8장. 텍스트 분석 (1)

텍스트 분석은 머신러닝, 언어 이해, 통계 등을 활용해 모델을 수립하고 정보를 추출해 비즈니스 인텔리전스나 예측 분석 등의 분석 작업을 주로 수행한다.

- 텍스트 분류: 문서가 특정 분류 또는 카테고리에 속하는 것을 예측하는 기법
- 감정 분석: 텍스트에서 나타나는 감정/판단/믿음/의견/기분 등의 주관적인 요소를 분석하는 기법
- 텍스트 요약: 텍스트 내에서 중요한 주제나 중심 사상을 추출하는 기법
- 텍스트 군집화와 유사도 측정: 비슷한 유형의 문서에 대해 군집화를 수행하는 기법, 문서들간의 유사도를 측정해 비슷한 문서끼리 모을 수 있는 방법

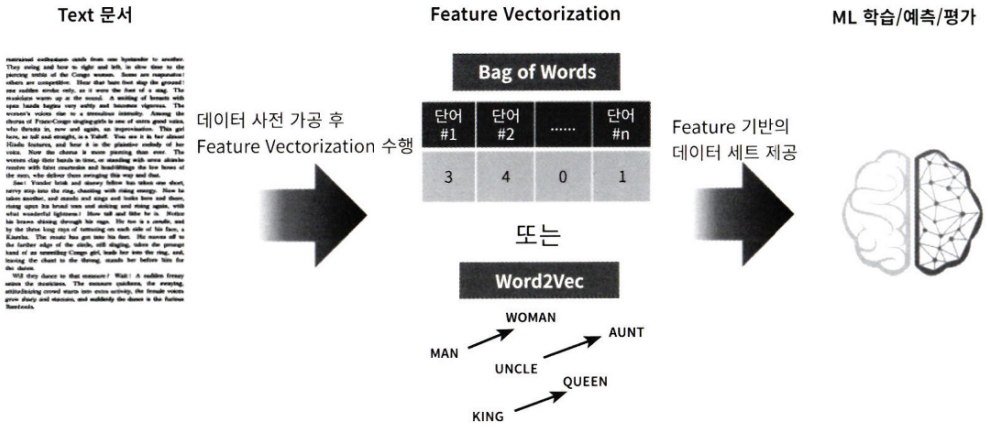
텍스트 분석 이해

텍스트를 머신러닝에 적용하기 위해서는 비정형 텍스트 데이터를 어떻게 피처 형태로 추출하고 추출된 피처에 의미 있는 값을 부여하는가 하는 것이 매우 중요

피처 벡터화(피처 추출): 텍스트를 word 기반의 다수의 피처로 추출하고 이 피처에 단어 빈도수와 같은 숫자 값을 부여하면 텍스트는 단어의 조합인 벡터값으로 표현될 수 있는데, 이렇게 텍스트를 변환하는 것을 말한다. → 대표적으로 BOW(Bag of Words)와 Word2Vec 방법이 있다.

텍스트 분석 수행 프로세스

1. **텍스트 사전 준비작업(텍스트 전처리)**: 텍스트를 피처로 만들기 전에 미리 클렌징, 대/소문자 변경, 특수문자 삭제 등의 클렌징 작업, 단어(Word) 등의 토큰화 작업, 의미 없는 단어(Stop word) 제거 작업, 어근 추출(Stemming/Lemmatization) 등의 텍스트 정규화 작업을 수행하는 것을 통칭합니다.
2. **피처 벡터화/추출**: 사전 준비 작업으로 가공된 텍스트에서 피처를 추출하고 여기에 벡터 값을 할당합니다. 대표적인 방법은 BOW와 Word2Vec이 있으며, BOW는 대표적으로 Count 기반과 TF-IDF 기반 벡터화가 있습니다.
3. **ML 모델 수립 및 학습/예측/평가**: 피처 벡터화된 데이터 세트에 ML 모델을 적용해 학습/예측 및 평가를 수행합니다.



〈Text Analytics 수행 프로세스〉

텍스트 사전 준비 작업(텍스트 전처리) - 텍스트 정규화

- **클렌징**: 텍스트에서 분석에 방해가 되는 불필요한 문자, 기호 등을 사전에 제거하는 작업
- **텍스트 토큰화**

  문장 토큰화: 문장의 마침표, 개행문자 등 문장의 마지막을 뜻하는 기호에 따라 분리하는 것이 일반적. 정규 표현식에 다른 문장 토큰화도 가능하다. NTLK에서 `sent_tokenize()`를 주로 사용.

