목차

1. 과제 목표

최근 다양한 음악 어플에서 사용자의 음악 감상 기록 데이터를 토대로 유사성이 높은 음악을 추천하는 기능을 제공하고 있다. 음악 추천 기능에 사용되는 데이터는 데이터 간 불균형, 이상치 및 결측치 등 분석의 방해 요소가 존재할 수 있어 분석 시 정확도에 악영향을 줄 수 있기 때문에 적절한 전처리 과정이 필수적이다. 본 과제에서는 재생목록 데이터에 대해 다양한 전처리 기법을 적용해보고 이를 토대로 구축된 모형들의 성능을 분석 및 비교함으로써, 음악 추천을 위한 재생목록 데이터에 가장 효과적인 전처리 방법에 관한 정보를 제공하고자 한다.

2. 과제 내용

본 과제의 전체적인 동작 과정은 다음과 같다.

(1) 음악 플레이리스트 데이터를 입력으로 받는다.

(2) 해당 데이터에 어떤 문제가 있는지 분석하고 다양한 전처리 기법을 적용해 생성된 각각의 데이터들에 대해 학습 모형을 구축한다.

(3) 각 모형에 대해 평가 지표를 적용해 결과를 도출한다.

(4) 모든 평가 결과를 표와 그래프로 시각화하여 비교한다.

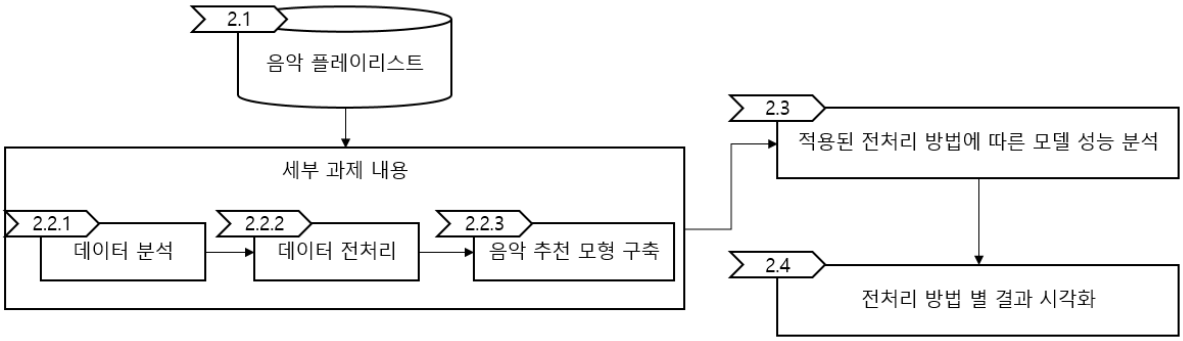

그림 1. 세부 과제 흐름도

2.1 음악 플레이리스트 데이터 입력

음악 추천 모델을 생성하기 위한 데이터로는 2020 년 kakao arena 대회에서 제공된 Melon Playlist 의 데이터셋을 사용한다.

데이터셋	속성	설명
플레이리스트	Id	플레이리스트 ID
	plylist_title	플레이리스트 제목
	Tags	태그 리스트
	Songs	곡 리스트
	like_cnt	좋아요 개수
	updt_date	수정 날짜
곡	_id	곡 ID
	album_id	앨범 ID
	artist_id_basket	아티스트 ID 리스트
	artist_name_basket	아티스트 리스트
	song_name	곡 제목
	song_gn_gnr_basket	곡 장르 리스트
	song_gn_dtl_gnr_basket	곡 세부 장르 리스트
	issue_date	발매일

표 1. Melon Playlist 데이터셋

2.2 세부 과제 내용

2.2.1 데이터 분석

수집한 음악 플레이리스트 데이터를 전처리하기 전 데이터셋의 특성을 이해하고 불균형, Outlier, Missing Value 와 같은 문제를 확인한다. 각 문제에 대한 설명과 탐색 방법은 다음과 같다.

Problem	설명	탐색 방법
불균형 데이터	범주형 데이터에 대해, 각 범주에 속하는 데이터 개수의 차이가 큰 데이터	인스턴스 분포 및 수치화를 통한 클래스 비율 비교

Outlier	관측된 데이터의 범위에서 많이 벗어난 아주 큰 값 또는 작은 값	IQR 을 통한 이상치 탐색
Missing Value	데이터에 값이 없는 것	데이터 테이블 탐색

표 2. 데이터 분석 방법

Outlier 탐색 방법에서 IQR 은, 데이터 분포를 1~4 사분위로 나타내어 25% -> Q1, 75% -> Q3 지점으로 두었을 때, Q1 – 1.5*(Q3 – Q1) 보다 작거나 Q3 + 1.5*(Q3 – Q1) 보다 큰 경우 이상치로 판단하는 기법이다.

2.2.2 데이터 전처리

데이터 분석에서 수행한 결과에 따라 전처리 기법을 적용한다. 본 과제에서 사용할 전처리 기법은 다음과 같다.

