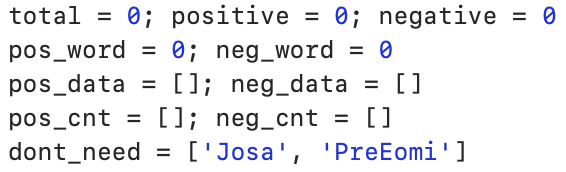
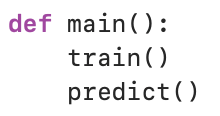
**Assignment 2**

2015004502 김형준

1. **코드 설명**
2. 전역 변수

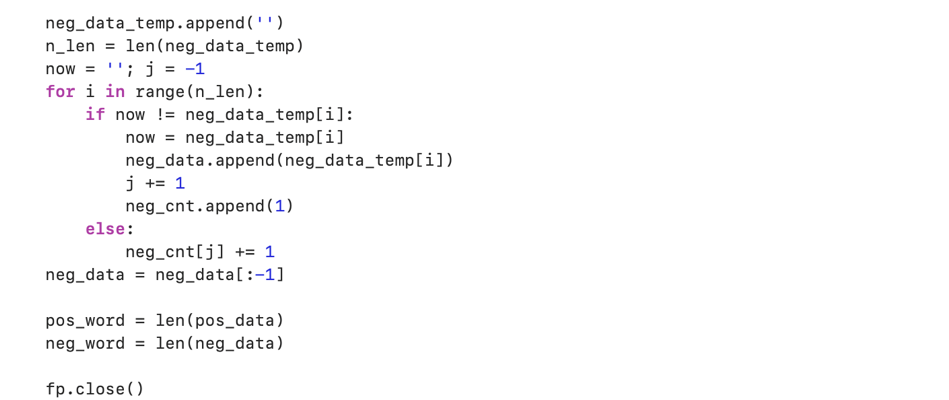
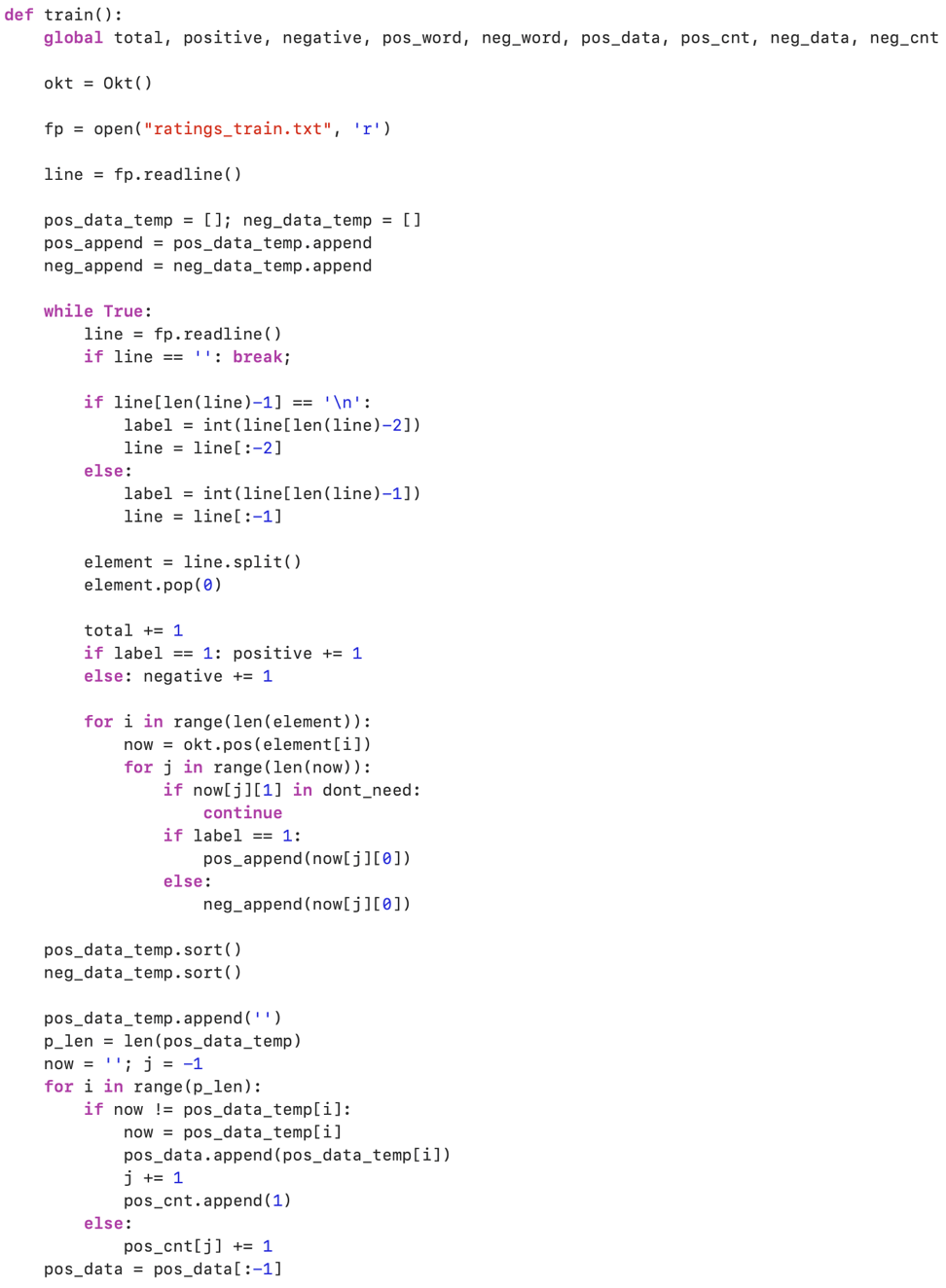
* Training 시 필요한 개수에 대한 정보와 형태소에 대한 정보를 저장하는 변수를 선언했다.
* 형태소 분석 시에 필요 없는 형태소의 종류를 정해주었다.