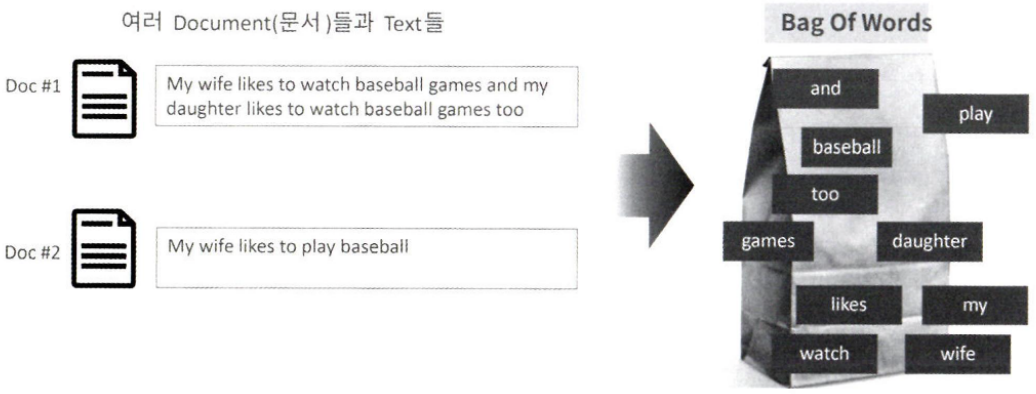
  단어 토큰화: 문장을 단어로 토큰화하는 것. 기본적으로 공백, 콤마, 마침표, 개행문자 등으로 단어를 분리하지만 정규표현식을 이용해 다양한 유형으로도 토큰화가 가능하다. NTLK에서 `word_tokenize()`를 주로 사용.
- **스톱 워드 제거**: is, the, a, will 등과 같이 문장을 구성하는 필수 문법 요소지만 문맥적으로 큰 의미가 없는 단어인 스톱워드를 제거하는 작업
- **Stemming과 Lemmatization**: 문법적인 요소에 따라 다양하게 변화한 혹은 의미적으로 변화한 단어의 원형을 찾는 작업

  Stemming은 원형 단어로 변환 시 일반 적인 방법을 적용하거나 더 단순화된 방법을 적용해 원래 단어에서 일부 철자가 훼손된 어근 단어를 추출하는 경향이 있다. 이에 반해 Lemmatization은 품사와 같은 문법적인 요소와 더 의미적인 부분을 감안해 정확한 철자로 된 어근 단어를 찾는다.

Bag of Words - BOW

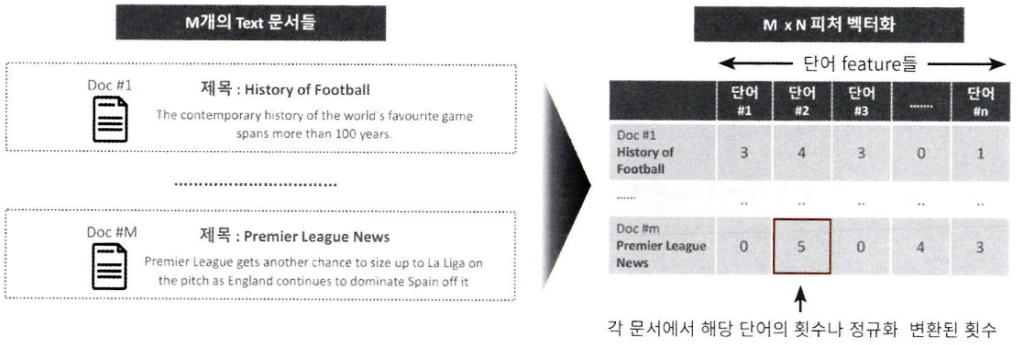
⭐ **Bag of Words** 모델은 문서가 가지는 모든 단어를 문맥이나 순서를 무시하고 일괄적으로 단어에 대해 빈도 값을 부여해 피처 값을 추출하는 모델

~~문맥 의미 반영 부족과 희소 행렬의 문제가 있다.~~



〈 Bag of Words 모델 〉

BOW 모델에서 피처 벡터화를 수행한다는 것은 모든 문서에서 모든 단어를 칼럼 형태로 나열하고 각 문서에서 해당 단어의 횟수나 정규화된 빈도를 값으로 부여하는 데이터 세트 모델로 변경하는 것



〈 BOW 피처 벡터화 〉

카운트 기반 벡터화: 단어 피처에 값을 부여할 때 각 문서에서 해당 단어가 나타나는 횟수, 즉 count를 부여

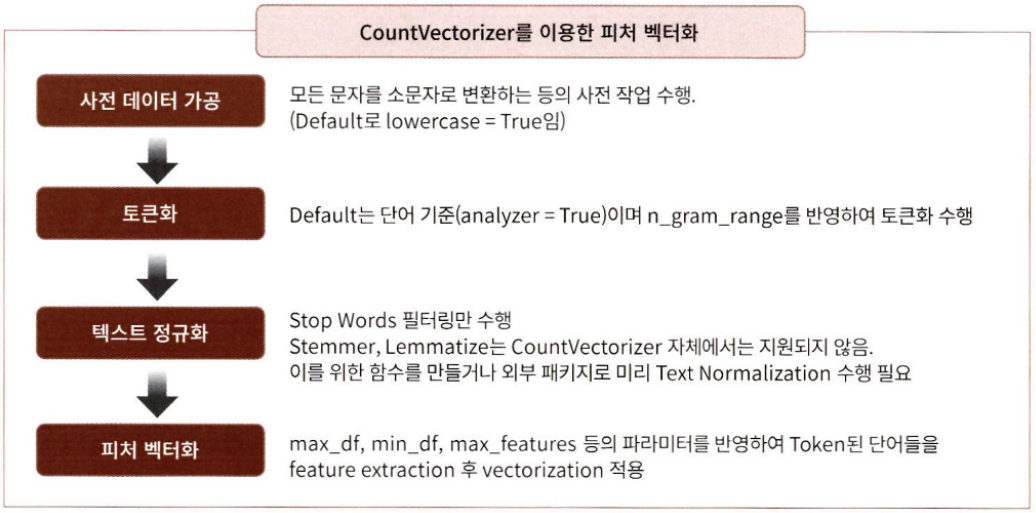
TF-IDF 벡터화: 개별 문서에서 자주 나타나는 단어에 높은 가중치를 주되, 모든 문서에서 전반적으로 자주 나타나는 단어에 대해서는 페널티를 주는 방식으로 값을 부여

