1. 인사, 목차
   1. 안녕하세요 가사 기반 카테고리별 노래 자동 분류 모델을 주제로 발표할 2조 전민정입니다. 저희 조는 김성모, 이성연, 이현중 님으로 구성되어 있습니다. 발표 시작하도록 하겠습니다.
   2. **[클릭]**목차는 배경 및 주제, 데이터 설명, 분석 과정, 분석 결과 순으로 발표하도록 하겠습니다.
2. 배경 및 주제선정
   1. **[클릭]**노래의 장르로는 어떤 장르가 있을까요?**[클릭]** 발라드,**[클릭]** 락,**[클릭]** 댄스와 같이 다양한 노래 장르로 나눌 수 있습니다. 또한 사랑을 주제로 하는 노래처럼 주제로 노래를 분류할 수도 있습니다. 다양한 노래 장르나 주제만큼 사람마다도 제각기 취향이 다르고 듣는 노래도 다릅니다. **[클릭]**다음은 2012년도에 지식인에 올라온 질문입니다. 이에 대한 답변은 무엇일까요?**[클릭]** ‘이 사람 노래가 내 마음을 읽습니다.’ **[클릭]**내 마음을 읽는다는 말은 호소력도 있다고도 생각되지만 가수가 부른 가사 내용에도 관심이 가게 되는데요, 가사는 노래에 중요한 부분입니다. 노래들의 가사에 나타나는 공통점과 차이점을 생각 해 보았고, 우리가 실제로 생각한 것과 생각한 것을 모델로 구현하였을 때 얼마나 일치하는지 보면 재밌을 것 같다고 생각했습니다.
   2. **[클릭]**이를 바탕으로 우리는 노래 가사의 단어를 추출하여, **[클릭]**가사를 바탕으로 노래의 장르 또는 주제를 분류할 수 있는 모델들을 구현해 보고, **[클릭]**나아가 어떤 모델이 노래를 잘 분류할 수 있는지 (우리가 구현한 모델 중) **[클릭]**최적의 모델을 찾아보자는 주제로 본 프로젝트를 진행하기로 하였습니다.
3. 데이터 설명
   1. **[클릭]**출처: 데이터의 출처는 멜론, 벅스, 지니입니다.
   2. **[클릭]**데이터 규모: 각 사이트별로 1960년도 부터 2021년까지 각 연도별 top 100 의 노래를 총 12000여곡을 크롤링하였습니다. 후에, 1960~63년도 데이터는 다른 년도의 데이터와 길이 등이 맞지 않고, 소실된 데이터가 많아 제외하였습니다. 중복되는 곡들을 소거하니 1964년도부터 년도별로 100개 이상의 곡이 모여 **[클릭]**총 6,200개의 곡이 있는 데이터프레임이 완성되었습니다.
   3. **[클릭]**데이터 피처로는 작곡 년도, 가수, 노래 제목, 가사가 있었습니다. 저희의 타겟은 노래 가사이며, 이 노래 가사들을 토큰화하였습니다.
   4. **[클릭]**각 노래 별로 부여한 라벨(장르, 주제)과 가사가 얼마나 밀접한 관련을 보이는지 그 상관관계를 학습하고자 하였습니다.
4. 분석과정
   1. **[클릭]**수집과정: 데이터의 수집과정은 앞에서 말씀드렸듯, 멜론, 지니, 벅스 웹사이트에서 크롤링을 진행하였고, **[클릭]**여러개 음원 사이트 중에서, 보시는 것처럼 연도별 음원차트가 있는 사이트만 사용하였습니다.
   2. **[클릭]**라벨링: 각 노래별 가사의 공통점을 파악하여 **[클릭]**최초 8개의 라벨(사랑, 효, 후회, 욕설, 자연, 희망, 돈, 우정)을 만들었습니다. 1명당 1550개 곡에 직접 라벨링을 하였으며, 추가적인 라벨 필요시 슬랙으로 소통하며, 6개의 라벨을 추가하여 최종적으로 총 14개의 라벨을 부여하게 되었습니다. 라벨 부여 기준은 해당 라벨과 가사의 상관관계가 어느 정도 있다고 판단될 경우로 하였으며, 애매한 곡들은 투표를 통해 진행하였습니다. 각 노래당 최소 1개의 라벨을 부여하였으며,**[클릭]** 복수개의 라벨을 부여한 노래도 존재합니다. **[클릭]**이렇게 복수 개의 라벨이 있었기 때문에**[클릭]** 각 라벨에 해당하는 열을 만들고 **[클릭]**원핫인코딩 방식으로 변환하였습니다.
   3. 전처리:
      1. **[클릭]**전처리를 크게 5단계로 나눠보았습니다.(잠깐 보여주고)**[클릭]** (잠깐 보여주고)**[클릭]**먼저는 데이터 크롤링이고, 다음은 정규표현식을 이용하여 노이즈를 제거하고, 번역을 진행하였습니다.