Problem	**전처리 기법**	**설명**	**세부 전처리 기법**
불균형 데이터	Oversampling	낮은 비율 클래스의 데이터 수를 늘려 불균형을 해소	Borderline SMOTE
			ADASYN
	Undersampling	높은 비율 클래스의 데이터 수를 줄여 불균형을 해소	One Sided Selection
	Weight	입력 신호가 결과에 영향을 미치는 중요도	ElasticNet
Outlier	Drop	이상치를 제거	
	Replace	이상치를 다른 값으로 대체	상한값
			하한값
			중앙값
			평균
	Scaling / Normalize	표준정규분포로의 표준화	
Missing Value	Drop	인스턴스를 제거	
	Replace	비어 있는 필드를 다른 값으로 대체	평균
			최빈값
			중앙값

고차원 데이터	차원 축소	데이터를 낮은 차원으로 투영	PCA
			LDA

표 3. 전처리 기법

각 전처리 기법 별 세부적인 방법들에 대한 추가적인 설명은 아래와 같다.

세부 전처리 기법	설명
Borderline SMOTE	• 소수 데이터 중 한 개를 선택해 K-NN 방식을 사용하여 가장 가까운 k 개의 소수 데이터 중 하나를 랜덤으로 선택해 새로운 데이터를 생성하는 SMOTE 방식을 활용 • Noise / Safe / Danger 관측치가 존재하며 Danger 에 속하는 경우에만 SMOTE 를 적용
ADASYN	• 소수 클래스 데이터와 가장 가까운 k 개의 소수 클래스 데이터 중 무작위로 선택된 데이터 사이의 직선상에 새로운 소수 클래스 데이터를 만드는 방법
One Sided Selection	• Tomek Links : 서로에게 더 가까운 다른 데이터가 존재하지 않는 서로 다른 클래스에 속하는 한 쌍의 데이터. 이러한 쌍을 찾아 다수 데이터에 속한 데이터를 제외시키는 방법 • CNN Rule : 소수 데이터들과 랜덤으로 선택된 다수 데이터 하나를 묶어 집합 S 를 생성하고, S 에 속하지 않는 데이터에 대해 k = 1 인 K-NN 을 사용해 제외시키는 방법 • One Sided Selection 은 Tomek Links 와 CNN Rule 을 합친 방법으로, Tomek Links 로 경계값의 다수 데이터를 제외한 후, CNN Rule 을 통해 나머지 데이터를 제외시킴
ElasticNet	• L2 : Bias 를 추가해 가중치의 절댓값을 줄이는 작업 • L1 : 중요하지 않은 특성들에 대해 가중치를 0 을 부여 • 엘라스틱넷은 이러한 L1 과 L2 를 함께 사용하여 가중치를 설정

PCA	• 입력 데이터가 들어왔을 때 분산이 가장 큰 축을 기준으로 차원을 축소
LDA	• 입력 데이터가 들어왔을 때 클래스 분리를 최대화 하는 축을 찾아 클래스 간 분산은 최대화하고 클래스 내부 분산은 최소화하는 차원 축소 기법

표 4. 세부 전처리 기법

2.2.3 음악 추천 모형 구축

초기 데이터셋에 각기 다른 전처리 기법을 적용하여 얻은 데이터에 대해 음악 추천 모형을 구축할 것이므로, Classification 을 사용한 모델을 생성할 것이다. 여기서 사용할 Classification 학습 알고리즘은 아래와 같다.

Classification 알고리즘	설명
Naïve Bayes	• feature 끼리 독립이라고 가정할 때, 그 확률을 단순히 곱하는 베이즈 공식을 사용해 분류
Logistic Regression	• Threshold 를 기준으로 특정 클래스에 속할 확률을 계산해 분류 • Binary Classification 에서 주로 사용
SVM	• 서로 다른 클래스 간에 속하는 데이터들이 최대 distance 를 가지도록 하는 Hyperplane 으로 분류 • 가장 효과적이지만, 가장 느리다는 단점도 포함
K-NN	• Lazy learning 을 통해 새로운 인스턴스에 대해 K-NN 을 적용해 분류

표 5. Classification 알고리즘

2.3 전처리 방법에 따른 모델 성능 분석

각 전처리 방법에 따른 음악 추천 모형의 성능 비교를 위해 Classification Model 의 대표적인 성능 평가 방법을 적용한다. 여기에서는 다음과 같은 평가 지표를 사용한다.

성능 평가 지표	설명
F1-Score	• Confusion Matrix 에서 측정한 Precision 과 Recall 의 조화평균 • Precision 과 Recall 중 더 작은 값에 영향을 더 크게 받게 됨
ROC - AUC	• Threshold 값을 변화시키며 분류 문제에 대한 성능을 측정 • ROC 곡선 아래의 면적 AUC 가 클수록 높은 성능

표 6. 성능 평가 지표

F1-Score 와 ROC – AUC 는 기본적으로 다음 Confusion matrix 에서 구할 수 있는 지표를 사용한다.

