**LSTM.ipynb** :노래 가사와 라벨링된 감정 데이터를 활용하여 데이터 전처리, 임베딩, 토큰화, 패딩, LSTM 딥러닝 다중분류 모델의 학습을 하고, 학습 후 test 셋을 활용하여 데이터의 학습 결과 정확도, f1 score 등을 확인할 수 있다. 또한 이진 분류 모델의 학습도 진행할 수 있으며, 자신이 원하는 노래에대한 감정 분류 결과 확인도 진행할 수 있다.

**bi\_LSTM\_multi.ipynb** : bi\_LSTM을 활용하여 keras 라이브러리로 구현한 노래 가사 감정 다중분류기이다. Train, test, validation set의 경로를 지정한 후 코드 셀을 차례대로 실행시켜 데이터 셋 구성 확인 및 토큰화, 임베딩, 패딩 등의 데이터 전처리, 모델 학습을 진행한다. 가사 데이터는 분노, 두려움, 슬픔, 행복 총 4가지의 감정으로 분류된다. 모델 학습 이후의 코드 셀 실행을 통해 모델의 성능(accuracy, F1score 등)을 확인할 수 있다. 그 후 감정분류를 원하는 데이터의 경로를 지정하여 데이터셋을 불러온 후 학습된 모델에 입력해 분류 결과를 확인한다.

**bi\_LSTM\_binary.ipynb** : bi\_LSTM을 활용하여 keras 라이브러리로 구현한 노래 가사 감정 이진분류기이다. 프로그램의 구성은 다중분류기와 거의 유사하다. Train, test, validation set의 경로를 지정한 후 코드 셀을 차례대로 실행시켜 토큰화, 임베딩, 패딩 등의 데이터 전처리 및 모델 학습을 진행하며 감정은 긍정과 부정 두 가지로 분류된다. 모델 학습 이후의 코드 셀 실행을 통해 모델의 성능(accuracy, F1score 등)을 확인할 수 있다. 그 후 감정분류를 원하는 데이터의 경로를 지정하여 데이터셋을 불러온 후 학습된 모델에 입력해 분류 결과를 확인한다.

**CNN\_multi.ipynb, CNN\_binary.ipynb** : CNN을 이용한 노래 가사 감정 다중/이진 분류기이다. 노래 가사에 감정이 라벨링된 데이터를 이용해 토큰화, 인코딩, 패딩 과정을 거쳐 전처리를 진행한 후, CNN 모델을 통해 학습을 진행하였다. test dataset을 통해 학습된 모델의 성능을 평가하였고, 새로운 노래 가사에 대해 어떻게 분류해내는지 파악할 수 있다.

**gru\_multi.ipynb:** gru를 활용하여 구현한 노래 가사 감정 다중분류기이다.

install: colab에 torchtext, konlpy, mecab 등을 설치해준다. 특히 torchtext.legacy 를 임포트 하기 위해 torchtext버전을 0.10.0로 설치

Import Library: 필요한 각종 라이브러리를 임포트

GPU Setting: colab의 GPU 설정

Hyper\_parameters: 하이퍼파라미터

Data Load: Train, Validation, Test set의 csv 파일을 데이터 프레임 불러온다.

Data PreProcessing: 인덱스열을 삭제한 후 다시 csv파일로 저장.

Create Field: 필드란 텐서로 표현 될 수 있는 텍스트 데이터 타입을 처리한다. 각 토큰을 숫자 인덱으로 맵핑시켜주는 단어장(Vocabulary) 객체가 있다. 또한 토큰화 하는 함수, 전처리 등을 지정할 수 있다.

Create Datasets: 데이터 세트는 위에 지정한 필드에 기반하여 데이터를 불러오는 작업을 한다. 보통 Train, Valid, Test 세트가 있으면 splits 메서드를 사용해서 만들어준다.

Build vocabulary: 토큰과 Interger index 를 매칭시켜주는 단어장을 생성한다. 단, 기본적으로 토큰을 0, 토큰을 1 로 만들어준다. 단, 필드지정시, 문장의 시작 토큰(init\_token)과, 끝의 토큰(eos\_token)을 넣으면 3, 4 번으로 할당된다. 메서드 안에는 생성한 데이터 세트를 넣어준다.

훈련 데이터를 기반으로 단어장을 생성하려면 아래의 명령어를 입력한다.

Create Data Loader: 데이터로더는 데이터셋에서 미니 배치만큼 데이터를 로드하게 만들어주는 역할을 합니다. 토치텍스트에서는 Iterator를 사용하여 데이터로더를 만듭니다.

GRU: 모델 정의

Train: 모델 학습

Evaluation: 모델 평가

**gru\_binary.ipynb** : gru를 활용하여 구현한 노래 가사 감정 이진분류기이다

install: colab에 torchtext, konlpy, mecab 등을 설치해준다. 특히 torchtext.legacy 를 임포트 하기 위해 torchtext버전을 0.10.0로 설치

Import Library: 필요한 각종 라이브러리를 임포트

GPU Setting: colab의 GPU 설정

Hyper\_parameters: 하이퍼파라미터

Data Load: Train, Validation, Test set의 csv 파일을 데이터 프레임 불러온다.

Data PreProcessing: 인덱스열을 삭제한 후 다시 csv파일로 저장.

Create Field: 필드란 텐서로 표현 될 수 있는 텍스트 데이터 타입을 처리한다. 각 토큰을 숫자 인덱으로 맵핑시켜주는 단어장(Vocabulary) 객체가 있다. 또한 토큰화 하는 함수, 전처리 등을 지정할 수 있다.

Create Datasets: 데이터 세트는 위에 지정한 필드에 기반하여 데이터를 불러오는 작업을 한다. 보통 Train, Valid, Test 세트가 있으면 splits 메서드를 사용해서 만들어준다.

Build vocabulary: 토큰과 Interger index 를 매칭시켜주는 단어장을 생성한다. 단, 기본적으로 토큰을 0, 토큰을 1 로 만들어준다. 단, 필드지정시, 문장의 시작 토큰(init\_token)과, 끝의 토큰(eos\_token)을 넣으면 3, 4 번으로 할당된다. 메서드 안에는 생성한 데이터 세트를 넣어준다.

훈련 데이터를 기반으로 단어장을 생성하려면 아래의 명령어를 입력한다.

Create Data Loader: 데이터로더는 데이터셋에서 미니 배치만큼 데이터를 로드하게 만들어주는 역할을 합니다. 토치텍스트에서는 Iterator를 사용하여 데이터로더를 만듭니다.

GRU: 모델 정의

Train: 모델 학습

Evaluation: 모델 평가