, Belmonte-Hernandez A, Alvarez Garcia F. A Multimodal End-to-End Deep Learning Architecture for Music Popularity Prediction. IEEE Access 2020;8:39361–74. https://doi.org/10.1109/ACCESS.2020.2976033.

[3] Introducing some changes to our Web API | Spotify for Developers n.d. https://developer.spotify.com/blog/2024-11-27-changes-to-the-web-api (accessed February 5, 2025).

[4] Home | Spotify for Developers n.d. https://developer.spotify.com/ (accessed February 5, 2025).

[5] Genius API n.d. https://docs.genius.com/ (accessed February 5, 2025).

[6] Moura L, Fontelles E, Sampaio V, França M. Music Dataset: Lyrics and Metadata from 1950 to 2019 2020;3. https://doi.org/10.17632/3T9VBWXGR5.3.

[7] Song Popularity Dataset n.d. https://www.kaggle.com/datasets/yasserh/song-popularity-dataset (accessed December 13, 2024).

[8] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention Is All You Need 2017.

[9] The Transformer model family n.d. https://huggingface.co/docs/transformers/en/model_summary (accessed February 6, 2025).

[10] Devlin J, Chang M-W, Lee K, Google KT, Language AI. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. n.d.

[11] Beltagy I, Peters ME, Cohan A. Longformer: The Long-Document Transformer 2020.

[12] Welcome to Spotipy! — spotipy 2.0 documentation n.d. https://spotipy.readthedocs.io/en/2.25.0/ (accessed February 5, 2025).

[13] LyricsGenius: a Python client for the Genius.com API — lyricsgenius documentation n.d. https://lyricsgenius.readthedocs.io/en/master/ (accessed February 5, 2025).

[14] HeavyHelium/song-popularity-prediction: This repository is meant to hold the code for the NLP class (@ fmi) coursework n.d. https://github.com/HeavyHelium/song-popularity-prediction/tree/main (accessed February 5, 2025).