		Predicted	
		Positive	Negative
Observed	Positive	TP	FN
	Negative	FP	TN

표 7. Confusion matrix

2.4 전처리 방법 별 결과 시각화

적용된 전처리 방법 별로 모델 생성 시간과 노래 추천을 위한 수행 시간을 합친 총 실행 시간, F1-Score, ROC-AUC Score 를 기반으로 표와 그래프를 생성한다. 이 때 각 평가 지표 별 결과값을 matplotlib 를 이용해 그래프로 시각화하고, pandas 를 이용해 csv 파일로 저장한다. 각 평가 지표에 활용할 그래프 형태는 다음과 같다.

평가 지표	그래프 형태
F1-Score	• X 축 : 적용된 전처리 기법

	• Y 축 : F1-Score • 그래프 형태 : 막대 그래프 (matplotlib.bar)
ROC - AUC	• X 축 : False Positive Rate ($FPR = \frac{FP}{FP+TN}$) • Y 축 : True Positive Rate ($TPR = \frac{TP}{TP+FN}$) • 그래프 형태 : 꺾은 선 그래프 (matplotlib.plot)
총 실행 시간 (모델 생성 시간 + 수행 시간)	• X 축 : 적용된 전처리 기법 • Y 축 : 시간 (ms) • 그래프 형태 : 막대 그래프 (matplotlib.bar)

표 8. 평가 지표 별 그래프 형태

3. 설계 문서

사용 기술 및 개발 환경	설명
Python	• 모델 개발은 Python 으로 수행한다. Python 은 다양한 서드파티 라이브러리의 존재 덕에 머신러닝 분야에서 다른 언어에 비해 효율적인 개발이 가능하다.
Scikit-learn	• Python 에서 머신러닝 개발을 지원하는 라이브러리인 scikit-learn 을 본 과제의 데이터 전처리 및 학습 모델 개발을 위해 채택했다.
Jupyter Notebook	• Python 및 scikit-learn 라이브러리를 기반으로 머신러닝 시스템을 효과적으로 개발하고 분석하기 위해 Jupyter Notebook 을 이용해 개발을 수행한다. • 블록 단위 코드 실행이 가능해 개발 시 디버깅이 편리하고 마크다운 언어를 지원해 소스 코드의 문서화에 용이하다.
Matplotlib	• Python 의 라이브러리인 matplotlib 은 다양한 그래프를 그리는 도구를 제공해 시각화를 용이하게 한다.
Pandas	• Python 의 라이브러리인 pandas 는 데이터 입출력 및 데이터 가공을 위한 다양한 기능을 제공한다.

표 9. 사용 기술 및 개발 환경

4. 전체 구상도

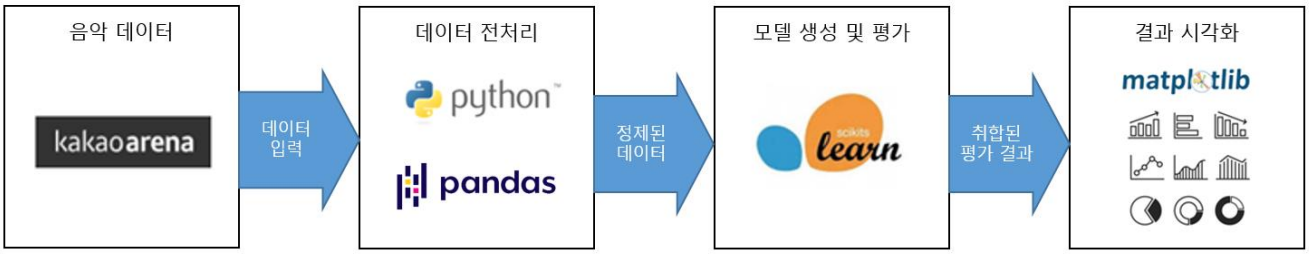

그림 2. 전체 구상도

5. 개발 일정 및 역할 분담

5.1. 개발 일정

5 월			6 월					7 월					8 월					9 월			
3 주	4 주	5 주	1 주	2 주	3 주	4 주	5 주	1 주	2 주	3 주	4 주	5 주	1 주	2 주	3 주	4 주	5 주	1 주	2 주	3 주	4 주
착수보고서																					
	데이터 전처리 기술 학습																				
						데이터 분석 및 시각화															
								데이터 전처리													
									음악 추천 모형 학습												
											중간보고서										
													음악 추천 모형 구축								
																결과 분석 및 최적화					
																		최종 테스트			
																		최종보고서 및 발표			

표 10. 개발 일정

5.2. 역할 분담

이름	역할
진현	음악 추천 모형 구축 모델 평가 및 최적화
임우영	데이터 분석 및 전처리 모델 평가 및 최적화
김영수	데이터 분석 및 전처리 데이터 시각화
공통	음악 추천 모형 원리 학습 중간보고서 및 최종보고서 작성

표 11. 역할 분담