      2. **[클릭]**번역: 최근 노래의 경우 영어로 된 가사가 많았고, 몇몇 노래의 경우 다른 나라 언어도 있었습니다. 그렇기에 번역을 하여 한국어로 변환하기로 하였습니다. 순서는 상관없기 때문에 영어를 추출하여 구글트랜스라는 API를 사용하여 번역하여 한국어 가사 뒤에 붙이는 방식으로 진행하였습니다. **[클릭]**다른 나라 언어로 되어있어도 영어로 적혀있고, 소수민족의 언어가 아닌 이상 자동으로 번역을 해주었습니다.
      3. **[클릭]**OKT를 사용하였고, **[클릭]**명사, 동사, 형용사, 부사로 분류되는 단어들은 남겨두었습니다. 불용어는 사전을 제작하였는데요, **[클릭]**불용어 기준은 각 단어별 document frequency 출력 이후, 상위 2% 빈도수 단어들을 직접 검토하여**[클릭]** 불용어 사전을 작성하였습니다.
      4. **[클릭]**벡터라이징: 벡터라이징은 Bow, TF-IDF 두 가지 방법으로 진행하였습니다. **[클릭]**최종적으로 이랬던 방대한 데이터 덩어리가 **[클릭]**이렇게 우리에게 필요한 토큰들이 되었습니다.
   4. 분석
      1. **[클릭]**먼저 불용어 사전 없이 결과값을 출력하여 보았습니다. 후에 불용어 사전을 만들어 전처리를 하니 더 좋은 결과값이 나왔습니다. 보였던 패턴은, 볼륨이 큰 라벨들의 데이터는 overfitting을 보이는 경향성을 띄었습니다 (very high train score, relatively low test score). 반면에, 볼륨이 작은 라벨들의 데이터는 학습을 거의 완벽하게 하였습니다. 저희는 이 이유가, 볼륨이 작은 라벨들의 데이터에 비해, 볼륨이 큰 라벨들은 다뤄야할 feature가 훨씬 많아서, 과도한 학습을 할 수 있다는 가능성을 있다고 생각하였습니다. 또한, 볼륨이 작은 라벨들 (e.g., 돈, 효, sf) 등은 좀 더 확실한 키워드가 요구되는 반면에 볼륨이 큰 라벨들(e.g., 사랑, 후회) 등은 그 키워드의 개수도 다양하다는 것이 문제였다고 생각합니다.
      2. **[클릭]**Stratified K-fold라는 방법으로 모든 데이터를 학습과 검증에 사용할 수 있는 교차검증을 진행하였고, 정확도가 평균으로부터 어느 정도 떨어져있는지를 찾아낸 이후에, 큰 편차가 없다면 runtime을 줄이기 위하여**[클릭]** k-fold의 데이터폴드 세트(n-split) 값을 10에서 3으로 낮춰주었습니다.
      3. **[클릭]**라벨별로 휴먼러닝을 진행하였습니다. 더 나은 test값을 도출해낼 수 있는 parameter가 있지 않을까 생각하여, 기본적으로 주어진 min\_df, max\_df, n\_estimators 등에 계속 변화를 주었지만, 변화에 있어 확실한 방향성을 제시하기는 어려웠습니다. **[클릭]**라벨 별로 결과가 잘 나오는 벡터라이징 방법이 달랐으며, 사랑의 경우는 Bow가, 후회의 경우 TF-IDF의 결과가 더 좋았습니다. 특정 라벨의 경우 50곡 미만이기도 하여 생략하기도 하였습니다. 후에 GridSearchCV를 사용하여 하이퍼파라미터 튜닝을 진행하였습니다.
5. 분석 결과 및 결론
6. 결과:
   1. **[클릭]**모델 별 교차검증 결과를 시각화하였습니다.
   2. 러닝커브: 볼륨이 큰 라벨과 볼륨이 작은 라벨을 구분하여 라벨 별로 제일 좋은 결과를 가진 모델, 제일 안좋은 결과를 가진 모델의 learning curve도 그려보았습니다.
   3. **[클릭]**보팅: 최종적으로 보팅을 하여 나온 결과는 다음과 같습니다. 다음은 하드 보팅과 소프트 보팅 각각의 값입니다.
   4. **[클릭]**어떤 모델이 가장 성능이 좋은지, 돌려본 9개의 모델 중 라벨별로 성능이 좋은 모델을 찾고 싶었습니다. 상위는 점수가 높게 나오고, 하위는 점수가 낮게 나온 모델입니다. **[클릭]**상위 2개를 놓고 봤을 때 RFC, SVC, EXT 의 점수가 높게 나왔습니다. **[클릭]**하위 2-3개를 보면 ADA, DTC, KNN이 많이 나왔습니다.
   5. **[클릭]**멀티레이블: 옆에는 멀티레이블에 대해 돌려본 결과입니다. 노래 특성 상 한 노래 당 여러 라벨이 붙을 수 있는 멀티레이블이기 때문입니다. **[클릭]**멀티레이블로 했을 때도 앞에서 언급했던 RFC, EXT가 높게 나왔고, ADA, DTC, KNN이 낮게 나왔습니다. 이러한 결과로 앞의 분석이 어느정도 의미는 있었다고 생각합니다.
7. 개선사항:

i. **[클릭]**이를 바탕으로 한 개선사항에 대해 말씀드리겠습니다. **[클릭]**라벨링의 기준이 어느정도 모호했다고 생각됩니다. 네 명이 나눠서 라벨링을 했기 때문에 애매한 것은 투표로 진행하였지만, 모호하게 넘어간 것도 분명히 있었을 것이라고 생각됩니다.

ii.**[클릭]** 라벨 자체의 아쉬움입니다. 예를 들어 ‘사랑’이라는 라벨에서도 모호한 라벨링이 아닌 ‘직접적인 사랑 표현이 들어간 가사의 노래’에만 한정짓는다면, 이별, 만남, 연애 등 세분화된 라벨을 썼더라면, 분류가 조금은 더 잘되진 않았을까 생각해보았습니다. 또한 라벨이 사랑인 경우 해당되는 노래가 많기 때문에 피처가 많아 지는데 이로인해 train score와 test score의 차이가 컸다고 생각되었습니다. 이렇기에 세분화해서 레이블링을 했다면 피처를 줄일 수 있다는 면에서 그 차이를 조금 더 줄일 수 있을지 않을까 생각이 들었습니다.

iii. **[클릭]**마지막으로, 우리가 가졌던 6,200여 곡보다 훨씬 많은 곡을 모아서 처리할 수 있었다면 스코어를 높일 수 있진 않았을까 생각해보았습니다.

1. 확장가능성:

i. **[클릭]**날마다 올라오는 최신곡들을 **[클릭]**하루에 한번씩 자동으로 크롤링하여 각 카테고리별로 분류하여 저장해주는 코드를 작성할 수 있습니다. 이처럼 최신곡이 나올 때마다 가사를 분석하여 **[클릭]**자동으로 분류해주는 모델로서의 확장이 가능합니다.

ii.**[클릭]** 더 나아가 이를 추천 알고리즘과 결합하여 **[클릭]**자신이 많이 듣는 노래와 비슷한 가사의 최신곡을 **[클릭]**자동으로 추천해주는 프로그램으로까지 연결이 가능할 것으로 보입니다.

1. **[클릭]**Q&A 시간입니다.
2. **[클릭]**지금까지 발표를 들어주셔서 감사합니다.