1. main



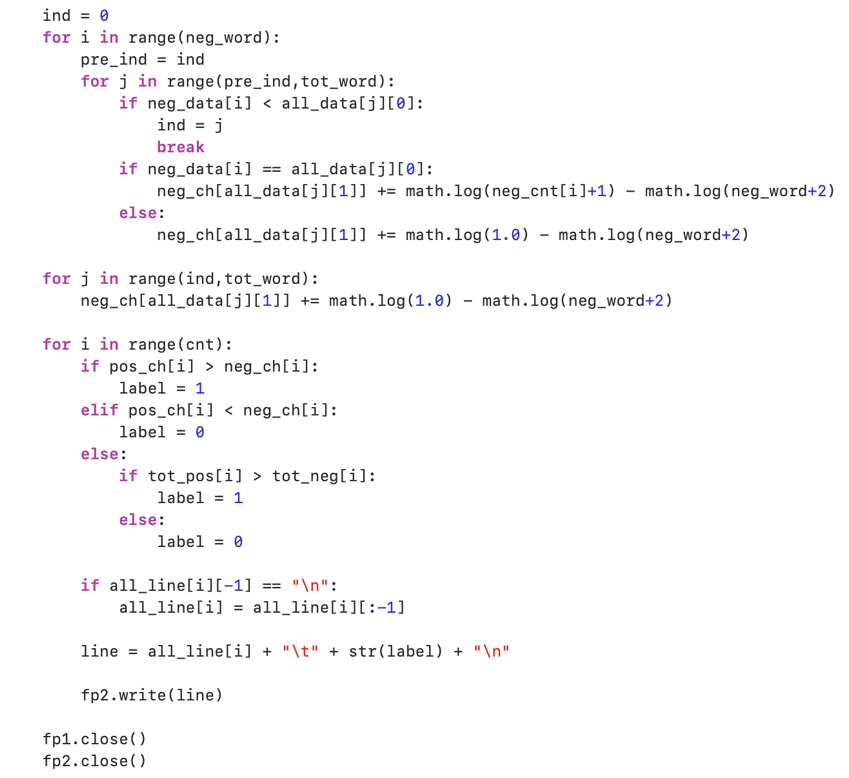
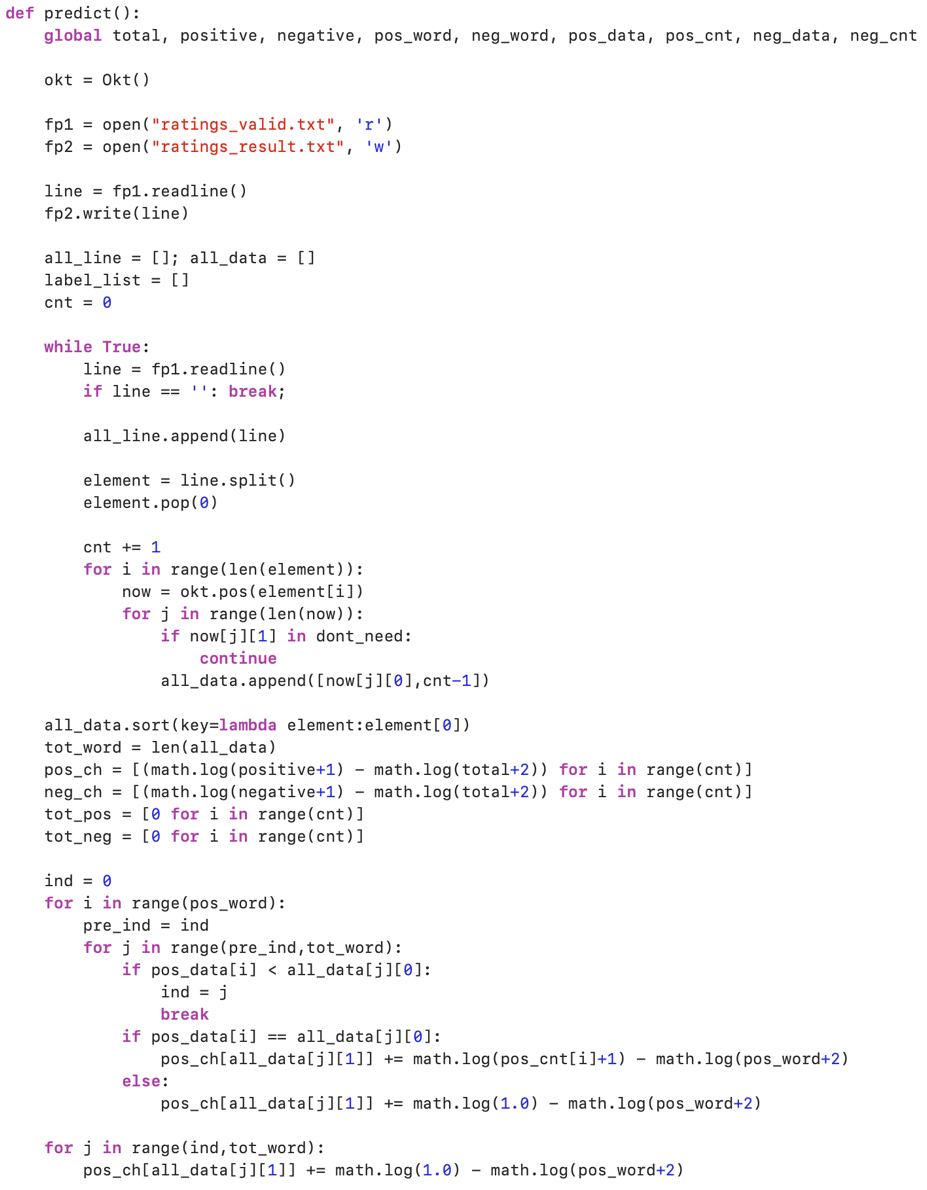
* main 함수를 통해 학습 시에 필요한 train 함수와 test 파일을 학습을 통한 결과를 도출해 내는 predict 함수를 호출하도록 설계했다.

