데이터과학 최종 보고서

2026010 김민수

2226014 김서희

2226015 김은호

2226028 박은지

2425405 서현택

1. 프로젝트 개요

주제	사용자 취향 기반 음원 추천 시스템 개발
팀명	**Red Gyarados**
프로젝트 기간	[2024.10.31] - [2024.12.12]

2. 문제 정의

문제 정의	사용자의 취향의 맞춰서 음원을 추천하고자 모델을 선택
데이터셋 출처 및 설명	Spotify Data Visualization Spotify 데이터셋을 사용하여 노래명을 직접 입력하여 원하는 개수의 노래를 추천해 줄 수 있는 모델을 만들었습니다.

3. 데이터 전처리

관련 시각화를 통한 EDA

Yellowbrick 라이브러리의 시각화 도구를 사용하여 주어진 특성(Feature)들이 타겟 변수(popularity)와 얼마나 강한 상관관계를 갖는지 시각적으로 확인하였습니다.
또한 음악 데이터에서 발표된 연도별 노래 수를 시각화하는 코드를 추가하여 10년대 별로 몇 곡이 포함되었는지 막대 그래프로 나타내었고 연도별 음악 특성 변화를 시각화하는 라인 차트를 나타냈습니다. 인기 있는 상위 10개 장르에 대해 4가지 음악 특성이 어떻게 분포하는지 보여주는 막대 그래프도 나타냈습니다.

누락 데이터, 이상치, 왜곡 분포 처리

각 데이터 프레임에서 결측치가 있는 열을 확인하고 결측치를 처리한 후 IRQ 방법을 사용하여 이상치를 제거하였습니다. 왜곡된 분포를 가진 열을 로그 변환하여 분포를 개선하였습니다.

4. 모델링

장르와 노래를 통한 사용자에게 노래를 추천해주기 위해 각각의 데이터를 그룹화하는 알고리즘들을 사용하였습니다. K – means, DBscan, GMM (**Gaussian Mixture Model)**) 세 모델링을 통해 Genre_data(장르별)을 기준으로 나누며, 각각의 장르 개수에 맞춰 클러스터 10개를 사용하여 분류하였습니다.

5. 결과 및 통찰

```
song_list = [
    {'name': 'Shape of You', 'year': 2017},  # 첫 번째 곡: 'Shape of You' (2017)
    {'name': 'Blinding Lights', 'year': 2020},  # 두 번째 곡: 'Blinding Lights' (2020)
    {'name': 'Someone Like You', 'year': 2011},  # 세 번째 곡: 'Someone Like You' (2011)
    {'name': 'Dynamite', 'year': 2020}  # 네 번째 곡: 'Dynamite' (2020) - BTS 곡 추가
]
```

Spotify API를 인증하는 방식으로 제작하였으며, 곡의 존재 여부로 곡을 추가할 수도 있으며 위 사진과 같이 4개의 노래를 입력하여 아래 사진과 같이 랜덤으로 10개의 노래를 추천 받을 수 있습니다.

```
추천된 곡 리스트:
곡 이름: Prisoner (feat. Dua Lipa), 연도 : 2020, 아티스트 : ['Miley Cyrus', 'Dua Lipa']
곡 이름: Some Say - Felix Jaehn Remix, 연도 : 2020, 아티스트 : ['Nea', 'Felix Jaehn']
곡 이름: Dead To Me, 연도 : 2018, 아티스트 : ['Kali Uchis']
곡 이름: Let You Down, 연도 : 2017, 아티스트 : ['NF']
곡 이름: Stay, 연도 : 2017, 아티스트 : ['Zedd', 'Alessia Cara']
곡 이름: Whiskey Glasses, 연도 : 2018, 아티스트 : ['Morgan Wallen']
곡 이름: Waiting For Love, 연도 : 2015, 아티스트 : ['Avicii']
곡 이름: Kings & Queens, 연도 : 2020, 아티스트 : ['Ava Max']
곡 이름: Lasting Lover, 연도 : 2020, 아티스트 : ['Sigala', 'James Arthur']
```

이렇게 추천 받은 곡은 곡의 이름, 연도, 아티스트 순서대로 출력합니다.

6. 향후 방향

Colab으로 **DBSCAN 모델**을 구동했을때, **DBSCAN**이 모든 데이터 값 다 처리하다보니 시간이 많이 걸려서 여러 eps값과 min_samples값을 전부 확인하지 못하였던 부분이 가장 어려웠습니다.

이러한 부분은 더 나은 환경에서 **GMM**과 **DBSCAN**을 사용하여 K-Means보다 더 유연하게 다양한 데이터를 구동하고, 좀더 확실한 **eps, min_Samples 값**을 찾아 모델을 학습시키는 것이 필요할 것 같습니다.

또한 랜덤의 노래를 추천받는 식으로 만들었지만,

장르를 통한 노래를 추출할 수 있도록 개선하고 싶습니다.