문서마다 텍스트가 길고 문서의 개수가 많은 경우 카운트 방식보다는 TF-IDF 방식을 사용하는 것이 더 좋은 예측 성능을 보장

사이킷런의 Count 및 TF-IDF 벡터화 구현

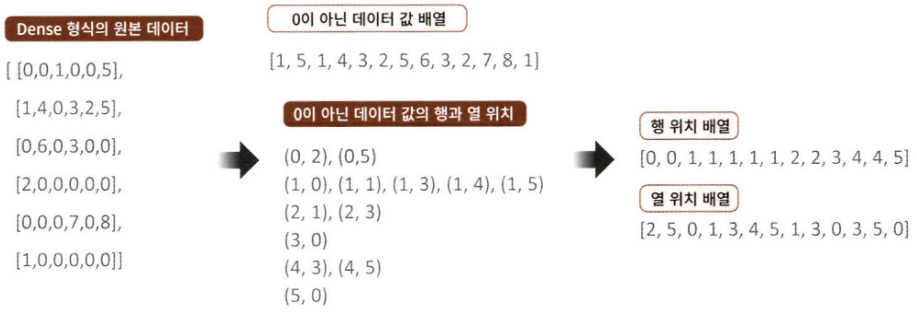
CountVectorizer 클래스는 카운트 기반의 벡터화를 구현. 피처 벡터화뿐만 아니라 소문자 일괄 변환, 토큰화, 스톱 워드 필터링 등의 텍스트 전처리도 함께 수행한다.

TfidfVectorizer 클래스는 TF-IDF 벡터화를 구현. 파라미터와 변환 방법은 위의 경우와 동일하다.

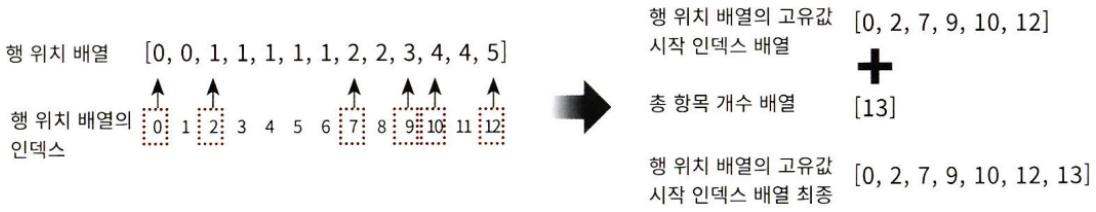


BOW 벡터화를 위한 희소 행렬

COO 형식: 0이 아닌 데이터만 별도의 배열에 저장하고, 그 데이터가 가리키는 행과 열의 위치를 별도의 배열로 저장하는 방식.



CSR 형식: COO 형식이 행과 열의 위치를 나타내기 위해서 반복적인 위치 데이터를 사용해야 하는 문제점을 해결한 방식. 행 위치 배열 내에 있는 고유한 값의 시작 위치만 다시 별도의 위치 배열로 가진다. COO 방식보다 메모리가 적게 들고 빠른 연산이 가능하다.



감성 분석

> ⭐ 감성 분석은 문서의 주관적인 감성/의견/감정/기분 등을 파악하기 위한 방법으로 문서 내 텍스트가 나타내는 여러 가지 주관적인 단어와 문맥을 기반으로 감성 수치를 계산한다.

이러한 감성 지수는 긍정 감성 지수와 부정 감성 지수로 구성되며 이들 지수를 합산해 긍정 감성 또는 부정 감성을 결정한다.

- 지도학습은 학습 데이터와 타깃 레이블 값을 기반으로 감성 분석 학습을 수행한 뒤 이를 기반으로 다른 데이터의 감성 분석을 예측하는 방법으로 일반적인 텍스트 기반의 분류와 거의 동일합니다.
- 비지도학습은 'Lexicon'이라는 일종의 감성 어휘 사전을 이용합니다. Lexicon은 감성 분석을 위한 용어와 문맥에 대한 다양한 정보를 가지고 있으며, 이를 이용해 문서의 긍정적, 부정적 감성 여부를 판단합니다.