1. train



* konlpy 의 okt 를 통해 한글 입력 값을 형태소 분석을 통해 구분해 주도록 설계했다.
* 각각 한 줄씩 입력을 받으며 label 즉, 0 인지 1 인지를 구분해주는 값을 추출 했다.
* 각각 입력 받은 형태소 요소에 따라서 긍정, 부정의 개수를 새는 것은 물론 모든 형태소를 pos\_data\_temp, neg\_data\_temp 에 추가해 주었다.
* 모든 값이 추가되면 정렬을 통해 같은 값들을 모아주고 각각 값들이 몇 개씩 존재하는지에 대한 정보들을 모두 저장해 주었다.
* 모든 형태소를 pos\_data, neg\_data 에 각각 형태소의 개수에 대한 정보를 pos\_cnt, neg\_cnt 에 저장해 주었다.

1. predict



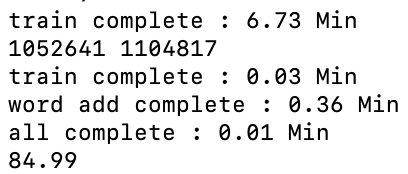
- test 파일 또한 각각 한 줄씩 입력 받으면서 형태소 분석을 해 모든 요소들을 all\_data 에 추가했다.

- 각 긍정에 들어가 있는 형태소, 부정에 들어가 있는 형태소를 all\_data 에 포함되어 있는지 없는지를 확인하며 모든 확률 정보를 log 를 통해 연산해 주었다. 이때, 확률이 0 이 되는 없는 형태소에 대한 값이 나오는 것을 방지하기 위해 모든 분자에는 1을 분모에는 2을 더해서 해결했다.

- 모든 값이 구해지면 초기에 설정한 전체 문장에서 긍정일 확률을 추가하면서 나이브 베이즈 분류 방식을 사용하도록 설계했다.

- 최종적으로 값이 결정 되면 긍정인지 부정인지 더 큰 확률이 나오는 것을 통해 결과 값을 정해주었고 만약 그 확률이 같게 된다면 각각 형태소가 긍정이나 부정에서 나오는 빈도수 비교를 통해 결과 값을 도출해 내었다.

1. **실험 결과**

****

* valid 파일을 통한 결과 값 비교를 실시해보니 입력을 처리해 처리한 결과 값을 정렬하고 train 을 마치는데 까지 약 6분이 소요 되었다.
* training 한 값으로 예측을 하는데 걸리는 시간은 짧은 시간이 소요되었으며 최종적으로 기존 결과 값과 맞게 되는 확률은 84.99% 가 나오게 되었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* test 파일을 train 을 통해 결과 값을 도출한 상위 일부 결과 값이다.
* 한눈에 보아도 정확도가 높음을 관측할 수가 있었다.